PENENTUAN QUERY OPTIMAL PADA BASIS DATA TERDISTRIBUSI MENGGUNAKAN METODE TEACHER LEARNER BASED OPTIMIZATION (TLBO)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Lucky Nafi' Agtiannisa
NIM: 125150400111032

SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

PENENTUAN QUERY OPTIMAL PADA BASIS DATA TERDISTRIBUSI MEGGUNAKAN METODE TEACHER – LEARNER BASED OPTIMIZATION (TLBO)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Lucky Nafi' Agtiannisa
NIM: 125150400111032

Skripsi ini telah diuji dan dinyatakan lulus pada
28 Juni 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Satrio Agung W., S.Kom, M.Kom	Niken Hendrakusma W., S.Kom., M.Kom
NIK: 19860521 201212 1 001	NIK: -

Mengetahui
Ketua Program Studi Sistem Informasi

Suprapto , S.T, M.T
NIP: 19710727 199603 1 001

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 6 Juni 2016

METERAI TEMPEL
PAJAK MEMBANGUN BANGSA
TGL. 20
94956ABF104247442
ENAM RIBU RUPIAH
6000 DJP

Lucky Nafi' Agtiannisa

NIM: 125150400111032

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

KATA PENGANTAR

Puji syukur alhamdulillah penulis panjatkan kepada Allah SWT, karena berkat, rahmat dan hidayah-Nya lah penulis dapat menyelesaikan penyusunan skripsi yang berjudul "Penentuan *Query* Optimal Pada Basis Data Terdistribusi Menggunakan Metode *Teacher Learner Based Optimization* (TLBO)".

Penyusunan skripsi ini diajukan untuk memenuhi salah satu persyaratan kelulusan serta memperoleh gelar Sarjana Komputer di Program Studi Sistem Informasi, Fakultas Ilmu Komputer Universitas Brawijaya Malang. Dalam menyusun skripsi ini tentu tidak lepas dari halangan dan rintangan, namun berkat bimbingan, bantuan, nasihat dan saran serta kerjasama dari berbagai pihak, maka segala halangan dan hambatan tersebut dapat diatasi dengan baik. Oleh karena itu penulis dengan tulus hati ingin menyampaikan ucapan terimakasih kepada:

1. Bapak Satrio Agung Wicaksono, S.Kom, M.Kom., selaku dosen pembimbing I yang telah memberikan tenaga, waktu, bimbingan, ilmu, arahan, nasihat dan masukan dengan tulus ikhlas selama proses penyusunan skripsi ini.
2. Ibu Niken Hendrakusuma Wardani , S.Kom, M.Kom., selaku dosen pembimbing II yang telah memberikan tenaga, waktu, bimbingan, ilmu, arahan, nasihat dan masukan dengan tulus ikhlas selama proses penyusunan skripsi ini.
3. Bapak Suprapto, S.T, M.T selaku ketua Program Studi Sistem Informasi.
4. Bapak Aditya Rachmadi, S.ST., M.TI, selaku dosen pembimbing akademik yang telah memberikan arahan, nasehat, masukan dan semangat selama penulis menjalani masa studi di Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc., selaku dosen fasilitator yang telah memberikan tenaga, waktu, ilmu, arahan, nasihat dan masukan dengan tulus ikhlas selama proses penyusunan skripsi ini.
6. Seluruh Bapak dan Ibu dosen beserta karyawan Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu dan bimbingan selama penulis menjalani masa studi.
7. Bapak Agus Purwanto, Ibu Lilik Nurhidayati dan Afifah Khairunnisa Agtiananda atas kesabaran, dukungan moril dan materil dan doa serta kasih sayang yang diberikan kepada penulis selama ini.
8. Muhammad Zainal Faizin yang selalu memberi dukungan, semangat dan motivasi kepada penulis selama menyelesaikan skripsi ini.
9. Auziyah Novina Cahyawardani, Yayi Fateka D.P, Merrysahla Fitria Alifia, Gladina Desideria dan Laqma Dica Fitrani sahabat tersayang yang telah memberikan dukungan dan motivasi kepada penulis selama menyelesaikan skripsi ini.
10. Seluruh Keluarga Besar Sistem Informasi 2012 yang telah memberikan dukungan dan motivasi kepada penulis selama ini.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

11. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan dukungan dan bantuan yang sangat berharga selama menjalani studi dan menyelesaikan skripsi ini.

Penulis menyadari bahwa penyusunan dan penulisan tugas akhir ini masih mempunyai banyak kekurangan. Oleh karena itu, kritik dan saran dari semua pihak dibutuhkan untuk kesempurnaan tugas akhir ini. Penulis berharap tugas akhir ini memberikan sumbangsih bagi pihak-pihak yang berkepentingan dan bagi masyarakat.

Malang, 6 Juni 2016

Penulis

lucky.agtiannisa@gmail.com

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

ABSTRAK

Basis data terdistribusi adalah basis data yang secara *logic* terhubung dari kumpulan-kumpulan data yang digunakan bersama-sama namun secara fisik terbagi menjadi beberapa situs yang didistribusikan melalui suatu jaringan komputer. *Query processing* lebih sulit pada lingkungan terdistribusi daripada lingkungan terpusat karena terdapat banyak parameter yang mempengaruhi kinerja *distributed query*. Sehingga memungkinkan terdapat beberapa *query equivalent* atau alternatif dari *query* terdistribusi. Dari beberapa *query equivalent* yang ada dicari mana yang paling efisien khusunya dari segi biaya eksekusi sehingga dibutuhkan metode untuk menentukan *query* yang paling optimal. Metode *Teacher learner based optimization (TLBO)* diusulkan pada penelitian ini untuk melakukan perbandingan dari keseluruhan *query equivalent* yang ada sehingga dapat ditentukan *query* yang paling optimal. Dimana metode *Teacher learner based optimization (TLBO)* berdasar pada proses belajar mengajar yang dilakukan guru dan murid, sehingga terdapat dua fase pada metode ini yakni fase *teacher* dan fase *learner*. Dengan melakukan perhitungan biaya dari tiap *query* yang *equivalent* berdasar parameter yang dipilih selanjutnya hasil biaya dari setiap *query equivalent* diproses dengan metode *Teacher learner based optimization (TLBO)* dan menghasilkan pengoptimalan biaya menjadi lebih rendah dan menghasilkan *query equivalent* mana yang memiliki biaya yang paling optimal.

Kata kunci: BDT, *Query* optimal, Pemrosesan *query* terdistribusi, *Teacher Learned Based Optimization (TLBO).*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

Distributed database is a database that logicly connected from data sets which are used together but physically divided into several sites that distributed through a computer network. Query processing is more difficult in a distributed environment rather than in a centralized environment because there are many parameters that affect the performance of distributed queries. So that there can some equivalent queries or distributed query alternatives. From some of the existing equivalent queries, sought the most efficient especially in terms of execution cost, so its need methods to determine which the most optimal query. Teacher learner-based optimization method (TLBO) proposed in this study to undertake a comparison of the overall existing equivalent query so that the most optimal query can be determined. Which Teacher learner-based optimization method (TLBO) is based on the teaching and learning process between teachers and students, so that there are two phases in this method, teacher phase and learner phase. By calculating the costs of each query that is equivalent based on the selected parameter, theb the results equivalent cost of each query is processed by the Teacher learner-based optimization method (TLBO) and yield optimization in lower costs and generate equivalent query which has the most optimal cost.

Keywords: BDT, Optimal Query, Distributed Query Processing, Teacher Learned Based Optimization (TLBO).

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

HALAMAN JUDUL i
PERSETUJUAN ..... **Error! Bookmark not defined.**
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK vi
ABSTRACT vii
DAFTAR ISI viii
DAFTAR TABEL xiv
DAFTAR GAMBAR xvii
DAFTAR LAMPIRAN xx
BAB 1 PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Rumusan masalah 3
1.3 Tujuan 3
1.4 Manfaat 3
1.5 Batasan masalah 4
1.6 Sistematika pembahasan 4
BAB 2 LANDASAN KEPUSTAKAAN 6
2.1 Kajian Pustaka 6
2.2 Basis Data 6
2.2.1 Basis Data Terdistribusi 6
2.2.2 Arsiektur Basis Data Terdistribusi 8
2.2.3 Klasifikasi Basis Data Terdistribusi 9
2.2.4 Fragmentasi 9
2.3 Aljabar Relasional 10
2.3.1 Operasi *Selection* 11
2.3.2 Operasi *Projection* 12
2.3.3 Operasi *Cartesian Product* 13
2.3.4 Operasi *Union* 14
2.4 Operator Aljabar Relasional 14

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.4.1 *Generalized Projection* 14
2.4.2 Agregate Function 15
2.5 Kalkulus Relasional 16
2.5.1 Kalkulus Relasional Tupel 16
2.5.2 Kalkulus Relasional Domain 17
2.6 *Distributed Query Processing* 17
2.6.1 *Query Decomposition* 18
2.6.2 Data *Localization* 18
2.6.3 *Global Query Optimization* 19
2.6.4 *Distributed Query Execution* 19
2.7 *Query Optimization* 19
2.7.1 *Search Space* 20
2.7.2 *Search Strategy* 20
2.7.3 *Cost Based Optimization* 21
2.8 Algoritme Optimasi 21
2.8.1 Deterministik 21
2.8.2 *Stochastic* 21
2.9 *Teacher-Learner Based Optimization (TLBO)* 22
2.9.1 Fase Teacher 23
2.9.2 Fase Leaner 24
2.9.3 Desain Parameter Heuristik 24
BAB 3 METODOLOGI 27
3.1 Studi Literatur 28
3.2 Observasi 28
3.3 Desain Basis Data Terdistribusi 29
3.4 Perancangan Populasi *Query Plan* 29
3.5 Implementasi Basis Data Terdistribusi 30
3.6 Implementasi Populasi *Query Plan* 30
3.7 Pengujian Populasi *Query Plan* 30
3.8 Pengujian Metode *Teacher Learner Based Optimization (TLBO)* 31
3.9 Analisis Hasil 32
BAB 4 HASIL PERANCANGAN 33

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.1 Jaringan Komputer 33
4.2 Basis Data 34
4.2.1 Relational Database Manaement System 34
4.2.2 Skema Fragmentasi Vertikal 36
4.2.3 Data SIAKAD Unversitas Brawijaya 38
4.2.4 Fragmentasi Data 39
4.2.5 Skema Basis Data Terdistribusi 41
4.3 Pemilihan *Query* 41
4.3.1 *Query* 1 42
4.3.2 *Query* 2 46
4.3.3 *Query* 3 49
4.4 Penerapan *Teacher Learner Based Optimization* 55
BAB 5 IMPLEMENTASI 59
5.1 Lingkungan Perangkat Lunak 59
5.2 Lingkungan Perangkat Keras 59
5.3 Implementasi *Virtual Server (Site)* 59
5.4 Implementasi Basis Data 60
5.4.1 Implementasi *Site* 60
5.4.2 Implementas Tabel 62
5.5 Implementasi Data 84
5.5.1 Tabel MASUK 85
5.5.2 Tabel K_TEMP 86
5.5.3 Generate Timestamp 87
5.5.4 Drop Tabel K_TEMP dan Tabel MASUK 88
5.6 Implementasi Federasi Basis Data 89
5.6.1 Konfigurasi Basis Data 89
5.6.2 *Catalog Node* 89
5.6.3 *Catalog* Basis Data 90
5.6.4 Membuat *Wrapper* 91
5.6.5 Membuat *Server* Federasi Basis Data 91
5.6.6 Pembuatan *User Mapping* 92
5.6.7 Pembuatan *Nick Name* Tabel 93

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5.7 Implementasi *Import* Data 96
5.7.1 Fragmen 1 (*Site* 1) 96
5.7.2 Fragmen 2 (*Site* 2) 98
5.7.3 Fragmen 3 (*Site* 3) 100
5.7.4 Fragmen 4 (*Site* 4) 101
5.7.5 Fragmen 5 (*Site* 5) 104
5.8 Implementasi Model Akses *Query* 106
5.8.1 *Query* 1 Model 1 106
5.8.2 *Query* 1 Model 2 107
5.8.3 *Query* 1 Model 3 108
5.8.4 *Query* 1 Model 4 109
5.8.5 *Query* 1 Model 5 110
5.8.6 *Query* 2 Model 1 111
5.8.7 *Query* 2 Model 2 112
5.8.8 *Query* 2 Model 3 113
5.8.9 *Query* 2 Model 4 114
5.8.10 *Query* 2 Model 5 115
5.8.11 *Query* 3 Model 1 116
5.8.12 *Query* 3 Model 2 117
5.8.13 *Query* 3 Model 3 118
5.8.14 *Query* 3 Model 4 119
5.8.15 *Query* 3 Model 5 120
BAB 6 PENGUJIAN DAN ANALISIS 122
6.1 Pengujian *Query* 1 122
6.1.1 *Query Affinity Cost* 122
6.1.2 *Query Localization Cost* 124
6.1.3 *Localization Processing Cost* 125
6.2 Pengujian Query 2 126
6.2.1 *Query Affinity Cost* 127
6.2.2 *Query Localization Cost* 128
6.2.3 *Localization Processing Cost* 130
6.3 Pengujian Query 3 131

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6.3.1 *Query Affinity Cost* 132
6.3.2 *Query Localization Cost* 133
6.3.3 *Localization Processing Cost* 135
6.4 Pengujian Metode *TLBO Query* 1 137
6.4.1 Fase *Teacher* 137
6.4.2 Fase *Learner* 139
6.5 Pengujian Metode *TLBO Query* 2 140
6.5.1 Fase *Teacher* 140
6.5.2 Fase *Learner* 142
6.6 Pengujian Metode *TLBO Query* 3 144
6.6.1 Fase *Teacher* 144
6.6.2 Fase *Learner* 146
6.7 Analisis Hasil 147
BAB 7 Penutup 151
7.1 Kesimpulan 151
7.2 Saran 152
DAFTAR PUSTAKA 153
LAMPIRAN A PERHITUNGAN *Query Affinity Cost (QAC)* 155
A.1 *QUERY* 1 155
A.2 *QUERY* 2 156
A.3 *QUERY* 3 158
LAMPIRAN B PERHITUNGAN *Query Localization Cost (QLC)* 161
B.1 *QUERY* 1 161
B.2 *QUERY* 2 179
B.3 *QUERY* 3 198
LAMPIRAN C PERHITUNGAN *Localization Processing Cost (LPC)* 233
C.1 *QUERY* 1 233
C.1.1 *Relation Processing Cost* 233
C.1.2 *LPC* untuk *remote site* yang digunakan pada *query plan* 237
C.1.3 *LPC* untuk *control site* yang digunakan pada *query plan* 239
C.1.4 *Local Processing Cost (LPC)* 241
C.2 *QUERY* 2 242

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

C.2.1 *Relation Processing Cost* 242
C.2.2 *LPC* untuk *remote site* yang digunakan pada *query plan* 247
C.2.3 *LPC* untuk *control site* yang digunakan pada *query plan* 249
C.2.4 *Local Processing Cost (LPC)* 251
C.3 *QUERY* 3 252
C.3.1 *Relation Processing Cost* 252
C.3.2 *LPC* untuk *remote site* yang digunakan pada *query plan* 258
C.3.3 *LPC* untuk *control site* yang digunakan pada *query plan* 260
C.3.4 *Local Processing Cost (LPC)* 263
LAMPIRAN D PERHITUNGAN TLBO 265
D.1 *QUERY* 1 265
D.1.1 TEACHER PHASE 266
D.1.2 LEARNER PHASE 267
D.2 *QUERY* 2 270
D.2.1 TEACHER PHASE 271
D.2.2 LEARNER PHASE 272
D.3 *QUERY* 3 279
D.3.1 TEACHER PHASE 280
D.3.2 LEARNER PHASE 281

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

Tabel 4.1 Jumlah Data Mahasiswa S1 aktif sampai dengan Desember 2015 39
Tabel 4.2 Jumlah data mahasiswa aktif S1 hasil fragmentasi 40
Tabel 4.3 Matrik relasi dan situs 56
Tabel 4.4 *Query Plan Query* 1 57
Tabel 4.5 *Query Plan Query* 2 57
Tabel 4.6 *Query Plan Query* 3 57
Tabel 5.1 Daftar IP *Adrress Virtual Server* 60
Tabel 5.2 Persebaran Tabel pada setiap *Site* 61
Tabel 5.3 Formula Excel 85
Tabel 6.1 Tabel Nilai *QAC Query* 1 122
Tabel 6.2 Tabel Nilai *QLC Query* 1 124
Tabel 6.3 Tabel Nilai *LPC Query* 1 125
Tabel 6.4 Tabel Nilai *QAC Query* 2 127
Tabel 6.5 Tabel Nilai *QLC Query* 2 129
Tabel 6.6 Tabel Nilai *LPC Query* 2 130
Tabel 6.7 Tabel Nilai *QAC Query* 3 132
Tabel 6.8 Tabel Nilai *QLC Query* 3 134
Tabel 6.9 Tabel Nilai *LPC Query* 3 135
Tabel 6.10 Tabel Daftar Biaya Setiap *Query Plan* 137
Tabel 6.11 Tabel Nilai Fitness dan Nilai Mean Query Plan 137
Tabel 6.12 Tabel *Teachng Factor* atau *Difference Factor* 138
Tabel 6.13 Tabel *Update Population* 01 Berdasar *Tf* 138
Tabel 6.14 Tabel Nilai *Fitness Population* 01 139
Tabel 6.15 Tabel *Update Population* 02 Berdasar Interaksi Antar *Learner* 139
Tabel 6.16 Tabel Perangkingan *Query Plan* 140
Tabel 6.17 Tabel Daftar Biaya Setiap *Query Plan* 141
Tabel 6.18 Tabel Nilai Fitness dan Nilai Mean Query Plan 141
Tabel 6.19 Tabel *Teachng Factor* atau *Difference Factor* 142
Tabel 6.20 Tabel *Update Population* 01 Berdasar *Tf* 142
Tabel 6.21 Tabel Nilai *Fitness Population* 01 142

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 6.22 Tabel *Update Population* 02 Berdasar Interaksi Antar *Learner* 143
Tabel 6.23 Tabel Perengkingan *Query Plan* 143
Tabel 6.24 Tabel Daftar Biaya Setiap *Query Plan*..... 144
Tabel 6.25 Tabel Nilai *Fitness* dan Nilai *Mean Query Plan*..... 145
Tabel 6.26 Tabel *Teachng Factor* atau *Difference Factor* 145
Tabel 6.27 Tabel *Update Population* 01 Berdasar *Tf* 145
Tabel 6.28 Tabel Nilai *Fitness Population* 01 146
Tabel 6.29 Tabel *Update Population* 02 Berdasar Interaksi Antar *Learner* 146
Tabel 6.30 Tabel Perangkingan *Query Plan* 147
Tabel 7.1 Klausa WHERE pada *Query* 1 Model 1 161
Tabel 7.2 Klausa WHERE pada *Query* 1 Model 2 165
Tabel 7.3 Klausa WHERE pada *Query* 1 Model 3 169
Tabel 7.4 Klausa WHERE pada *Query* 1 Model 4 172
Tabel 7.5 Klausa WHERE pada *Query* 1 Model 5 176
Tabel 7.6 Klausa WHERE pada *Query* 2 Model 1 180
Tabel 7.7 Klausa WHERE pada *Query* 2 Model 2 183
Tabel 7.8 Klausa WHERE pada *Query* 2 Model 3 187
Tabel 7.9 Klausa WHERE pada *Query* 2 Model 4 191
Tabel 7.10 Klausa WHERE pada *Query* 2 Model 5 194
Tabel 7.11 Klausa WHERE pada *Query* 3 Model 1 198
Tabel 7.12 Klausa WHERE pada *Query* 3 Model 2 205
Tabel 7.13 Klausa WHERE pada *Query* 3 Model 3 212
Tabel 7.14 Klausa WHERE pada *Query* 3 Model 4 219
Tabel 7.15 Klausa WHERE pada *Query* 3 Model 5 226
Tabel 7.16 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 1..... 233
Tabel 7.17 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 2..... 234
Tabel 7.18 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 3..... 235
Tabel 7.19 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 4..... 236
Tabel 7.20 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 5..... 236
Tabel 7.21 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 1..... 242
Tabel 7.22 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 2..... 243
Tabel 7.23 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 3..... 244

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 7.24 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 4..... 245
Tabel 7.25 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 5..... 246
Tabel 7.26 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 1..... 252
Tabel 7.27 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 2..... 253
Tabel 7.28 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 3..... 255
Tabel 7.29 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 4..... 256
Tabel 7.30 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 5..... 257
Tabel 7.31 Tabel Nilai *Fitness* dan Nilai *Mean Query Plan*..... 265
Tabel 7.32 Tabel *Update Population* 01 berdasar *Tf* 266
Tabel 7.33 Tabel Nilai *Fitness Population* 01 267
Tabel 7.34 Tabel *Update Population* 02 berdasar Interaksi *Learner* 269
Tabel 7.35 Tabel Nilai *Fitness* dan Nilai *Mean Query Plan*..... 270
Tabel 7.36 Tabel *Update Population* 01 berdasar *Tf* 272
Tabel 7.37 Tabel Nilai *Fitness Population* 01 272
Tabel 7.38 Tabel *Update Population* 02 berdasar Interaksi *Learner* 279
Tabel 7.39 Tabel Nilai *Fitness* dan Nilai *Mean Query Plan*..... 279
Tabel 7.40 Tabel *Update Population* 01 berdasar *Tf* 281
Tabel 7.41 Tabel Nilai *Fitness Population* 01 281
Tabel 7.42 Tabel *Update Population* 02 berdasar Interaksi *Learner* 286

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

Gambar 2.1 Skema Basis Data Terdistribusi 7
Gambar 2.2 Arsitektur Basis Data erdistribusi 8
Gambar 2.3 Contoh Relasi *Database* 9
Gambar 2.4 Hasil dari Fragmentasi Horizontal 10
Gambar 2.5 Hasil dari Fragmentasi Vertikal 10
Gambar 2.6 Operasi SELECT 11
Gambar 2.7 Contoh Operasi SELECT 12
Gambar 2.8 Operasi PROJECT 12
Gambar 2.9 Operasi *Cartesian Product* 13
Gambar 2.10 Contoh dari *Cartesian Product* 13
Gambar 2.11 Operasi UNION 14
Gambar 2.12 Hasil dari operasi (RESULT) = RESULT1 u RESULT2 14
Gambar 2.13 Kolom pada tabel EMPLOYEE 15
Gambar 2.14 Formula Perhitungan Bonus 15
Gambar 2.15 Operasi RANAME 15
Gambar 2.16 Contoh Operasi Agregasi 16
Gambar 2.17 Contoh Kalkulus Relasional Tupel 16
Gambar 2.18 Contoh Kalkulus Relasional Domain 17
Gambar 2.19 Proses dari query terdistribusi 18
Gambar 2.20 Optimasi Query 20
Gambar 2.21 Alur Algoritme *TLBO* 23
Gambar 2.22 Tahap pada fase *learner* 24
Gambar 3.1 Alur Penelitian ..... **Error! Bookmark not defined.**
Gambar 3.2 Skema *TLBO* untuk menentukan *Query Plan* yang Optimal pada Basis Data Terdistribusi ..... **Error! Bookmark not defined.**
Gambar 4.1 Topologi Mesh 33
Gambar 4.2 Skema RDBMS SIAKAD 35
Gambar 4.3 Skema Mahasiswa 36
Gambar 4.4 Skema Mahasiswa Biodata 36
Gambar 4.5 Skema Orang Tua Mahasiswa 37

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 4.6 Skema KHS Mahasiswa 37
Gambar 4.7 Skema Keuangan Mahasiswa 38
Gambar 4.8 Basis Data Terdistribusi 41
Gambar 5.1 *Source Create Table* FAKULTAS 62
Gambar 5.2 *Source Create Table* JURUSAN 63
Gambar 5.3 *Source Create Table* PROG_STUDI 64
Gambar 5.4 *Source Create Table* JENJANG 65
Gambar 5.5 *Source Create Table* SELEKSI 66
Gambar 5.6 *Source Create Table* MAHASISWA 67
Gambar 5.7 *Source Create Table* NEGARA 68
Gambar 5.8 *Source Create Table* PROPINSI 69
Gambar 5.9 *Source Create Table* KOTA 70
Gambar 5.10 *Source Create Table* SEKOLAH 71
Gambar 5.11 *Source Create Table* MAHASISWA_BIODATA 72
Gambar 5.12 *Source Create Table* MAHASISWA 74
Gambar 5.13 *Source Create Table* PEKERJAAN 75
Gambar 5.14 *Source Create Table* MAHASISWA_ORTU 76
Gambar 5.15 *Source Create Table* KHS 78
Gambar 5.16 *Source Create Table* NILAI 79
Gambar 5.17 *Source Create Table* MATA_KULIAH 80
Gambar 5.18 *Source Create Table Item* 82
Gambar 5.19 *Source Create Table Transaksi* 82
Gambar 5.20 *Source Create Table* Keuangan 83
Gambar 5.21 *Source Create Table* MASUK 85
Gambar 5.22 *Source Create Table* K_TEMP 86
Gambar 5.23 *Source Create Procedure* PROC_TS 87
Gambar 5.24 *Source Call Procedure* PROC_TS 88
Gambar 5.25 *Source Drop Table* 88
Gambar 5.26 *Source update dbm cfg* 89
Gambar 5.27 *Source catalog node* 89
Gambar 5.28 *Source catalog database* 90
Gambar 5.29 *Source create wrapper* 91

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 5.30 *Source create server* 91
Gambar 5.31 *Source user mapping* 92
Gambar 5.32 *Source create nick name* 93
Gambar 5.33 Perintah *Load* Data *Site* 1 96
Gambar 5.34 Perntah *Load* Data *Site* 2 98
Gambar 5.35 Perntah *Load* Data *Site* 3 100
Gambar 5.36 Perntah *Load* Data *Site* 4 102
Gambar 5.37 Perntah *Load* Data *Site* 5 105
Gambar 5.38 *Query* 1 Model 1 107
Gambar 5.39 *Query* 1 Model 2 108
Gambar 5.40 *Query* 1 Model 3 109
Gambar 5.41 *Query* 1 Model 4 110
Gambar 5.42 *Query* 1 Model 5 111
Gambar 5.43 *Query* 2 Model 1 112
Gambar 5.44 *Query* 2 Model 2 113
Gambar 5.45 *Query* 2 Model 3 114
Gambar 5.46 *Query* 2 Model 4 115
Gambar 5.47 *Query* 2 Model 5 116
Gambar 5.48 *Query* 3 Model 1 117
Gambar 5.49 *Query* 3 Model 2 118
Gambar 5.50 *Query* 3 Model 3 119
Gambar 5.51 *Query* 3 Model 4 120
Gambar 5.52 *Query* 3 Model 5 121

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR LAMPIRAN

LAMPIRAN A PERHITUNGAN *Query Affinity Cost (QAC)* 155
A.1 *QUERY* 1 155
A.2 *QUERY* 2 156
A.3 *QUERY* 3 158
LAMPIRAN B PERHITUNGAN *Query Localization Cost (QLC)* 161
B.1 *QUERY* 1 161
B.2 *QUERY* 2 179
B.3 *QUERY* 3 198
LAMPIRAN C PERHITUNGAN *Localization Processing Cost (LPC)* 233
C.1 *QUERY* 1 233
C.1.1 *Relation Processing Cost* 233
C.1.2 *LPC* untuk *remote site* yang digunakan pada *query plan* 237
C.1.3 *LPC* untuk *control site* yang digunakan pada *query plan* 239
C.1.4 *Local Processing Cost (LPC)* 241
C.2 *QUERY* 2 242
C.2.1 *Relation Processing Cost* 242
C.2.2 *LPC* untuk *remote site* yang digunakan pada *query plan* 247
C.2.3 *LPC* untuk *control site* yang digunakan pada *query plan* 249
C.2.4 *Local Processing Cost (LPC)* 251
C.3 *QUERY* 3 252
C.3.1 *Relation Processing Cost* 252
C.3.2 *LPC* untuk *remote site* yang digunakan pada *query plan* 258
C.3.3 *LPC* untuk *control site* yang digunakan pada *query plan* 260
C.3.4 *Local Processing Cost (LPC)* 263
LAMPIRAN D PERHITUNGAN TLBO 265
D.1 *QUERY* 1 265
D.1.1 TEACHER PHASE 266
D.1.2 LEARNER PHASE 267
D.2 *QUERY* 2 270
D.2.1 TEACHER PHASE 271

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

D.2.2 LEARNER PHASE 272
D.3 *QUERY* 3 279
D.3.1 TEACHER PHASE 280
D.3.2 LEARNER PHASE 281

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 1 PENDAHULUAN

1.1 Latar belakang

Basis data terdistribusi secara logik terhubung dari kumpulan-kumpulan data yang digunakan bersama-sama, dan didistribusikan melalui suatu jaringan komputer (Valduriez et al, 2011). *Distributed Database Management System* (DDBMS) memiliki satu logikal basis data yang dibagi ke dalam beberapa fragmen, dimana setiap fragmen disimpan pada satu atau lebih komputer di bawah kontrol dari *Database Management System* (DBMS) yang terpisah, dengan mengkoneksikan komputer menggunakan jaringan komunikasi (Mishra et al, 2015). Penggunaan basis data terdistribusi menawarkan peningkatan kehandalan, peningkatan kinerja, peningkatan ketersediaan data dan kemudahan perkembangan modular (Valduriez et al, 2011). Seperti halnya basis data terpusat, basis data terdistribusi harus menyediakan *user interface* yang efisien, dimana menyembunyikan seluruh detail pokok pada distribusi data pada *database* terdistribusi dari *user* (Ghaemi et al, 2008). Penggunaan *query relational* mengizinkan *user* untuk mendeskripsikan data yang dibutuhkan tanpa tahu dimana data secara fisik disimpan (Victor, 1981). Data yang diterima dari *site* yang berbeda di *database* terdistribusi dikenal sebagai *Distributed Query Processing* (DQP) (Ghaemi et al, 2008).

Query processing memiliki lebih banyak kesulitan pada lingkungan terdistribusi daripada lingkungan terpusat karena terdapat banyak parameter yang mempengaruhi kinerja *distributed query* seperti pengiriman data, relasi mungkin direplikasi dan/atau difragmentasi serta *query* dieksekusi lebih dari satu situs lokal dan mengingat banyaknya situs yang diakses, respon *time query* akan menjadi sangat tinggi (Ghaemi et al, 2008). Proses *query* pada basis data terdistribusi berlangsung pada beberapa lapisan, yakni 3 lapisan di situs utama dan 1 lapisan di situs lokal yang meliputi *query decompsition*, *data localization*, *global query optimization* pada situs utama dan *distributed query execution* pada situs lokal (Valduriez et al, 2011). Sehingga performa dari sebuah DDBMS ditentukan oleh kemampuannya dalam memproses *query* dengan efisien. Biaya utama terjadi ketika memproses *query* pada DDBMS adalah CPU/IO dan biaya komunikasi, dimana biaya komunikasi benar-benar dipertimbangkan sebagai faktor paling penting yang berdampak pada biaya pemrosesan *query* (Kumar et al, 2011). Pendistribusian data menimbulkan tantangan pada *query processing*, dan untuk sebuah *query* terdapat banyak *Query Equivalent Plans* (QEP's) yang mungkin, alternatif-alternatif ini *equivalent* pada hasil yang diterima (Misrha, 2015). Salah satu cara penyusunan QEP dapat dilakukan dengan melibatkan jumlah *site* yang berbeda atau konsentrasi relasi yang berbeda pada setiap *site*. Tujuan QDP adalah untuk mengeksekusi suatu *query* seefisien mungkin agar meminimalkan respon *time* dimana pengguna harus menunggu hasil ataupun waktu *delay* dari program aplikasi (Ghaemi et al, 2008).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Query processing dan optimasi menjadi salah satu faktor utama untuk meningkatkan kinerja *query* pada basis data terdistribusi (Seed, 2013). Dengan demikian strategi DQP bertujuan untuk menghasilkan rencana pemrosesan *query* yang mengurangi jumlah transfer data antara situs dan mengurangi waktu respon *query* terdistribusi (Ghaemi et al, 2008). Proses penentuan *query* yang optimal dapat dilakukan oleh DBMS itu sendiri maupun dengan algoritma (metode) optimasi yang tersedia, dimana *query processing* telah berhubungan dengan banyak area penelitian basis data sehingga dampak pada pemrosesan *query* yang efisien menjadi terlihat jelas pada meningkatnya jumlah metode optimasi (Mishra, 2015).

Pada umumnya algoitma optimasi dapat dibagi menjadi dua kategori yakni algoritma deterministik dan algoritma stokastik, detrministik algoritma digunakan untuk menyelesaikan permasalahan dengan data yang jelas dan dapat diterapkan dengan menerapkan perhitungan statistik sedangkan algoritma *stochastic* menyelesaikan masalah yang memiliki data random (Kumar, 2015). Untuk algoritma *stochastic* dibagi menjadu dua, heuristik yang bersifat *probelm dependent* atau bergantung pada jenis permasalahannya dan meta-heuristik bersifat *problem independent* yang dapat diterapkan dalam berbagai jenis optimasi (Repinsek et al, 2012).

Teacher Learner Based Optimization (TLBO) merupakan salah satu metode metaheuristik yang baru-baru ini diusulkan oleh (Rao et al, 2011), TLBO adalah sebuah algoritma optimasi berbasis populasi yang diinspirasi dari penyampaian pengetahuan di lingkungan kelas, dimana *learner* pertama kali memperoleh pengetahuan dari seorang *teacher* dan kemudian dari interaksi dengan teman sekelasnya. *Teacher* secara umum dianggap sebagai orang berpengalaman tinggi yang membagi pengetahuannya kepada *learner* dimana kualitas dari seorang *teacher* berdampak pada hasil dari *learner* (Rao et al, 2012b). Hal itu jelas *teacher* yang baik melatih *learner* sehingga mereka dapat memiliki hasil yang lebih baik pada nilai mereka ditambah lagi *learner* juga belajar dari interaksi antar *learner* sendiri, dimana juga membantu meningkatkan hasil mereka (Rao et al, 2012b). Pada metode TLBO populasi diibaratkan sebagai sebuah kelompok dari *learner* atau sebuah kelas dari *learner* dimana pada algoritma optimasi, populasi terdiri dari desan variabel yang berbeda (Rao et al, 2012b). Pada metode TLBO, desain variable yang berbeda dianalogikan sebagai subjek yang berbeda yang ditawarkan pada learner dan hasil hasil dari *learner* dianalogikan sebagai *fitness*, seperti pada teknik optimasi berbasis populasi lainnya (Rao et al, 2012a). TLBO hanya membutuhkan parameter kontrol yang umum seperti ukuran populasi dan jumlah generasi untuk dapat diproses, metode TLBO dapat dikatakan sebagai sebuah algoritma yang spesifik dengan sedikit parameter (Rao et al, 2012a).

Salah satu parameter pendukung metode TLBO sebagai parameter heuristik, proses optimalisasi pada *query* plan berdasar pada *cost* model yang berbeda, *cost* model tersebut diasumsikan sebagai atribut pada *query processing* (Mishra, 2012). Salah satu cara umum untuk mengidentifikasi *query* yang optimal adalah dengan melakukan *scoring* pada semua objek *database* pada beberapa fungsi

scoring. Pada metode TLBO ini *cost* model yang digunakan sebagai atribut *scoring* adalah *Querry Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC) (Mishra 2015).

Salah satu bagian penting dalam pemrosesan query terdistribusi adalah optimasi *query*, sehingga pada penelitian ini akan dilakukan perancangan *query* yang optimal pada *database* terdistribusi dengan strategi penerapan algoritma TLBO. Dengan tujuan ingin menerapkan algoritma baru yang memiliki kelebihan yaitu parameter yang didefinisikan hanya parameter secara umum dan tidak membutuhkan parameter khusus untuk kinerja optimal dari algoritma dan juga perubahan parameter tidak mempengaruhi tingkat efektifitas algoritma.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka rumusan masalah dari penelitian ini adalah:

1. Bagaimana mengimplementasikan metode TLBO untuk menentukan *query* yang optimal sebagai metode *query optimization* pada basis data terdistribusi?
2. Apakah perbedaan jumlah *site* dan konsentrasi relasi pada populasi *query plan* berpengaruh pada hasil *query* yang optimal berdasar hasil *query optimization* dengan metode TLBO?
3. Bagaimana pengaruh dari tiga parameter heuristik QAC, QLC dan LPC dalam *query optimization* dengan metode TLBO pada penentuan query yang optimal pada basis data terdistribusi?
4. Bagaimana metode TLBO dapat diterapkan sebagai metode dalam *query optimization* pada basis data terdistribusi?

1.3 Tujuan

Berdasarkan rumusan masalah di atas, adapun tujuan penelitian ini adalah:

1. Mengetahui bagaimana mengimplementasikan metode TLBO untuk menentukan *query* yang optimal sebagai metode *query optimization* pada basis data terdistribusi.
2. Mengetahui apakah perbedaan jumlah *site* dan konsentrasi relasi pada populasi *query plan* berpengaruh pada hasil *query* yang optimal berdasar hasil *query optimization* dengan metode TLBO.
3. Mengetahui bagaimana pengaruh dari tiga parameter *heuristic* QAC, QLC dan LPC dalam *query optimization* dengan metode TLBO pada penentuan query yang optimal pada basis data terdistribusi.
4. Mengetahui bagaimana metode TLBO dapat diterapkan sebagai metode dalam *query optimization* pada basis data terdistribusi.

1.4 Manfaat

Dengan adanya penelitian ini diharapkan dapat memberikan manfaat mengenai algoritma optimasi baru yang dapat menjadi referensi penentuan *query*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

yang optimal pada basis data terdistribusi. Sehingga dapat merencanakan *query* yang optimal dalam pemanggilan data pada *database* terdistribusi.

1.5 Batasan masalah

Berdasarkan uraian masalah yang diangkat dalam penelitian, maka ruang ringkup penelitian ini adalah sebagai berikut:

1. Penelitian ini berfokus pada implementasi metode TLBO pada basis data terdistribusi untuk penentuan *query*yan optimal.
2. Pengolahan data dan proses *query* dilakukan pada basis data terdistribusi dengan fragmentasi data secara vertikal.
3. Data yang digunakan pada penelitian ini merupakan data *dummy* dari data akademik mahasiswa Universitas Brawijaya yang dibuat sesuai dengan kebutuhan penelitian.
4. Rancangan basis data yang digunakan dalam penerapan metode TLBO didesain dengan mengacu pada skema basis data akademik mahasiswa Universitas Brawijaya.
5. Topologi yang digunakan adalah topologi mesh dan dengan *transfer rate* yang sama.

1.6 Sistematika pembahasan

Sistematika pembahasan ini dimaksud untuk memberikan gambaran mengenai bab-bab yang akan disusun dalam laporan skripsi, yaitu:

BAB I PENDAHULUAN

Bab ini merupakan pendahuluan yang mencakup uraian tentang latar belakang, rumusan masalah, tujuan, manfaat, dan sistematika pembahasan.

BAB 2 LANDASAN TEORI

Bab ini berisi teori-teori pendukung mengenai basis data, database ralational, metode optimasi *query*, dan metode *Teacher-Learner Based Optimization* yang meliputi kajian pustaka, basis data, aljabar relasional, operasi aljabar relasional, kalkulus relasional, *distributed query processing*, *query optimization*, algoritma optimasi dan *Teacher-Learner Based Optimization*.

BAB 3 METODOLOGI

Bab ini menguraikan secara rinci metode pengumpulan data serta metode pembangunan sistem yang digunakan dalam menganalisis , merancang dan mengimpelementasikan sistem yang meliputi studi literatur, observasi, desain basis data terdistribusi, perancangan populasi *query plan*, implementasi basis data terdistribusi, implementasi populasi *query plan*, pengujian populasi qu*ery plan* dan pengujian *Teacher Learner Based Optimization*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 4 PERANCANGAN

Bab ini menjelaskan mengenai desain atau perancangan dari bentuk implementasi penelitian yang meliputi perancangan jaringan komputer, basis data, pemilihan *query* dan penerapan *Teacher Learner Based Optimization*.

BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang bentuk implementasi dari rancangan yang telah dibangun pada bab 4. Pada bab ini menjelaskan mengenai implementasi media *virtual server*, basis data terdistribusi hingga model-model *query* untuk saling dibandingkan yang meliputi lingkungan perangkat lunak, lingkungan perangkat keras, implementasi *virtual server (site*), implementasi basis data, implementasi *import* data, dan implementasi model akses *query*.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan mengenai pengujian dan analisis hasil dari eksekusi *query*. Proses analisis yang dilakukan adalah dengan melakukan perbandingan biaya yang dibutuhkan oleh sebuah *query* untuk menghasilkan data yang diinginkan pada basis data terdistribusi yang telah tebentuk yang meliputi pengujian *query* 1, pengujian *query* 2, pengujian *query* 3, pengujian metode *Teacher Learner Based Optimization query* 1, pengujian *Teacher Learner Based Optimization query* 2, pengujian *Teacher Learner Based Optimization query* 3 dan analisis hasil.

BAB 7 PENUTUP

Bab ini menjelaskan secara singkat tentang kesimpulan dan saran yang dapat digunakan dalam mendesain pembagian data dan mengoptimasi pemanggilan data pada basis data terdistribusi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan Penentuan *Query* Optimal Pada Basis Data Terdistribusi Menggunakan Metode *Teacher Learner Based Optimization* (TLBO).

2.1 Kajian Pustaka

Pada penelitian sebelumnya yang dilakukan oleh Vikash Mishra dan Vikhram Singh (2015) yakni "Generating Optimal Query Plans for Distributed *Query Processing* using *Tacher Learner Based Optimization*" yang dijadikan sebagai acuan dalam penelitian ini. Metode *Teacher-Learner Based Optimization* merupakan metode yang mengacu pada proses pembelajaran antar guru dengan muridnya dimana semakin optimal guru dalam mengajar maka akan menghasilkan murid yang berkualitas. Tetapi tidak semua murid yang telah menerima ilmu dari guru akan menyerap ilmunya secara maksimal, tentu saja hanya satu orang yang mampu menerima dengan sangat maksimal dan paling berkualitas. Begitu juga dengan *query* yang digunakan untuk akses data, dari beberapa rancangan *query* yang akan dibuat tentu terdapat satu yang memiliki biaya akses data yang paling kecil sehingga mampu dieksekusi dengan sangat optimal serta menghemat biaya dan sumberdaya.

2.2 Basis Data

Basis data adalah kumpulan data, dengan data berarti fakta yang kita ketahui dapat direkam dan memiliki makna yang tersirat (Elmasri et al, 2010). Sebagai contoh, perhatikan nama-nama, nomor telepon, dan alamat rumah dari orang yang kita kenal mungkin telah dicatat dalam sebuah buku alamat yang diindeks atau mungkin telah disimpan pada hard drive menggunakan PC dan perangkat lunak seperti Micrsoft Acces atau Excel, pengumpulan dari data-data yang saling terkait secara implisit adalah merupakan basis data (Elmasri et al, 2010).

Basis data dapat dihasilkan dan dikelola secara manual atau mungkin terkomputerisasi (Elmasri et al, 2010). Sebuah basis data yang terkomputerisasi dapat dibuat dan dikelola baik oleh kelompok program aplikasi yang ditulis secara khusus untuk tugas itu atau oleh DBMS (Elmasri et al, 2010). Sebuah DBMS adalah kumpulan program yang memungkinkan pengguna untuk membuat dan memelihara *database*, DBMS adalah perangkat lunak sistem secara umum bertujuan untuk memfasilitasi proses mendefinisikan, membangun, memanipulasi, dan berbagi *database* antara berbagai pengguna dan aplikasi (Elmasri et al, 2010).

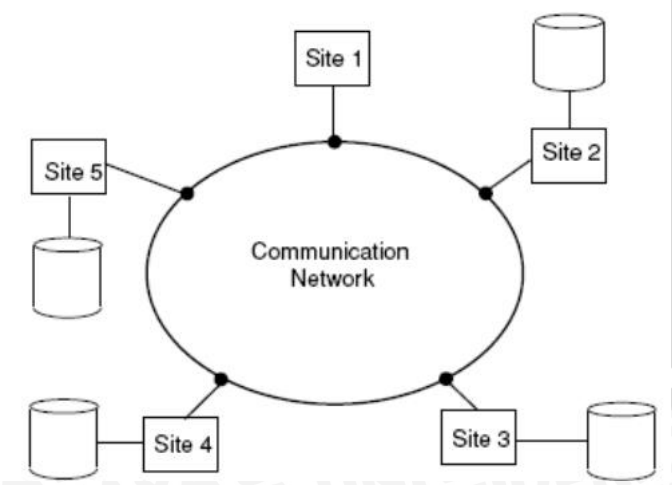
2.2.1 Basis Data Terdistribusi

Basis data terdistribusi dihasilkan dari penggabungan dua teknologi yaitu teknologi basis data dan jaringan dan teknologi komunikasi data (Elmasri et al, 2010). Jaringan komputer memungkinkan pemrosesan data secara terdistribusi,

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

database terdistribusi memungkinkan integrasi informasi dan pengolahannya dengan aplikasi yang mungkin terpusat atau didistribusikan (Elmasri et al, 2010). Apa yang didistribusikan adalah proses *logic*, fungsi, data dan kontrol dimana kebutuhan data yang terdistribusi lebih cocok untuk memenuhi kebutuhan organisasi saat ini dalam perusahaan yang terdistribusi secara luas dan sistem tersebut lebih handal dan responsif (Valduriez et al, 2011). Memiliki data yang didistribusikan di beberapa situs sangat menguntungkan untuk organisasi besar yang memiliki kantor dan tenaga kerja mereka yang didistribusikan di seluruh dunia, serta menawarkan akses ke data umum dari beberapa lokasi, memiliki data yang terdistribusi dapat menawarkan manfaat lain termasuk meningkatkan ketersediaan data (Taylor, 2010).

Kehandalan dan ketersediaan adalah dua potensi keuntungan yang paling umum untuk basis data terdistribusi. Kehandalan secara luas didefinisikan sebagai kemungkinan bahwa sistem sedang berjalan (tidak *down*) pada titik waktu tertentu, sedangkan ketersediaan adalah kemungkinan bahwa sistem ini terus tersedia selama interval waktu tertentu (Elmasri et al, 2010). Ketika data dan perangkat lunak DDBMS didistribusikan ke beberapa situs, satu situs mungkin gagal sementara situs lain terus beroperasi, hanya data dan perangkat lunak yang ada di situs gagal yang tidak dapat diakses (Elmasri et al, 2010). Hal ini meningkatkan baik kehandalan dan ketersediaan sedangkan dalam sistem terpusat, kegagalan di satu situs membuat seluruh sistem tidak tersedia untuk semua pengguna (Elmasri et al, 2010). Dalam sebuah basis data terdistribusi, beberapa data mungkin tidak terjangkau, namun pengguna masih dapat mengakses bagian lain dari basis data jika data di situs gagal telah direplikasi di situs lain sebelum kegagalan, maka pengguna tidak akan terpengaruh sama sekali (Elmasri et al, 2010). Lingkungan basis data terdistribusi ditunjukkan dalam gambar 2.1.

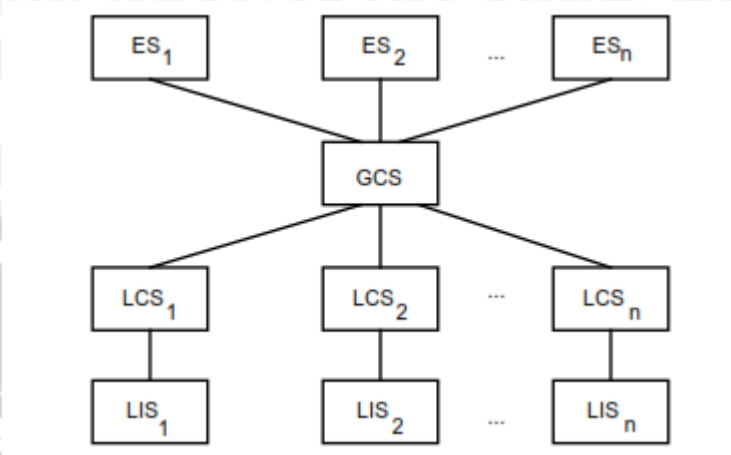


Gambar 2.1 Skema Basis Data Terdistribusi

Sumber:Valduriez et al (2011)

2.2.2 Arsiektur Basis Data Terdistribusi

Terdapat beberapa arsitektur dalam basis data terdistribusi yang dapat digunakan, diantaranya adalah arsitektur *peer to peer*. Arsitektur *peer to peer* dapat dilihat dalam gambar 2.2, dimana terdapat beberapa komponen penyusun yang memiliki tugas masing-masing.



Gambar 2.2 Arsitektur Basis Data erdistribusi

Sumber:Valduriez et al (2011)

Dalam gambar 2.2 merupakan arsitektur *peer to peer* dimana setiap situs memiliki fungsi yang sama. Transparansi lokasi dan replikasi didukung dengan mendefinisikan lokal dan global konseptual skema, transparansi jaringan juga didukung dengan pendefinisian global konseptual skema (Valduriez et al, 2011). Terdapat 4 bagian penyusun yakni *Local Internal Schema* (LIS), *Global Conceptual Schema* (LCS), *Local Conceptal Schema* (GCS) dan *External Schemas* (ES). Berikut merupakan penjelasan dari setiap bagian :

1) LIS *(Local Internal Schema*)
   Bagian yang mendefinisikan *individual internal schema* pada setiap situs.
2) GCS (*Global Conceptual Schema*)
   Bagian yang menggambarkan gambaran data yang luas, disebut global karena mendefinisikan struktur data logik secara global pada semua situs yang ada.
3) LCS (*Local Conceptal Schema*)
   Merupakan layer ketiga pada arsitektur yang berfungsi untuk menangani fragmentasi dan replikasi serta pengorganisasian data logik pada setiap situs.
4) ESs (*External Schemas*)
   Bagian yang mendukung aplikasi user dan akses user pada *database*.

2.2.3 Klasifikasi Basis Data Terdistribusi

Sebuah basis data terdistribusi dapat diklasifikasikan menjadi sebagai berikut:

1. **Heterogen**

   Heterogeneus distributed database dapat diterapkan dengan menggunakan *software* DBMS yang berbeda pada setiap site (Elmasri et al, 2010). Sistem heterogen menghasilkan *site* yang individual dimana mengimplementasikan basis data dan penyatuan datanya dilakukan di tahap berikutnya. Pada sistem heterogen diperlukan penterjemah untuk mengkomunikasikan DBMS yang berbeda. Untuk menghasilkan transparansi DBMS, pengguna harus dapat menggunakan bahasa pemrogaman yang digunakan oleh DBMS pada *local site*.

2. **Homogen**

   Homogeneus distributed database diterapkan dengan menggunakan *software* DBMS yang sama pada semua *site* (Elmasri et al, 2010). Sistem homogen lebih mudah dirancang dan dikelola. Pendekatan ini memberikan perkembangan yang baik, tidak mengalami kesulitan dalam membuat sebuah *site* yang baru.

2.2.4 Fragmentasi

Fragmentasi pada basis data terdistribusi berfungsi sebagai algoritme pemecahan tabel pada setiap *site*, dengan mempartisi basis data yang dipisahkan ke dalam fragmen-fragmen, dimana setiap fragmen dialokasikan pada satu *site* (Valduriez et al, 2011). Strategi yang dapat digunakan diantaranya adalah fragmentasi horizontal dan fragmentasi vertikal (Valduriez et al, 2011).

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

Gambar 2.3 Contoh Relasi *Database*

Sumber:Valduriez et al (2011)

$PROJ_1$

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York

$PROJ_2$

PNO	PNAME	BUDGET	LOC
P3	CAD/CAM	255000	New York
P4	Maintenance	310000	Paris

Gambar 2.4 Hasil dari Fragmentasi Horizontal

Sumber:Valduriez et al (2011)

$PROJ_1$

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000

$PROJ_2$

PNO	PNAME	LOC
P1	Instrumentation	Montreal
P2	Database Develop.	New York
P3	CAD/CAM	New York
P4	Maintenance	Paris

Gambar 2.5 Hasil dari Fragmentasi Vertikal

Sumber:Valduriez et al (2011)

Fragmenasi vertikal dari sebuah relasi R menghasilkan fragmen R1, R2,…….,Rr dimana setiap fragmen mengandung sebuah subset dari R atibut yang dikenal sebagai *primary key* dari R, tujuan dari fragmentasi vertikal adalah untuk mempartisi sebuah relasi menjadi sebuah set dari relasi-relasi yang lebih kecil sehingga banyak aplikasi pengguna dapat menjalankan hanya satu fragmen setiap aplikasi (Valduriez et al, 2011). Pada konteks ini, sebuah fragmentasi yang optimal adalah satu yang memproduksi sebuah skema fragmentasi yang meminimalkan waktu eksekusi dari aplikasi pengguna yang menjalankan fragmen tersebut (Valduriez et al, 2011).

Dua tipe dari pendekatan heuristik yang ada untuk fragmentasi vertikal pada relasi global (Valduriez et al, 2011) adalah :

1. *Grouping* : dimulai dari menetapkan setiap atribut pada satu fragmen, dan pada setiap langkah, *join* beberapa dari fragmen sampai beberapa kriteria terpenuhi.
2. *Splitting* : dimulai dengan relasi dan menentukan pada manfaat partisi berdasar pada perilaku akses aplikasi pada atribut.

2.3 Aljabar Relasional

Relational Algebra (aljabar relasional) merupakan kumpulan operasi terhadap relasi dimana setiap operasi menggunakan satu atau lebih relasi untuk

menghasilkan satu relasi yang baru dan termasuk kategori prosedural dan juga menyediakan seperangkat operator untuk memanipulasi data.Terdapat lima operasi dasar dalam aljabar relasional, yaitu (Tutorials Point, 2015):

1. *Selection* (σ)
2. *Projection* (π)
3. *Cartesian – product* (X, juga disebut sebagai *cross product*)
4. *Union* (∪)

Aljabar relasional sangat penting karena beberapa alasan (Elmasri et al, 2010). Pertama, ia menyediakan dasar formal untuk operasi model relasional. Kedua, dan mungkin lebih penting, digunakan sebagai dasar untuk melaksanakan dan mengoptimalkan *query* dalam pengolahan *query* dan optimasi modul yang merupakan bagian dari Sistem Manajemen Database Relasional (RDBMSs).

2.3.1 Operasi *Selection*

Operasi SELECT digunakan untuk memilih *subset* dari tupel pada relasi yang memenuhi pilihan kondisi. Pertama mempertimbangkan operasi SELECT menjadi filter yang hanya menghasilkan tupel yang memenuhi kondisi kualifikasi (Elmasri et al, 2010). Atau, kita dapat mempertimbangkan operasi SELECT untuk membatasi tupel dalam kaitannya dengan hanya tupel yang memenuhi kondisi(Elmasri et al, 2010). Operasi SELECT juga dapat divisualisasikan sebagai partisi horizontal relasi menjadi dua set tupel dimana mereka yang memenuhi syarat dan yang dipilih, dan tupel yang tidak memenuhi kondisi dibuang (Elmasri et al, 2010). Misalnya, untuk memilih tupel EMPLOYEE dengan departemen adalah 4, atau mereka yang gajinya lebih besar dari $ 30.000, kita dapat menentukan secara individual masing-masing dua kondisi ini dengan operasi SELECT yang ditunjukkan dalam gambar 2.6 (Elmasri et al, 2010):

$$\sigma_{Dno=4}(\text{EMPLOYEE})$$
$$\sigma_{Salary>30000}(\text{EMPLOYEE})$$

Gambar 2.6 Operasi SELECT

Sumber:Elmasri et al (2010)

Pada umumnya operasi SELECT dinotasikan sebagai berikut :

$$\sigma_{<\text{selection condition}>}(R) \qquad (2.4)$$

dimana simbol σ (sigma) digunakan untuk menunjukkan operator SELECT dan *selection condition* adalah ekspresi *Boolean* (kondisi) ditentukan pada atribut relasi R. Perhatikan bahwa R umumnya ekspresi aljabar relasional hasilnya adalah sebuah relasi, sederhananya ekspresi seperti ini hanya nama dari relasi basis data (Elmasri et al, 2010). Hubungan yang dihasilkan dari operasi SELECT memiliki atribut yang sama dengan R (Elmasri et al, 2010). Contoh operasi SELECT ditunjukkan dalam gambar 2.7.

Results of SELECT and PROJECT operations. (a) $\sigma_{(Dno=4 \text{ AND } Salary>25000) \text{ OR } (Dno=5 \text{ AND } Salary>30000)}$ (EMPLOYEE). (b) $\pi_{Lname, Fname, Salary}$(EMPLOYEE). (c) $\pi_{Sex, Salary}$(EMPLOYEE).

(a)

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5

(b)

Lname	Fname	Salary
Smith	John	30000
Wong	Franklin	40000
Zelaya	Alicia	25000
Wallace	Jennifer	43000
Narayan	Ramesh	38000
English	Joyce	25000
Jabbar	Ahmad	25000
Borg	James	55000

(c)

Sex	Salary
M	30000
M	40000
F	25000
F	43000
M	38000
M	25000
M	55000

Gambar 2.7 Contoh Operasi SELECT

Sumber:Elmasri et al (2010)

2.3.2 Operasi *Projection*

Jika kita berpikir tentang hubungan sebagai sebuah tabel, operasi SELECT memilih beberapa baris data (tupel) dari tabel sementara membuang baris lainnya, di sisi lain operasi PROJECT memilih kolom tertentu dari tabel dan membuang kolom lainnya (Elmasri et al, 2010). Oleh karena itu, hasil dari operasi PROJECT dapat divisualisasikan sebagai partisi vertikal dari relasi menjadi dua relasi: satu memiliki kolom yang dibutuhkan (atribut) dan berisi hasil operasi, dan yang lainnya berisi kolom dibuang, misalnya untuk daftar nama pertama dan terakhir, dan gaji setiap karyawan, kita dapat menggunakan operasi PROJECT yang ditunjukkan dalam gambar 2.8 (Elmasri et al, 2010):

$$\pi_{Lname, Fname, Salary}(\text{EMPLOYEE})$$

Gambar 2.8 Operasi PROJECT

Sumber:Elmasri et al (2010)

Pada umumnya operasi PROJECT dinotasikan sebagai berikut :

$$\pi_{<\text{attribute list}>}(R) \qquad (2.5)$$

di mana π (pi) adalah simbol yang digunakan untuk mewakili operasi PROJECT, dan <attribute list> adalah *sublist* atribut yang diinginkan dari atribut relasi R (Elmasri et al, 2010). Sekali lagi, perhatikan bahwa secara umum R adalah ekspresi aljabar relasional yang hasilnya adalah relasi, dalam kasus yang paling sederhana adalah nama dari relasi basis data (Elmasri et al, 2010). Hasil operasi PROJECT hanya memiliki atribut yang ditentukan dalam <attribut list> dalam urutan yang sama seperti yang muncul dalam daftar (Elmasri et al, 2010).

2.3.3 Operasi *Cartesian Product*

CARTESIAN PRODUCT yang juga biasa dikenal sebagai CROSS PRODUCT atau CROSS JOIN-yang dilambangkan dengan × merupakan set operasi yang menghasilkan elemen baru dengan menggabungkan setiap anggota (tupel) dari satu relasi dengan setiap anggota (tupel) dari relasi lain (Elmasri et al, 2010). Sebagai contoh, misalkan kita ingin mengambil daftar nama-nama tanggungan masing-masing karyawan perempuan yang ditunjukkan dalam gambar 2.9 dan hasil dari operasi ditunjukkan dalam gambar 2.10.

$$\text{FEMALE_EMPS} \leftarrow \sigma_{\text{Sex='F'}}(\text{EMPLOYEE})$$
$$\text{EMPNAMES} \leftarrow \pi_{\text{Fname, Lname, Ssn}}(\text{FEMALE_EMPS})$$
$$\text{EMP_DEPENDENTS} \leftarrow \text{EMPNAMES} \times \text{DEPENDENT}$$
$$\text{ACTUAL_DEPENDENTS} \leftarrow \sigma_{\text{Ssn=Essn}}(\text{EMP_DEPENDENTS})$$
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Dependent_name}}(\text{ACTUAL_DEPENDENTS})$$

Gambar 2.9 Operasi *Cartesian Product*

Sumber:Elmasri et al (2010)

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Alicia	J	Zelaya	999887777	1968-07-19	3321Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291Berry, Bellaire, TX	F	43000	888665555	4
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

EMPNAMES

Fname	Lname	Ssn
Alicia	Zelaya	999887777
Jennifer	Wallace	987654321
Joyce	English	453453453

EMP_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	...
Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	...
Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	...
Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	...
Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	...
Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	...
Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	...
Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	...
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	...
Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...
Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	...
Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	...
Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	...
Joyce	English	453453453	333445555	Alice	F	1986-04-05	...
Joyce	English	453453453	333445555	Theodore	M	1983-10-25	...
Joyce	English	453453453	333445555	Joy	F	1958-05-03	...
Joyce	English	453453453	987654321	Abner	M	1942-02-28	...
Joyce	English	453453453	123456789	Michael	M	1988-01-04	...
Joyce	English	453453453	123456789	Alice	F	1988-12-30	...
Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	...

Gambar 2.10 Contoh dari *Cartesian Product*

Sumber:Elmasri et al (2010)

2.3.4 Operasi *Union*

Operasi UNION adalah operasi yang biasanya digunakan untuk menggabungkan elemen-elemen dari dua relasi (Elmasri et al, 2010). Misalnya, untuk mengambil nomor Jaminan Sosial dari semua karyawan yang bekerja di departemen baik 5 atau langsung mengawasi karyawan yang bekerja di departemen 5, kita dapat menggunakan operasi UNION yang ditunjukkan dalam gambar 2.11 :

$$\text{DEP5_EMPS} \leftarrow \sigma_{Dno=5}(\text{EMPLOYEE})$$
$$\text{RESULT1} \leftarrow \pi_{Ssn}(\text{DEP5_EMPS})$$
$$\text{RESULT2(Ssn)} \leftarrow \pi_{Super_ssn}(\text{DEP5_EMPS})$$
$$\text{RESULT} \leftarrow \text{RESULT1} \cup \text{RESULT2}$$

Gambar 2.11 Operasi UNION

Sumber:Elmasri et al (2010)

Hasil operasi UNION, dilambangkan dengan R ∪ S, adalah hubungan yang mencakup semua tupel yang baik dalam R atau S atau di kedua R dan S. Duplikasi tupel dieleminasi. Hasil dari operasi UNION ditunjukkan dalam gambar 2.12.

RESULT1

Ssn
123456789
333445555
666884444
453453453

RESULT2

Ssn
333445555
888665555

RESULT

Ssn
123456789
333445555
666884444
453453453
888665555

Gambar 2.12 Hasil dari operasi (RESULT) = RESULT1 u RESULT2

Sumber:Elmasri et al (2010)

2.4 Operator Aljabar Relasional

Pada bagian ini kita mendefinisikan operasi tambahan untuk mengekspresikan permintaan pada operasi basis data. Operasi ini meningkatkan kekuatan ekspresif dari aljabar relasional aslinya (Elmasri et al, 2010).

2.4.1 *Generalized Projection*

Operasi *Generalized Projection* memperluas operasi *projection* dengan memungkinkan fungsi atribut untuk dimasukkan dalam daftar *projection*. (Elmasri et al, 2010). Bentuknya umum dapat dinyatakan sebagai:

$$\pi_{F1,\, F2,\, \ldots,\, Fn}(R) \qquad (2.6)$$

di mana F1, F2, ..., Fn adalah fungsi pada atribut dalam relasi R dan mungkin melibatkan operasi aritmatika dan nilai-nilai konstan (Elmasri et al, 2010). Operasi ini sangat membantu ketika mengembangkan laporan di mana nilai-nilai yang dihitung harus diproduksi dalam kolom dari hasil *query* (Elmasri et al, 2010). Sebagai contoh, mempertimbangkan hubungan dalam gambar 2.13 :

EMPLOYEE (Ssn, Salary, Deduction, Years_service)

Gambar 2.13 Kolom pada tabel EMPLOYEE

Sumber:Elmasri et al (2010)

Sebuah laporan mungkin diperlukan untuk menunjukkan gaji bersih dengan rumus (*Salary – Deduction*), rumus perhitungan bonus dan pajak seperti yang ditunjukkan dalam gambar 2.14.

Net Salary = Salary – Deduction,
Bonus = 2000 * Years_service, and
Tax = 0.25 * Salary.

Gambar 2.14 Formula Perhitungan Bonus

Sumber:Elmasri et al (2010)

Kemudian proyeksi umum dikombinasikan dengan penggantian nama dapat ditunjukkan dalam gambar 2.15:

$$\text{REPORT} \leftarrow \rho_{(\text{Ssn, Net_salary, Bonus, Tax})}(\pi_{\text{Ssn, Salary} - \text{Deduction, } 2000 * \text{Years_service, } 0.25 * \text{Salary}}(\text{EMPLOYEE})).$$

Gambar 2.15 Operasi RANAME

Sumber:Elmasri et al (2010)

2.4.2 Agregate Function

Tipe lain dari permintaan yang tidak dapat dinyatakan dalam aljabar relasional dasar untuk menentukan fungsi agregat matematika pada basis data . Contoh fungsi tersebut termasuk mengambil gaji rata-rata atau total semua karyawan atau jumlah tupel karyawan, fungsi ini digunakan dalam *query* statistik sederhana yang merangkum informasi dari tupel basis data (Elmasri et al, 2010). Fungsi umum diterapkan untuk koleksi nilai numerik termasuk SUM, AVERAGE, MAXIMUM, dan MINIMUM. Fungsi COUNT digunakan untuk menghitung tupel atau nilai-nilai (Elmasri et al, 2010).

Kita dapat mendefinisikan operasi AGREGATE FUNCTION, menggunakan simbol $\Im$ (diucapkan F) untuk menentukan jenis permintaan sebagai berikut:

$$_{\text{<grouping attributes>}}\Im_{\text{<function list>}}(R) \qquad (2.7)$$

di mana <grouping attribute> adalah daftar atribut relasi ditentukan dalam R, dan <function list> adalah daftar pasangan (<function> <attribute>). Dalam masing-masing pasangan tersebut, <function> adalah salah satu fungsi-seperti diperkenankan SUM, AVERAGE, MAXIMUM, MINIMUM, COUNT-dan <attribute> adalah atribut dari relasi ditentukan oleh R. Contoh Operasi Agregasi ditunjukkan dalam gambar 2.16.

The aggregate function operation.

a. $\rho_{R(\text{Dno, No_of_employees, Average_sal})}({}_{\text{Dno}}\Im_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE}))$.

b. ${}_{\text{Dno}}\Im_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

c. $\Im_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

R

(a)

Dno	No_of_employees	Average_sal
5	4	33250
4	3	31000
1	1	55000

(b)

Dno	Count_ssn	Average_salary
5	4	33250
4	3	31000
1	1	55000

(c)

Count_ssn	Average_salary
8	35125

Gambar 2.16 Contoh Operasi Agregasi

Sumber:Elmasri et al (2010)

2.5 Kalkulus Relasional

Terdapat bahasa lain permintaan formal untuk model relasional yang disebut kalkulus relasional. Bagian ini memperkenalkan bahasa yang dikenal sebagai tupel kalkulus relasional, dan memperkenalkan variasi yang disebut domain kalkulus relasional (Elmasri et al, 2010). Dalam kedua variasi kalkulus relasional, kita menulis satu ekspresi deklaratif untuk menentukan permintaan pencarian; oleh karena itu, tidak ada gambaran tentang bagaimana, atau dalam rangka apa *query* tersebut dievaluasi (Elmasri et al, 2010). Ekspresi kalkulus menentukan apa yang akan diambil bukan bagaimana untuk mengambilnya, oleh karena itu kalkulus relasional dianggap bahasa *nonprocedural* berbeda dari aljabar relasional di mana kita harus menulis urutan operasi untuk menentukan permintaan pencarian dalam urutan tertentu dalam menerapkan operasi, dengan demikian dapat dianggap sebagai cara prosedural untuk menyatakan *query* (Elmasri et al, 2010). Ekspresi kalkulus dapat ditulis dalam cara yang berbeda, tetapi cara yang tertulis tidak memiliki aturan pada bagaimana *query* harus dieksekusi (Elmasri et al, 2010).

2.5.1 Kalkulus Relasional Tupel

Ekspresi umum dari tupel kalkulus relasional adalah dalam bentuk

$$\{t_1.A_j, t_2.A_k, \ldots, t_n.A_m \mid \text{COND}(t_1, t_2, \ldots, t_n, t_{n+1}, t_{n+2}, \ldots, t_{n+m})\} \quad (2.8)$$

dimana t1, t2, ..., tn, tn + 1, ..., tn + m adalah variabel tupel, setiap Ai adalah atribut dari relasi ti dan COND adalah kondisi atau formula dari tupel kalkulus relasional. Contoh dari kalkulus relasional tupel adalah sebagai berikut:

Query 1. List the name and address of all employees who work for the 'Research' department.

Q1: $\{t.\text{Fname}, t.\text{Lname}, t.\text{Address} \mid \text{EMPLOYEE}(t)$ **AND** $(\exists d)(\text{DEPARTMENT}(d)$ **AND** $d.\text{Dname}=\text{'Research'}$ **AND** $d.\text{Dnumber}=t.\text{Dno})\}$

Gambar 2.17 Contoh Kalkulus Relasional Tupel

Sumber:Elmasri et al (2010)

Satu-satunya variabel tupel bebas pada tupel ekspresi kalkulus relasional adalah tupel yang muncul di sebelah kiri bar (|), pada Q1 t adalah satu-satunya variabel bebas; itu kemudian terikat berturut-turut untuk setiap tupel (Elmasri et al, 2010). Jika tupel memenuhi kondisi yang ditentukan setelah bar di Q1, atribut Fname, lname, dan Alamat yang akan diambil untuk setiap tupel tersebut, kondisi EMPLOYEE (t) dan DEPARTMENT (d) menentukan hubungan jarak untuk t dan d dan kondisi d.Dname = 'Research' adalah kondisi seleksi dan sesuai dengan operasi SELECT di aljabar relasional, sedangkan kondisi d.Dnumber = t.Dno adalah kondisi JOIN dan mirip seperti operasi (INNER) pada operasi JOIN (Elmasri et al, 2010).

2.5.2 Kalkulus Relasional Domain

Jenis lain dari kalkulus relasional disebut domain kalkulus relasional diman domain kalkulus berbeda dari tupel kalkulus dalam jenis variabel yang digunakan dalam formula, bukan memiliki variabel rentang tupel tetapi variabel yang dimiliki adalah nilai domain atribut (Elmasri et al, 2010). Ekspresi dari kalkulus domain adalah dalam bentuk :

$$\{x_1, x_2, \ldots, x_n \mid \text{COND}(x_1, x_2, \ldots, x_n, x_{n+1}, x_{n+2}, \ldots, x_{n+m})\} \quad (2.9)$$

di mana x1, x2, ..., xn, xn + 1, xn + 2, ..., xn + m adalah variabel domain (atribut), dan COND adalah kondisi atau formula dari domain kalkulus relasional.

Q0A: $\{u, v \mid \text{EMPLOYEE}(\text{'John'},\text{'B'},\text{'Smith'},t,u,v,w,x,y,z)\}$

Query 1. Retrieve the name and address of all employees who work for the 'Research' department.

Q1: $\{q, s, v \mid (\exists z)\ (\exists l)\ (\exists m)\ (\text{EMPLOYEE}(qrstuvwxyz)\ \textbf{AND}\ \text{DEPARTMENT}(lmno)\ \textbf{AND}\ l=\text{'Research'}\ \textbf{AND}\ m=z)\}$

Gambar 2.18 Contoh Kalkulus Relasional Domain

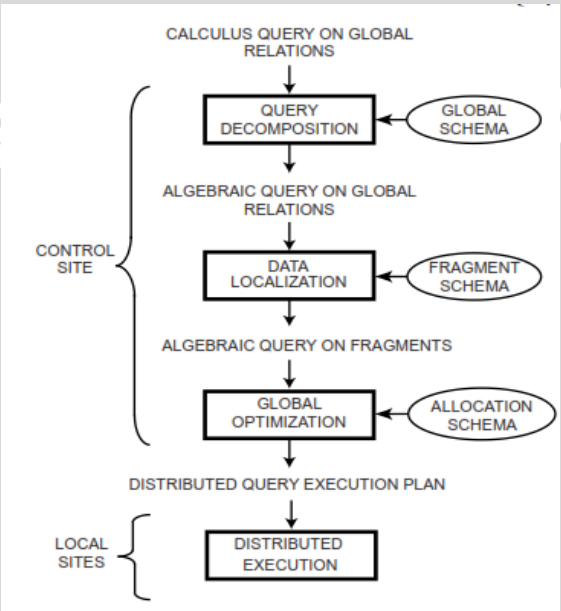
Sumber:Elmasri et al (2010)

Sebuah kondisi yang berkaitan dua variabel domain yang berkisar lebih dari atribut dari dua relasi, seperti m = z di Q1, adalah kondisi JOIN, sedangkan kondisi yang berhubungan variabel domain untuk konstan, seperti l = 'Research', adalah sebuah pilihan kondisi (Elmasri et al, 2010).

2.6 *Distributed Query Processing*

Distributed Query Processing (DQP) adalah sebuah faktor penting pada keseluruhan performansi dari sistem basis data terdistribusi (Alom, 2009). *Query processing* adalah sebuah aktifitas utama pada basis data dan hal tersebut dikontrol leh DBMS, kinerja dari sebuah DBMS ditentukan oleh kemampuannya untuk memproses *query* secara efektif dan efisien (Mishra, 2015). Dalam DQP, biaya utama yang dikeluarkan adalah CPU, I / O dan biaya komunikasi situs-situs diantaranya biaya komunikasi intra-situs adalah biaya yang dominan, biaya ini dapat dikurangi jika situs yang lebih sedikit terlibat dalam pengolahan *query* terdistribusi (Mishra, 2015). Untuk memproses permintaan didistribusikan, data

yang dibutuhkan mungkin harus diperoleh dari beberapa situs yang didistribusikan melalui jaringan komputer, selain itu sebagai jumlah situs yang berisi hubungan yang diakses oleh peningkatan permintaan, jumlah kemungkinan rencana permintaan yang berlaku juga meningkat (Mishra, 2015). Sehingga menjadi penting untuk sampai pada rencana pemrosesan *query* yang memakai biaya yang optimal untuk pemrosesan *query*, dengan demikian ruang pencarian besar yang terdiri dari semua rencana permintaan mungkin perlu dieksplorasi untuk menghitung rencana permintaan yang optimal (Mishra, 2015). Pemrosesan *query* di lingkungan seperti ini tugas yang sulit untuk *query processor*, seperti untuk permintaan pengguna ada beberapa permintaan alternatif setara yang memungkinkan, alternatif ini setara dengan ketentuan *output* (Mishra, 2015).



Gambar 2.19 Proses dari query terdistribusi

Sumber:Valduriez et al (2011)

2.6.1 *Query Decomposition*

Pada lapisan pertama ini *query* yang berbentuk *calculus* pada *global schema* akan diubah menjadi *algebraic query*, proses transformasi dari *quey* kalkulus menjadi *query* aljabar memanfaatkan data relasi yang tersimpan pada skema konseptual global (Valduriez et al, 2011). Pada layer ini juga dilakukan proses normalisasi data untuk menghilangkan adanya redundansi data dimana proses yang terjadi pada layer ini, *query* kalkulus ditulis ulang dalam bentuk normalisasi kemudian akan dibentuk *normalized query,* setelah terbentuk *query* yang benar maka akan ditransfrmasikan ke *query* aljabar (Valduriez et al, 2011).

2.6.2 Data *Localization*

Pada layer ini, *query* keluaran dari layer *query decomposition* berbentuk aljabar dialokasikan menggunakan *data distribution schema* yang ada pada *fragment shema* (Valduriez et al, 2011). Fungsi dari layer ini adalah menentukan *fragment* mana yang digunakan pada sebuah *query* dan mentransformasikan

distributed query menjadi sebuah *query* pada *fragment* (Valduriez et al, 2011). Keluaran dari proses ini adalah aljabar *query* yang terbentuk pada setiap *fragment* (Valduriez et al, 2011).

2.6.3 *Global Query Optimization*

Pada layer ini, masukan berupa *query* aljabar yang terdapat pada skema fragmen dengan tujuan dari layer ini adalah mencari strategi untuk memproses sebuah *query* dengan maksud mencari biaya yang paling kecil untuk mengoptimalkan *query processing* (Valduriez et al, 2011). Sehingga pada basis data terdistribusi strategi yang dipilih mempertimbangkan beberapa aspek seperti biaya proses CPU, ruang *disk*, I/O *disk* dan juga biaya transfer antar *site* (Valduriez et al, 2011).

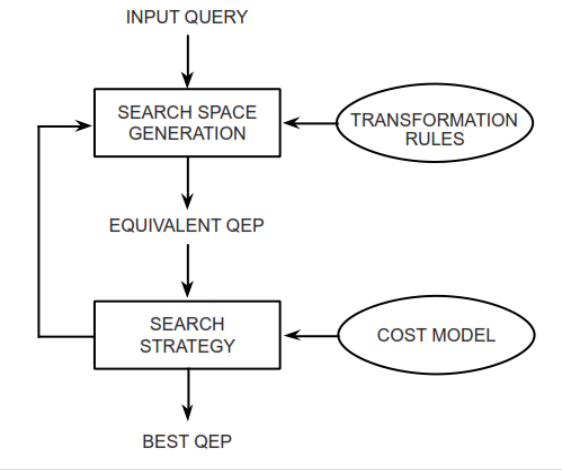
2.6.4 *Distributed Query Execution*

Lapisan terakhir dilakukan oleh semua situs yang memiliki fragmen yang terlibat dalam *query* diaman setiap *subquery* mengeksekusi pada satu situs, yang disebut *query* lokal, kemudian dioptimalkan menggunakan skema situs lokal dan dieksekusi (Valduriez et al, 2011). Pada saat ini, algoritme untuk melakukan operator relasional dapat dipilih. Optimasi lokal menggunakan algoritme sistem terpusat (Valduriez et al, 2011).

2.7 *Query Optimization*

Query optimizaton merupakan tugas sulit pada sebuah lingkungan *client/server* yang terdistribusi dimana lokasi data menjadi faktor utama. Agar optimasi *query* secara akurat dapat dilakukan, beberapa informasi dibutuhkan untuk menentukan mana teknik akses data yang paling efektif (Contoh : tabel dan cardinalitas kolom, informasi organisasi dan *index* yang tersedia). Algoritme optimasi memiliki dampak penting pada performansi QDP (Alom, 2009). Pada DDB sistem *query processing* dan optimasi *query* dibatasi oleh parameter biaya yang berbeda seperti, *communication cost*, *local processing cost*, *optimization cost*, *query localization cost* dll (Mishra, 2015).

Agar operasi *query* tidak lambat diperlukan metode yang dapat mengoptimalkan operasi *join query* pada akses basis data terdistribusi. Suatu metode optimasi *query* mencoba untuk menentukan cara yang paling optimal untuk mengeksekusi *query* yang diberikan dengan mempertimbangkan rencana *query* yang mungkin (Wang, 2009). Optimasi *query* mengacu pada proses menghasilkan rencana eksekusi *query* / *Query Equivalent Plan (QEP)* yang merupakan strategi eksekusi untuk *query*. QEP ini meminimalkan fungsi biaya objektif (Valduriez et al, 2011). Sebuah *query optimizer*, modul *software* yang melakukan optimasi *query*, biasanya terdiri dari tiga komponen: ruang pencarian, model biaya, dan strategi pencarian (Valduriez et al, 2011).



Gambar 2.20 Optimasi Query

Sumber:Valduriez et al (2011)

2.7.1 *Search Space*

Rencana eksekusi *query* biasanya diabstraksikan dengan cara *operator tree* yang mendefinisikan urutan di mana operasi dieksekusi (Valduriez et al, 2011). Mereka diperkaya dengan informasi tambahan, seperti algoritme terbaik yang dipilih untuk setiap operasi (Valduriez et al, 2011). Untuk *query* yang diberikan, *search space* sehingga dapat didefinisikan sebagai kumpulan *operator tree* setara yang dapat diproduksi menggunakan aturan transformasi (Valduriez et al, 2011).

2.7.2 *Search Strategy*

Strategi pencarian yang paling populer digunakan untuk pengoptimalan *query* adalah *dynamic progamming*. Strategi deterministik dengan membangun rencana, mulai dari hubungan dasar, bergabung dengan salah satu hubungan yang lebih pada setiap langkah sampai rencana lengkap diperoleh, *dynamic progamming* membangun semua rencana yang mungkin, tahap pertama, sebelum memilih rencana "terbaik" (Valduriez et al, 2011). Untuk mengurangi biaya optimasi, rencana parsial yang tidak cenderung mengarah ke rencana optimal dipangkas (dibuang) sesegera mungkin dan sebaliknya, strategi lain deterministik, *greedy algorithm*, membangun hanya satu rencana (Valduriez et al, 2011).

Dynamic progamming hampir lengkap dan meyakinkan bahwa dari semua rencana yang ditemukan adalah "terbaik" (Valduriez et al, 2011). Biaya optimasi diterima (dalam hal waktu dan ruang) ketika jumlah relasi dalam *query* kecil, namun pendekatan ini menjadi terlalu mahal jika jumlah hubungan lebih besar dari 5 atau 6 (Valduriez et al, 2011). Untuk pertanyaan lebih kompleks, strategi random telah diusulkan, yang mengurangi kompleksitas optimasi namun tidak menjamin yang terbaik dari semua rencana (Valduriez et al, 2011). Tidak seperti strategi deterministik, strategi random memungkinkan optimizer untuk menukar waktu optimasi untuk waktu eksekusi (Lanzelotte et al 1993).

2.7.3 *Cost Based Optimization*

Sebuah *cost based optimization* termasuk fungsi biaya untuk memprediksi biaya operator, statistik dan data dasar, dan formula untuk mengevaluasi ukuran antar hasil (Valduriez et al, 2011). Biaya adalah dalam waktu eksekusi, sehingga fungsi biaya merupakan waktu eksekusi *query*, biaya strategi eksekusi terdistribusi dapat dinyatakan sehubungan baik dengan total waktu atau waktu respon (Valduriez et al, 2011). Total waktu adalah jumlah dari semua waktu (juga disebut sebagai biaya) komponen, sementara waktu respon adalah waktu yang telah berlalu dari inisiasi untuk penyelesaian *query* (Valduriez et al, 2011).

2.8 Algoritme Optimasi

Dalam kehidupan sehari-hari seringkali kita berhadapan dengan pencarian solusi suatu masalah seperti contoh berikut:

i. Pembuatan jadwal kuliah yang mencakup ketersediaan dosen dan ruangan. Jadwal harus dibuat tujuan untuk menghindari seorang dosen/mahasiswa terjadwal di lebih dari satu kelas pada waktu yang sama.
ii. Persoalan transportasi yang mencakup pendistribusian suatu komoditas atau produk dari sejumlah sumber kepada sejumlah tujuan dengan tujuan meminimumkan ongkos pengangkutan yang terjadi.
iii. Pemilihan rute terpendek (biaya terkecil) untuk mengunjungi sejumlah kota.
iv. Penentuan komposisi makanan ternak dengan biaya minimum yang harus memenuhi batasan minimal untuk setiap komponen nutrisi.

Penyelesaian masalah di atas akan mudah dilakukan jika ukuran data relatif kecil, tetapi masalah akan menjadi kompleks jika data berukuran besar atau melibatkan sejumlah besar entitas (Mahmudy, 2013). Pada masalah kompleks dibutuhkan juga formulasi matematika yang kompleks yang bisa jadi sangat sulit dibangun atau membutuhkan waktu yang lama. Berdasarkan model matematis yang dibangun bisa dilakukan analisis untuk mencari solusi yang terbaik (optimum), solusi optimum mungkin dapat diperoleh tetapi memerlukan proses perhitungan yang panjang (Mahmudy, 2013).

2.8.1 Deterministik

Detrministic algoritme digunakan untuk menyelesaikan permasalahan dengan data yang jelas dan dapat diterapkan dengan menerapkan perhitungan statistika (Mahmudy, 2013).

2.8.2 *Stochastic*

Algoritme *stochastic* menyelesaikan masalah yang memiliki data random. *Stochastic optimization* menggunakan bilangan acak (*random*) dalam pencarian solusi (Mahmudy, 2013). Sebagai konsekuensinya, sebuah algoritme dalam kelas ini setiap dijalankan akan menghasilkan solusi akhir yang berbeda, meskipun diterapkan dalam permasalahan yang sama (Mahmudy, 2013).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.8.2.1 Meta-Heuristik

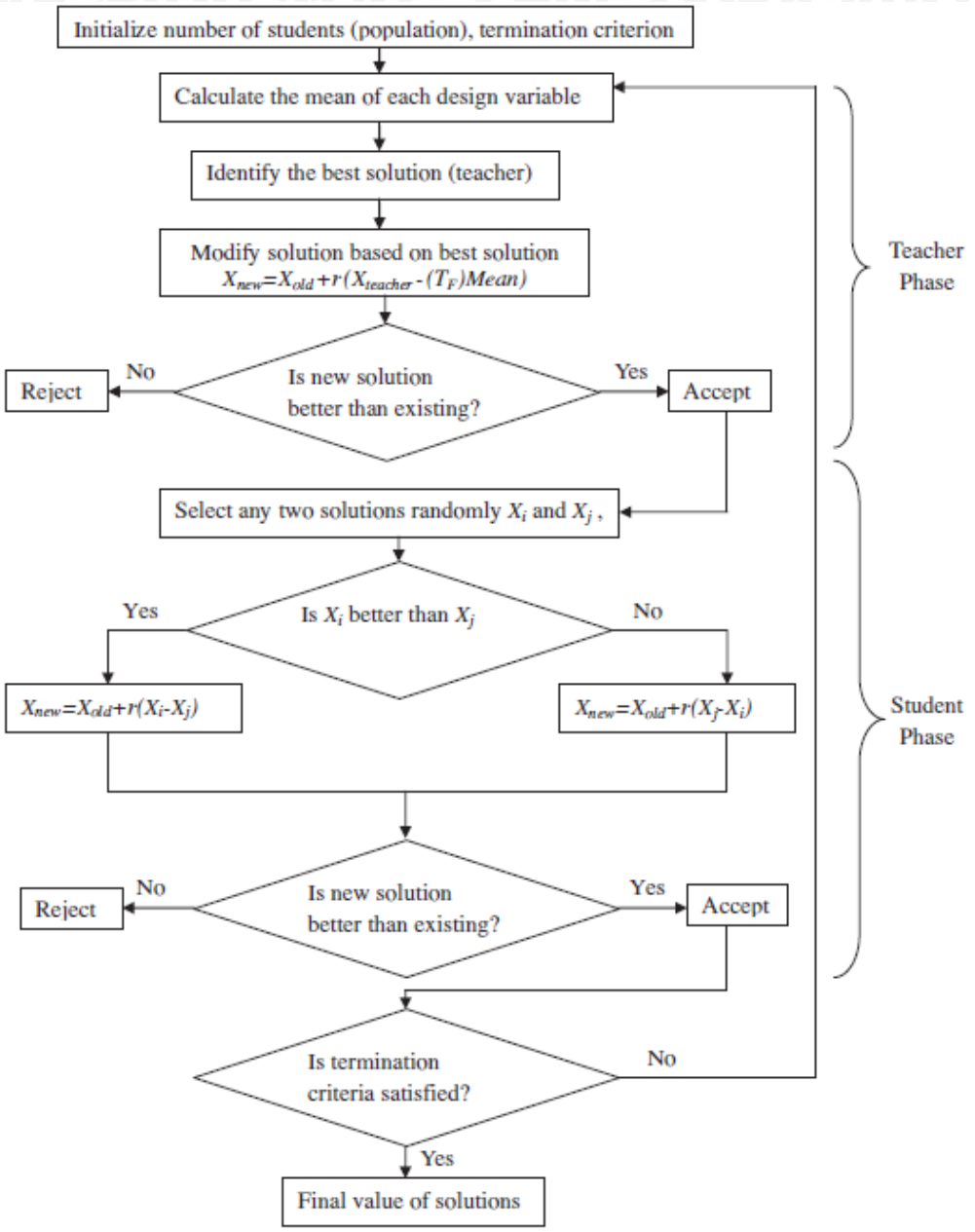
Sebuah algoritme *meta-heuristic* bertindak sebagai "manajer" dari beberapa algoritme *heuristic* untuk secara terorganisir mencari solusi dari sebuah permasalahan, sebagai contoh adalah algoritme genetika yang me-*manage* beberapa *genetic operator* seperti *crossover*, *mutation*, dan s*election* (Mahmudy, 2013).

2.9 *Teacher-Learner Based Optimization (TLBO)*

Algoritme TLBO adalah sebuah algoritme yang diinspirasi dari proses belajar-mengajar yang diusulkan oleh Rao et al (2011) berdasar pada efek dari pengaruh *teacher* pada keluaran *learner* di kelas. Ide utama dari TLBO adalah semulasi dari proses klasik belajar di sekolah yang terdiri dari dua tahap, tahap pertama disebut fase *teacher*, seorang guru memberi pengetahuan secara langsung pada muridnya dimana semakin baik guru semakin banyak pengetahuan yang diperoleh murid tetapi bagimanapun, kemungkinan dari kesuksesan guru selama mengajar dalam prakteknya sangat jarang murid yang dapat mengerti semua materi yang diberikan oleh guru dimana kebanyakan murid akan menerima sebagian dari materi baru yang diajarkan dan dibeberapa kasus guru tidak memberi efek langsung pada pengetahuan murid, selanjutnya tahap kedua adalah fase *learner*, seorang murid mungkin belajar dengan bantuan dari murid lain (Crepinsek, 2012). Secara keseluruhan, berapa banyak pengetahuan yang diberikan ke seorang murid tidak hanya bergantung pada guru mereka tetapi juga dari interaksi antar murid selama pembelajaran (Rao et al, 2012b).

Algoritme ini menganggap sebuah kelompok *learner* sebagai populasi dan subjek-subjek yang berbeda ditawarkan pada *learner* yang dianggap sebagai desain objektif yang berbeda dan hasil dari *learner* dianalogikan sebagai nilai "fitness" dari permasalahan optimasi (Rao et al, 2012a). Solusi terbaik dari keseluruhan populasi dianggap sebagai teacher dan desain objektif sebenarnya adalah parameter-parameter yang terlibat pada fungsi objektif yang diberikan pada permasalahan optimasi dan solusi terbaik adalah nilai terbaik dari fungsi objektif (Rao et al, 2012a).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 2.21 Alur Algoritme *TLBO*

Sumber:Rao et al (2012)

2.9.1 Fase Teacher

Selama fase *teacher*, seorang guru mencoba untuk meningkatkan hasil dari nilai *mean* dari murid (*learner*) di kelas pada subjek yang diajarkan bergantung pada kemampuan dari guru (Rao et al, 2012a). Bagaimanapun, guru adalah orang dengan pengetahuan tinggi yang melatih murid sehingga mereka mendapatkan hasil yang lebih baik (Rao et al, 2012a). Perbedaan nilai dari hasil *mean* yang ada dari setiap subjek dan nilai dari guru untuk setiap subjek dihitung dengan:

$$Difference_Mean_i = r_i(M_{new} - T_F M_i) \quad (2.10)$$

Mi adalah nilai *mean* dan Ti adalah *QP* yang dianggap sebagai *teacher*. Ti akan mencoba untuk meningkatkan nilai *mean* Mi, sehingga nilai dari Ti adalah Mnew dan Tf adalah sebuah *teaching factor* yang menentukan nilai dari *mean* untuk diubah dan ri adalah nomor acak pada rentang [0,1] (Rao et al, 2012b). Nilai dari Tf bisa 1 atau 2 dipilih secara random berdasar :

$$T_F = round[1 + rand(0,1)\{2 - 1\}]. \quad (2.11)$$

Berdasar pada *Difference Mean* maka akan dilakukan *update* pada solusi yang ada di fase teacher dengan cara :

$$X_{new,i} = X_{old,i} + Difference_Mean_i \quad (2.12)$$

2.9.2 Fase Leaner

Pada fase *learner*, murid meningkatkan pengetahuan mereka dengan dua cara yang berbeda yaitu pertama melalui masukan dari guru dan lainnya melalui interaksi di antara mereka (Rao et al, 2012b). Seorang pelajar berinteraksi secara acak dengan pelajar lain dengan bantuan diskusi kelompok, presentasi, komunikasi formal, dll pelajar belajar sesuatu yang baru jika pelajar lainnya memiliki pengetahuan lebih dari dia (Rao et al 2012b). Modifikasi pelajar dinyatakan sebagai,

For $i = 1: P_n$
Randomly select another learner X_j, such that $i \neq j$

If $f(X_i) < f(X_j)$
$X_{new,i} = X_{old,i} + r_i(X_i - X_j)$

Else
$X_{new,i} = X_{old,i} + r_i(X_j - X_i)$
End If
End For
Accept X_{new} if it gives a better function value.

Gambar 2.22 Tahap pada fase *learner*

Sumber: Rao et al, 2012b

Dimana nilai ri adalah nomor acak pada rentang [0,1]. Nilai i dan j sesuai dengan jumlah *query plan* (Rao et al, 2012b).

2.9.3 Desain Parameter Heuristik

Objektif desain heuristik, untuk menyelesaikan permasalahan optimasi sangat optimal bila menggunakan parameter *heuristic* (Mishra, 2015). Berikut merupakan parameter *heuristic* yang dipakai :

1) Objective* 1 - *Query Affinity Cost (QAC)

Jumlah *site* yang terlibat pada setiap *query plan*.

$$QAC = \sum_{i=1}^{M} \frac{ki}{N} \left(1 - \frac{ki}{N}\right) \quad (2.13)$$

Dimana

M = Jumlah *site* yang diakses oleh *query plan*

ki = Waktu yang digunakan *site* ke i pada *query plan*

N = Jumlah relasi yang diakses oleh *query plan*

2) *Objective* 2 - *Query Localization Cost (QLC)*

Jumlah derajat komunikasi antara dua *site* yang berbeda pada *query plan*. Memiliki peran krusial dalam memutuskan *site control* untuk balasan *query* pada masing-masing *query plan*.

$$QPcost = MIN_{i=1}^{Nr}\left[\sum_{j=1 \,\&\&\, j\neq i}^{Nr} \frac{Size\ (Rsj)}{\sum_{ki}^{Nr} Size\ (Rk)}\right] \quad (2.14)$$

Dimana fungsi MIN mendapatkan nilai minimum dari i= (1-Nr)

Nr = Jumlah relasi pada RSM / relasi yang ada pada klausa FROM di *query*

Size (Rsj) = Jumlah tupel pada relasi di *site* Sj

Size (Rk) = Jumlah tupel pada relasi Rk

3) *Objective* 3 - *Local Processing Cost (LPC)*

Fokus pada relasi yang terletak pada *site* lokal dan kemampuannya untuk memilih pada relasi lokal. Pengukuran *Local Processing Cost (LPC)* dibagi menjadi dua sub-perhitungan, pertama adalah *Local Processing Cost (LPC)* pada *remote site* dan kedua *Local Processing Cost (LPC)* pada *control site*.

Relation Processing Cost

$$RPC = Nt * Sr / \sum_{K=1}^{Nr} Nt\ (k) \quad (2.15)$$

a. LPC untuk remote site yang digunakan pada query plan

$$RLPC = Max_{i=1\ to\ Rs}[RPC(i)] \quad (2.16)$$

b. LPC untuk control site yang digunakan pada query plan

$$CLPC = Max_{i,j-1\ to\ Nr}[Nt(JOIN(Ri,Rj)) * Sj(JOIN(Ri,Rj)) / \sum_{k=1}^{Nr} Nt(k)] \quad (2.17)$$

$$LPC\ dari\ query\ plan\ QPi = \sum_{k=1}^{sqp-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ Nr\&\&}R(i)CS\&JOIN[RPC(i)] \quad (2.18)$$

Dimana

Nt = Jumlah tupel

Nr = Jumlah relasi

Sqp = Jumlah total *site* pada *query plan*

Rs = Jumlah total ralasi pada *site* lokal

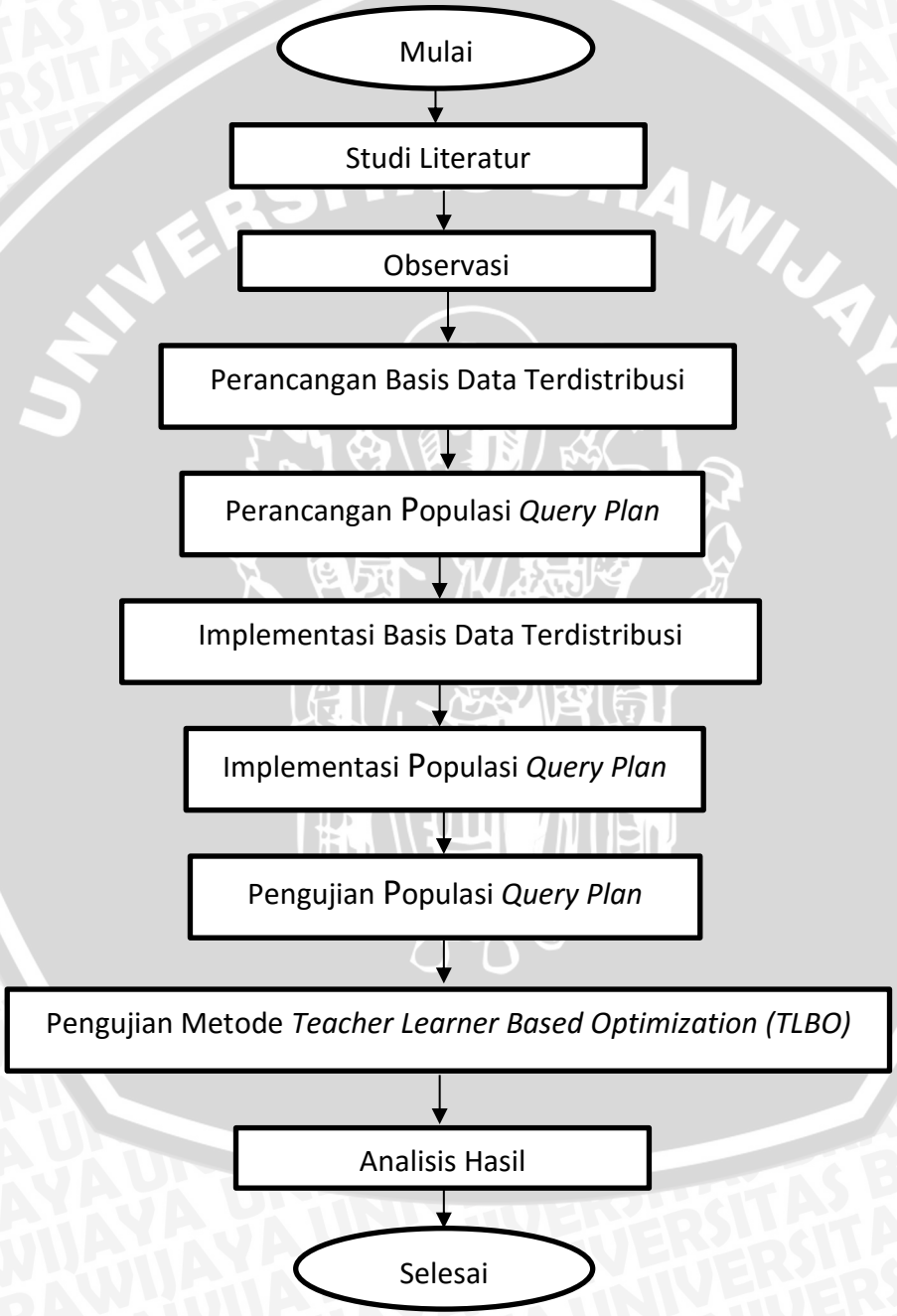
Sr = Kemampuan memilih dari relasi R pada *site* lokal

Sj = Kemampuan JOIN

Nj = Jumlah JOIN pada *query plan*

BAB 3 METODOLOGI

Metodologi penelitian yang akan digunakan untuk menjelaskan pada bab ini. Metodologi menjelaskan metode yang digunakan untuk implementasi pembuatan basis data hingga *query processing* sesuai dengan tujuan penelitian ini yaitu Penentuan Query Optimal Pada Basis Data Terdistribusi Menggunakan Metode *Teacher Learner Based Optimization* (TLBO). Alur penelitian yang diterapkan ditunjukkan dalam gambar 3.1.



Gambar 3.1 Alur Penelitian

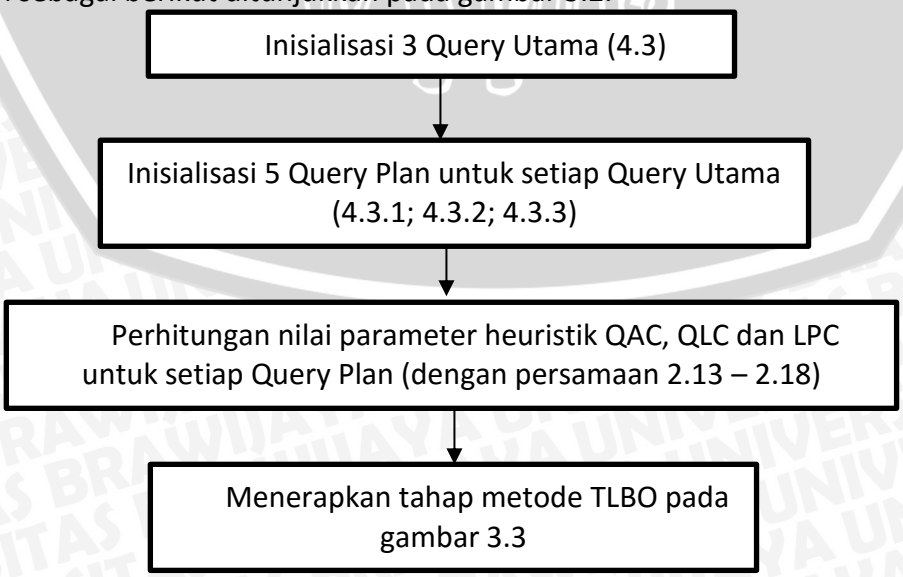
3.1 Studi Literatur

Pada tahap ini, dilakukan studi literatur yakni mengumpulkan data dengan cara mengkaji beberapa referensi yang berhubungan dengan hal-hal yang mendukung proses dan tujuan penelitian ini. Hal-hal yang ingin dipelajari adalah mengenai basis data terdistribusi, optimasi *query*, fragmentasi data, aljabar realsional dan metode yang digunakan pada penelitian ini yakni TLBO sebagai metode untuk menentukan *query* yang optimal pada basis data terdistribusi. Studi literatur yang dilakukan adalah melalui jurnal-jurnal ilmiah yang berhubungan dengan topik, buku mengenai basis data dan juga penelitian yang sejenis.

3.2 Observasi

Pada tahap observasi dilakukan observasi data dengan cara berdiskusi bersama dengan dosen pembimbing skripsi tentang studi kasus, skema basis data, jenis fragmentasi yang digunakan dan data yang akan dijadikan sebagai objek penelitian. Untuk studi kasus yang digunakan adalah data akademik mahasiswa Universitas Brawijaya. Skema dirancang berdasar kebutuhan dalam penelitian ini tetapi tetap mengacu pada skema data akademik mahasiswa Universitas Brawijaya. Data yang dirancang merupakan data yang tidak absah, merupakan data *dummy* yang dibuat sendiri. Karena dalam penelitian ini peneliti hanya akan menguji tentang penggunaan metode TLBO terhadap penentuan *query* yang optimal pada basis data terdistribusi. Jumlah data yang digunakan didapat dari *website* dikti (www.dikti.go.id) dan fragmentasi yang akan digunakan adalah fragmentasi vertikal.

Untuk penerapan metode *Teacher* TLBO dibutuhkan parameter yakni parameter metaheuristik yang berupa jumlah populasi penelitian dan jumlah generasi data, dan parameter heuristik yang berperan sebagai subjek yang ditawarkan pada setiap *learner* maka berdasar jurnal yang diacu pada penerapan metode TLBO parameter heuristik yang digunakan untuk menilai hasil belajar setiap *learner* adalah QAC, QLC *dan* LPC. Tahap sebelum penerapan metode TLBO adalah sebagai berikut ditunjukkan pada gambar 3.2.



Gambar 3.2 Tahap Sebelum Pengujian TLB0

3.3 Desain Basis Data Terdistribusi

Pada tahap ini yaitu mendesain lingkungan basis data terdistribusi yang akan digunakan untuk implementasi *query* basis data. Mendesain basis data terdistribusi dengan menggunakan lima *virtual server*. Desain jaringan basis data terdistribusi didapat melalui proses diskusi bersama dosen. *Virtual server* yang dibentuk memiliki peranan masing-masing seperti satu *virtual server* yang di desain sebagai *site* utama dan empat *virtual server* lainnya bertindak sebagai *site* lokal dimana data sebagai objek penelitian akan ditempatkan. Dari data yang akan digunakan sebagai objek penelitian, data-data yang ada akan dilakukan proses fragmentasi secara vertikal pada level tabel menjadi 5 bagian berdasarkan dari fungsi akses data dari masing-masing *site* sehingga pada akhir pembagian akan terbentuk 5 fragmen yang akan disimpan pada lima *site* atau server.

Pada pendefinisian fragmen menggunakan alternatif vertikal dengan cara memulai mendefinisikan berapa jumlah fragmen yang ingin dibentuk. Disini diputuskan untuk membentuk 5 fragmen dimana setiap fragmen menyimpan data yang berbeda. Lima rencana data yang disimpan pada setiap fragmen adalah data Mahasiswa, data Mahasiswa Biodata, data Orang Tua Mahasiswa, data KHS dan data Keungan Mahasiswa. Fragmenasi dimulai dengan memilih alternatif *splitting* pada fragmentasi vertikal, dimulai dengan level skema dan menentukan manfaat setiap fragmen berdasar jenis data yang disimpan pada setiap fragmen seperti yang didefinisikan sebelumnya. Sehingga ima fragmen dihasilkan dari membagi skema utama berdasar jenis akses data yang dapat dikelompokkan. Kelima fragmen tersebut adalah *site* 1 menangani data Mahasiswa, *site* 2 menangani data Biodata Mahasiswa, *site* 3 menangani data Orang Tua Mahasiswa, *site* 4 menangani data Kartu Hasil Studi mahasiswa dan *site* ke 5 menangani data Keuangan Mahasiswa.

3.4 Perancangan Populasi *Query Plan*

Pada penelitian ini dilakukan penelitian mengenai *reporting query. Reporting query* yang dimaksud merupakan sebuah *query* yang digunakan untuk. Proses optimasi pada *query* yang digunakan untuk menampilkan data IPK setiap fakultas dimaksudkan agar eksekusi *query* dapat diproses secara cepat dan efisien untuk menghemat waktu dan sumber daya yang ada. Tujuan proses penentuan *query* yang optimal ini dikarenakan data yang diproses sangat besar dan juga diperlukan proses JOIN dengan tabel lain yang ada pada *site* berbeda sehingga jika diproses dengan menggunakan *query* yang tidak optimal akan membutuhkan waktu lama dan juga transfer data yang sangat besar. Pada penulisan ini mengangkat hasil data yang akan dieksekusi sebagai berikut :

1. Menampilkan Kartu Hasil Studi Mahasiswa yang meliputi nim, nama, jenjang/fakultas, jurusan program studi, status, semester, tahun akademik, kode mata kuliah, nama mata kuliah, sks dan nilai mahasiswa.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2. Menampilkan data Keuangan Mahasiswa yang meliputi nim, nama, jenjang/fakultas, jurusan program studi, status, semester, transaksi, status, waktu, debet dan kredit.
3. Menampilkan data Orang Tua Mahasiswa yang meliputi data nim, nama, jenjang/fakultas, jurusan program studi, status, semester, nama ayah, nama ibu, alamat, kode pos, Negara, propinsi, kota, pekerjaan ayah, gaji, pendidikan ayah, pekerjaan ibu, gaji dan pendidikan ibu.

Dari 3 *query* yang dirancang di atas akan dilakukan perancangan model *query plan* yang *equivalent* dimanan model *query* berbeda tetapi mengembalikan data yang sama dari setiap *query* tersebut. Terdapat 5 *query plan* yang *equivalent* untuk setiap *query*, *query plan* yang *equivalent* ini memiliki perbedaan model *query* pada jumlah *site* yang terlibat dan juga konsentrasi relasi pada setiap *site*.

3.5 Implementasi Basis Data Terdistribusi

Pada tahap implementasi basis data terdistribusi disiapkan 5 *server* pada *Vmware Workstation* dengan sistem operasi yang digunakan adalah *centos* 6.5 *64bit*. Pada setiap *server* dilakukan penginstallan *server* DB2 yakni *IBM DB2 expc 64 bit* dan pengaturan jaringan serta IP pada setiap *server* sehingga dapat saling terhubung antar satu *server* dengan *server* lain. Dari skema basis data yang telah dilakukan fragmentasi sebelumnya menjadi 5 skema data yang akan diimplementasikan pada setiap fragmen dengan tahap membuat tabel dari keseluruhan skema fragmentasi ke setiap *server* basis data. Tabel pada setiap *server* telah dibuat dan telah ditentukan mana *server* yang merupakan *site* utama dan *site* lokal, selanjutnya dilakukan federasi pada basis data sehingga antar *site* dapat saling berhubungan mengirim dan meminta data satu sama lain. Setelah proses federasi sukses maka tahap terakhir pada implementasi basis data terdistribusi adalah mengisi tabel yang telah dibuat dengan data *dummy* yang sebelumnya telah dirancang sesuai dengan skema dan jumlah data mahasiswa yang didapatkan.

3.6 Implementasi Populasi *Query Plan*

Pada tahap ini dilakukan implementasi aljabar relasional dari populasi *query plan* menjadi *query* yang dapat dieksekusi pada *IBM Data Studio.* Selanjutnya populasi *query plan* yang *equivalent* tersebut dieksekusi pada *IBM Data Studio* dengan tujuan mengetahui apakah 5 model *query plan* yang telah dirancang benar-benar menghasilkan data keluaran yang sama sehingga akan sah bila dilakukan perbandingan.

3.7 Pengujian Populasi *Query Plan*

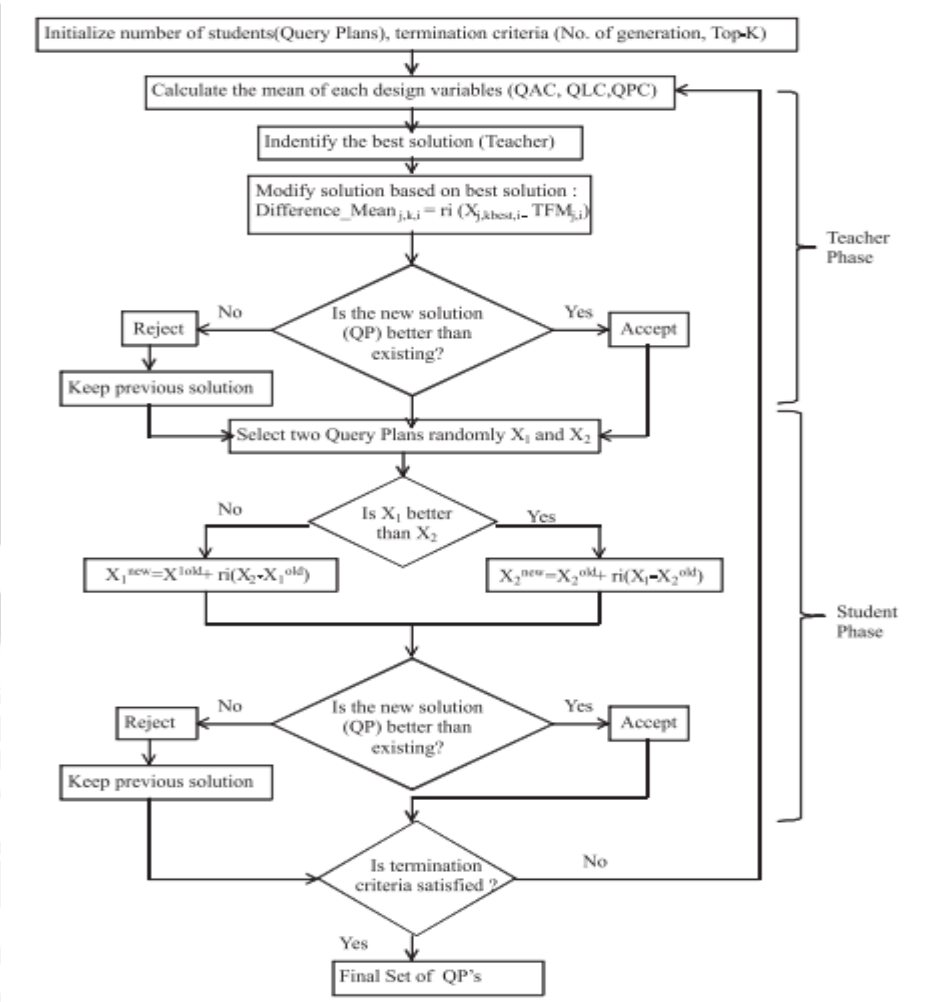
Pada tahap ini dilakukan pengujian terhadap populasi *query plan*. Pengujian yang dimaksud disini adalah menilai setiap *query plan* yang dianalogikan sebagai *learner* berdasar parameter heuristik yang dianalogikan sebagai subjek yang ditawarkan kepada *learner*. Setiap *query plan* akan dinilai dengan melakukan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

perhitungan berdasar 3 parameter heuristik yang dipilih yaitu QAC, QLC *dan* , pada bab 2 telah dijabarkan persamaan dari ketiga parameter tersebut maka dengan persamaan tersebut akan dilakukan perhitungan dan mengahsilkan nilai/hasil yang merupakan biaya eksekusi dari setiap *query plan*.

3.8 Pengujian Metode *Teacher Learner Based Optimization (TLBO)*

Pada tahap ini berdasar hasil perhitungan tersebut menjadikan setiap *query plan* memiliki 3 nilai yang mewakili biaya eksekusi mereka. Nilai-nilai tersebut akan menjadi masukan pada implementasi metode TLBO. Untuk mempermudah proses perbandingan maka dibutuhkan satu nilai yang mampu mewakili ketiga nilai tersebut, dalam algoritme berbasis populasi biasanya nilai tersebut disebut nilai *fitness*. Pada permasalahan ini, nilai *fitness* digunakan sebagai nilai yang mampu mewakili ketiga nilai dari setiap *query plan* maka nilai *fitness* diambil dari rata-rata dari ketiga nilai setiap *query plan*. Berikut adalah skema implementasi metode TLBO ditunjukkan dalam gambar 3.3.



Gambar 3.3 Skema *TLBO* untuk menentukan *Query Plan* yang Optimal pada Basis Data Terdistribusi

Sumber : Mishra (2015)

Seperti dalam gambar 3.2 terdapat fase *teacher* dimana guru memberikan pengetahuan pada *learner* sehingga meningkatkan nilai mereka, selanjutnya fase *learner* dimana murid berinteraksi dengan murid lainnya di kelas untuk meningkatkan nilai mereka sehingga hasil akhir merupakan nilai yang paling optimal.

3.9 Analisis Hasil

Pada tahap ini melakukan analisis hasil dari implementasi metode TLBO sehingga dapat diketahui *query plan* mana yang paling optimal serta dapat mengetahui pengaruh dari setiap parameter heuristik yang terlibat. Menganalisis bagaimana metode TLBO dapat digunakan untuk menentukan *query* yang optimal dan menganalisis bagaimana metode tersebut dapat bekerja pada studi kasus yang dipilih berdasar peran dari setiap parameter heuristik.

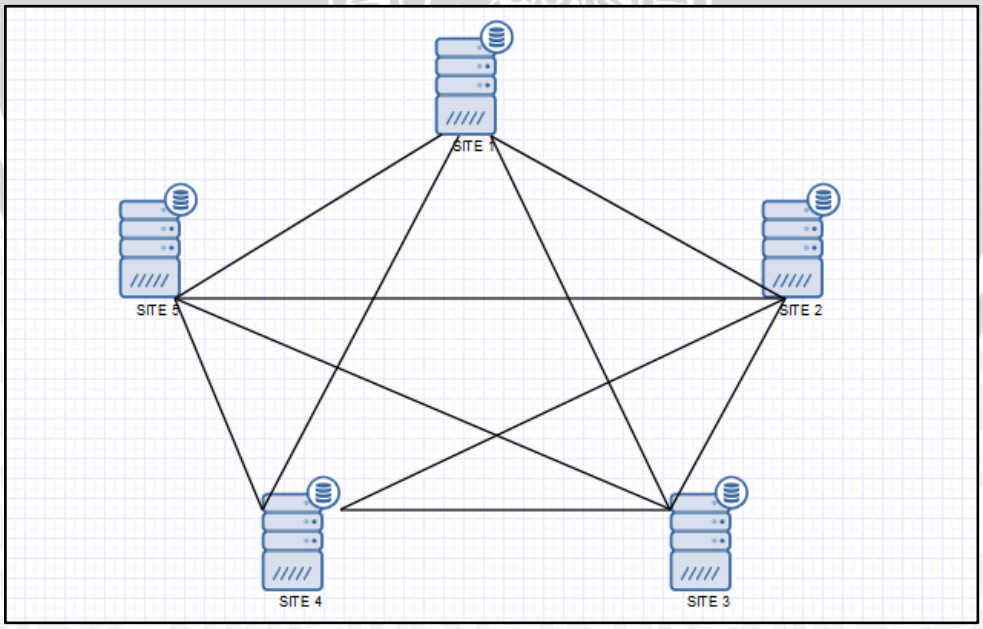
BAB 4 HASIL PERANCANGAN

Pada bab ini akan menjelaskan mengenai sketsa atau rancangan dari proses implementasi penelitian. Rancangan ini digunakan sebagai kerangka bentuk proses penyusunan basis data terdistribusi, fragmentasi data dan juga *query plan*. Ketiga rancangan tersebut nantinya akan dibentuk menjadi sebuah implementasi sehingga dapat dilakukan uji coba dan evaluasi terhadap hasil yang terbentuk dan ditarik menjadi sebuah kesimpulan.

4.1 Jaringan Komputer

Untuk dapat saling berhubungan antar *site* pada sistem *database* terdistribusi diperlukan jaringan komunikasi komputer. Jaringan komputer adalah interkoneksi antara sejumlah komputer autonomous yang dapat saling bertukar informasi antara komputer yang terhubung. Bentuk komputer yang saling terhubung biasanya disebut dengan *node*, *host* atau *site*. Bentuk hubungan antar komputer tidak hanya melalui kawat tembaga saja, tetapi dapat melalui serat *optic*, gelombang mikro dan satelit komunikasi.

Dalam penelitian ini, menggunakan aplikasi *vmware* untuk membentuk *virtual* sehingga jaringan yang digunakan merupakan topologi *mesh*. Topologi *mesh* memungkinkan setiap *virtual* yang terbentuk dapat saling berkomunikasi dengan *virtual* lainnya. Terdapat 5 *virtual server* yang dibentuk untuk penelitian ini dimana setiap *virtual server* dapat saling berkomunikasi dalam jaringan yang dibentuk oleh media aplikasi virtualisasi. Jumlah *virtual server* sesuai dengan jumlah *site* yang akan dibentuk.



Gambar 4.1 Topologi Mesh

Dalam gambar 4.1 menunjukkan lima *virtual server* yang dibuat. Kelima server dalam skema jaringan di atas saling terhubung sehingga membentuk jaringan dengan topologi *mesh*. Dengan menggunakan topologi *mesh* maka memungkinkan kelima *virtual serve*r basis data dapat saling terhubung.

4.2 Basis Data

Basis data terdistribusi pada penelitian ini merupakan basis data terdistribusi yang bersifat homogen, dimana menggunakan DBMS yang sama untuk semua *site*. Sistem homogen lebih mudah dirancang dan dikelola. Pendekatan ini memberikan perkembangan yang baik, tidak mengalami kesulitan dalam membuat sebuah *site* yang baru.

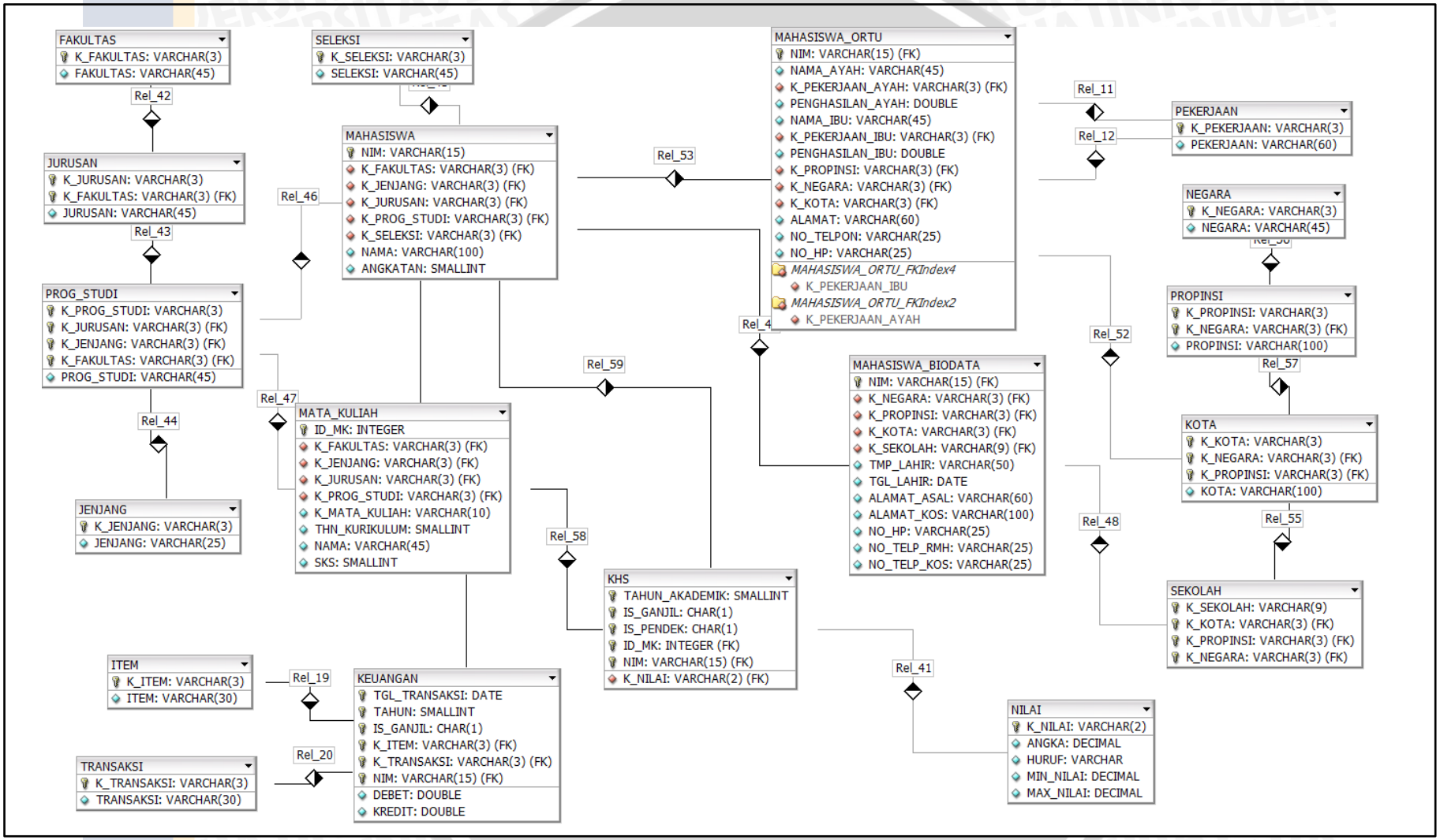
4.2.1 Relational Database Manaement System

Dalam penelitian ini menggunakan skema basis data sistem informasi akademik mahasiswa dari Universitas Brawijaya. Skema dirancang sendiri untuk memenuhi kebutuhan penelitian tetapi tetap mengacu pada skema sistem informasi akademik mahasiswa Universitas Brawijaya.

Dalam penelitian ini, skema relasi antar tabel pada setiap *site* yang digunakan dirancang berbeda. Skema pada setiap *site* memiliki tugas masing-masing untuk pengaksesan data tertentu. Masing-masing tabel pada setiap *site* memiliki relasi antar satu dengan tabel yang lainnya. Pada setiap tabel terdapat *primary key* sebagai kolom kunci utama pada setiap tabel dan *foreign key* sebagai kunci penghubung relasi antar tabel. Rancangan *relational database* yang digunakan ditunjukkan dalam gambar 4.2.

Skema RDBMS dalam gambar 4.2 terdapat beberapa relasi yang saling menghubungkan antar satu tabel dengan tabel lainnya. Terdapat sembilan belas tabel dan dua puluh satu relasi, relasi yang terdapat pada skema adalah *one to many*. Pada relasi *one to many* menandakan bahwa satu *record* data pada setiap tabel tertentu digunakan pada beberapa data pada tabel relasinya. Pemecahan relasi pada skema utama dilakukan dengan cara fragmentasi vertikal, pemecahan skema dilakukan pada tabel level, sehingga 1 tabel tidak difragmentasi tetapi langsung satu skema yang dipecah per tabel.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

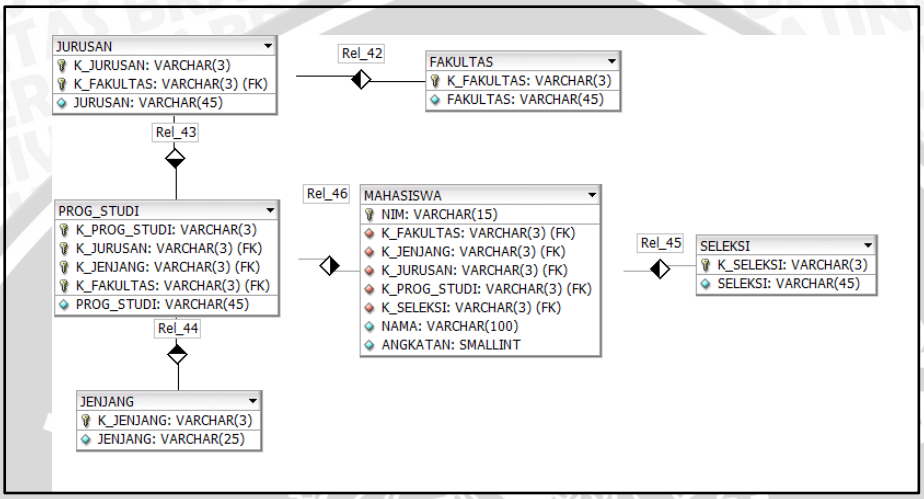


Gambar 4.2 Skema RDBMS SIAKAD

Sumber:Diadaptasi dari Nugroho (2015) dan Prawira (2012)

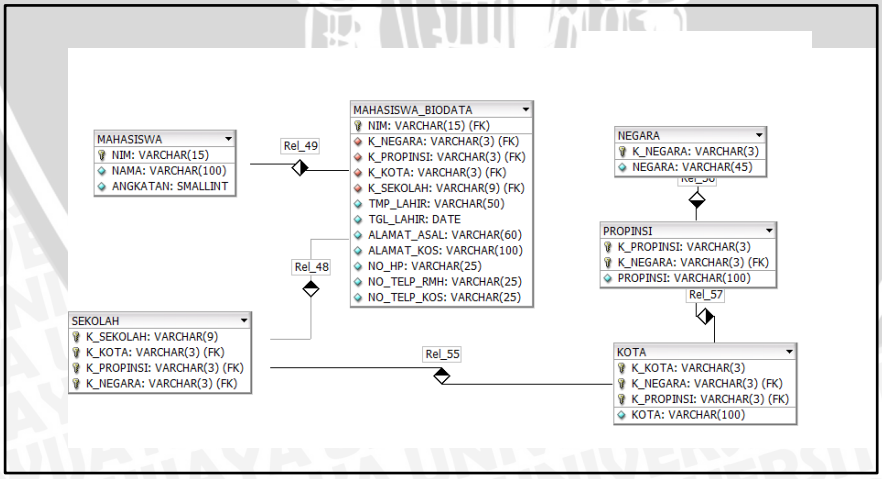
4.2.2 Skema Fragmentasi Vertikal

Dari skema dalam gambar 4.2 selanjutnya akan dipecah menjadi beberapa *site* dengan menggunakn fragmentsi vertikal. Seperti yang telah dijelaskan di atas skema pada setiap *site* berbeda. Dengan skema yang berbeda maka data yang terdapt disetiap site berbeda juga, terdapat lima *site* yang akan mengandung data masing-masing, data mahasiswa, data biodata mahasiswa, data biodata orang tua mahasiswa, data KHS mahasiswa dan data keuangan mahasiswa.



Gambar 4.3 Skema Mahasiswa

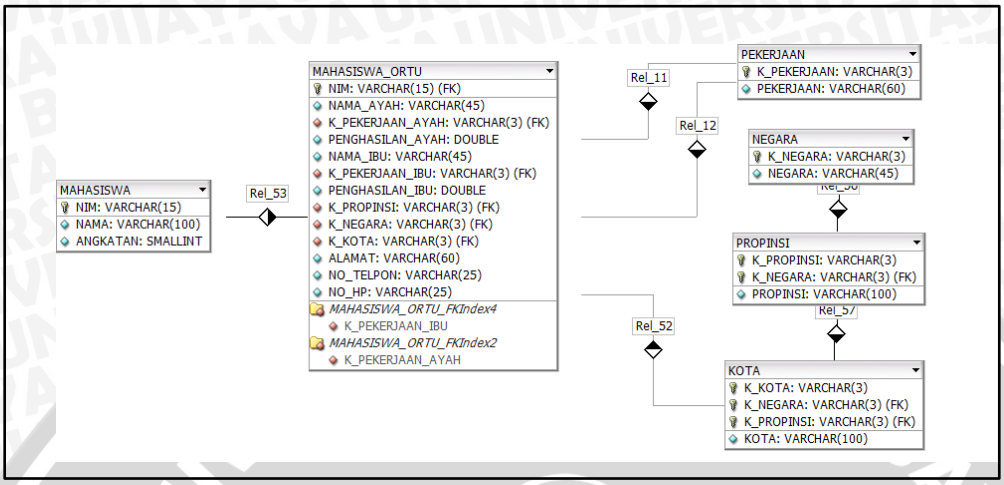
Dalam gambar 4.3 adalah skema Mahasiswa hasil dari fragmentasi vertikal skema utama Sistem Infromasi Akademik Mahasiswa. Skema tersebut dihasilkan dari fragmentasi vertikal terhadap skema utama dengan mempertimbangkan kebutuhan dari data Mahasiswa maka terdapat enam tabel pada skema Mahasiswa, yakni tabel Fakultas, tabel Jurusan, tabel Program studi, tabel Jenjang, tabel Mahasiswa dan tabel Seleksi.



Gambar 4.4 Skema Mahasiswa Biodata

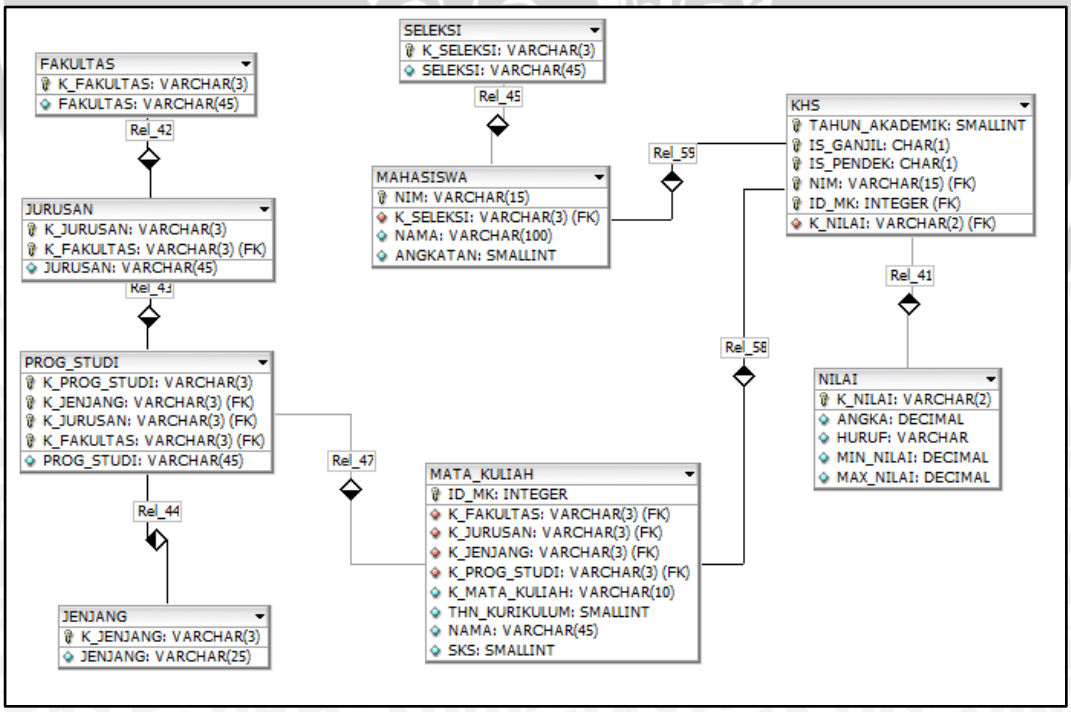
Dalam gambar 4.4 adalah skema Mahasiswa Biodata hasil dari fragmentasi vertikal skema utama Sistem Infromasi Akademik Mahasiswa. Skema tersebut

dihasilkan dari fragmentasi vertikal terhadap skema utama dengan mempertimbangkan kebutuhan dari data Mahasiswa Biodata maka terdapat enam tabel pada skema Mahasiswa Biodata, yakni tabel Negara, tabel Propinsi, tabel Kota, tabel Sekolah, tabel Mahasiswa dan tabel Mahasiswa Biodata.



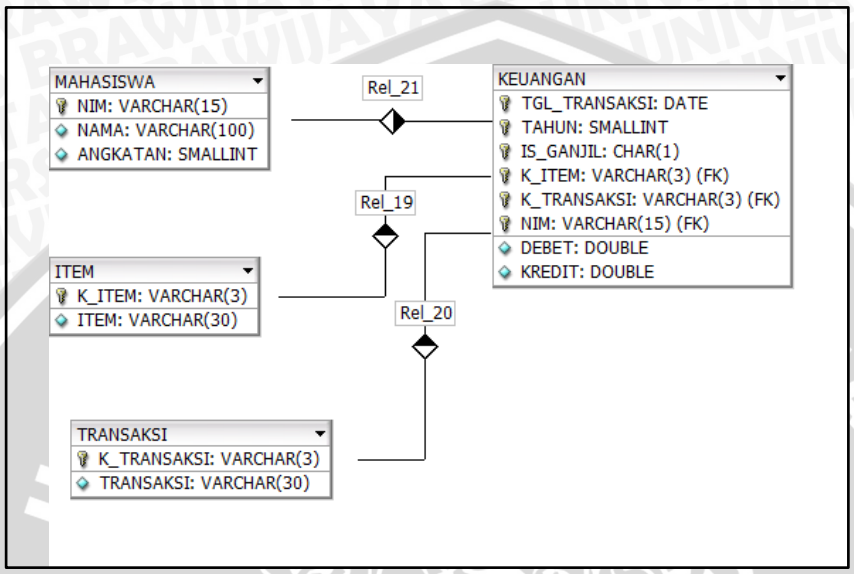
Gambar 4.5 Skema Orang Tua Mahasiswa

Dalam gambar 4.5 adalah skema Biodata Orang Tua hasil dari fragmentasi vertikal skema utama Sistem Infromasi Akademik Mahasiswa. Skema tersebut dihasilkan dari fragmentasi vertikal terhadap skema utama dengan mempertimbangkan kebutuhan dari data Biodata Orang Tua maka terdapat enam tabel pada skema Biodata Orang Tua, yakni tabel Pekerjaan, tabel Negara, tabel Propinsi, tabel Kota, tabel Mahasiswa dan tabel Mahasiswa Ortu.



Gambar 4.6 Skema KHS Mahasiswa

Dalam gambar 4.6 adalah skema KHS Mahasiswa hasil dari fragmentasi vertikal skema utama Sistem Infromasi Akademik Mahasiswa. Skema tersebut dihasilkan dari fragmentasi vertikal terhadap skema utama dengan mempertimbangkan kebutuhan dari data Mahasiswa maka terdapat enam tabel pada skema KHS Mahasiswa, yakni tabel Fakultas, tabel Jurusan, tabel Program studi, tabel Jenjang, tabel Mahasiswa, tabel Seleksi, tabel Nilai, tabel KHS dan tabel Mata Kuliah.



Gambar 4.7 Skema Keuangan Mahasiswa

Dalam gambar 4.7 adalah skema Keuangan Mahasiswa hasil dari fragmentasi vertikal skema utama Sistem Infromasi Akademik Mahasiswa. Skema tersebut dihasilkan dari fragmentasi vertikal terhadap skema utama dengan mempertimbangkan kebutuhan dari data Mahasiswa maka terdapat enam tabel pada skema Keuangan Mahasiswa, yakni tabel Mahasiswa, tabel Item, tabel Transaksi dan tabel Mahasiswa.

4.2.3 Data SIAKAD Unversitas Brawijaya

Data yang digunakan dalam penelitian ini merupakan data *dummy* dari basis data SIAKAD Universitas Brawijaya. Data SIAKAD yang digunakan adalah jumlah data mahasiswa S1 aktif semester genap tahun ajaran 2014/2015 yang valid (sumber : forlap.dikti.go.id). Jumlah data tersebut diambil pada bulan Desember 2015. Pada tahun ajaran 2015/2016 semester ganjil, Universitas Brawijaya memiliki 16 Fakultas. Table 4.1 merupakan data mahasiswa S1 aktif setiap fakultas per semester ganjil bulan desember 2015 tahun ajaran 2015/2016. Jumlah total mahasiswa dari ke 15 fakultas adalah 48181 mahasiswa.

Tabel 4.1 Jumlah Data Mahasiswa S1 aktif sampai dengan Desember 2015

NO	FAKULTAS	JUMLAH MAHASISWA
1	Ekonomi Bisnis	3694
2	Hukum	2093
3	Ilmu Administrasi	6568
4	Peternakan	1932
5	Teknik	4797
6	Ilmu Budaya	3499
7	Ilmu Sosial dan Ilmu Politik	5041
8	Kedokteran	2208
9	Matematika dan IPA	1756
10	Pertanian	4514
11	Perikanan dan Ilmu Kelautan	4762
12	Teknologi Pertanian	2739
13	Ilmu Komputer f.k.a PTIIK	3431
14	Pendidikan Kedokteran Hewan	801
15	Pendidikan Kedokteran Gigi	346

4.2.4 Fragmentasi Data

Sebelum menerapkan metode pemilihan *query* yang optimal pada basis data, terlebih dahulu dilakukan fragmentasi vertikal untuk didistribusikan pada setiap *site*. Fragmentasi data dilakukan agar tabel terkait menjadi lebih ringkas dan meningkatkan ketersediaan data. Dengan demikian penggunaan metode menjadi lebih maksimal. Selain itu, penggunaan fragmentasi vertikal untuk mengkodisikan penelitian seperti jurnal acuan dalam penerapan metode *Tearcher-Learner Based Optimization*. Dari data tabel 4.1 akan dilakukan *pre-processing* menggunakan *query* untuk membagi data menjadi lima fragmen berdasarkan data yang selanjutnya dimasukkan ke basis data pada setiap situs. Anggota setiap fragmen didasarkan pada jumlah mahasiswa yang ada pada setiap fakultasnya. Dengan *pre-processing* tersebut dihasilkan fragmen yang dapat dilihat pada table 4.2.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.2 Jumlah data mahasiswa aktif S1 hasil fragmentasi

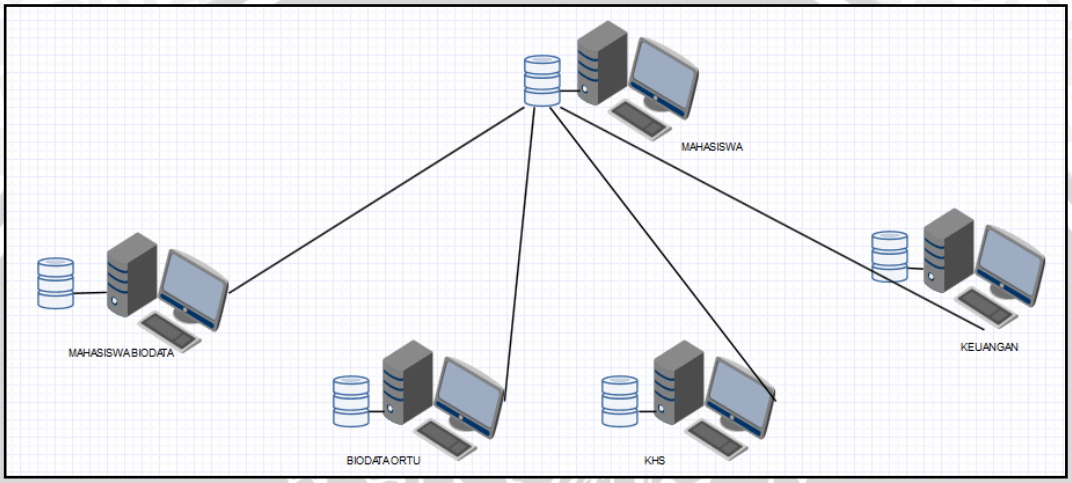
No Fragmen / SItus	Akses Data	Relasi	Jumlah Data
1	Mahasiswa	Fakultas	15
		Jurusan	64
		Program Studi	64
		Jenjang	5
		Seleksi	27
		Mahasiswa	48181
2	Mahasiswa Biodata	Mahasiswa	48181
		Mahasiswa Biodata	48181
		Sekolah	2096
		Kota	131
		Negara	2
		Propinsi	33
3	Data Orang Tua	Mahasiswa	48181
		Mahasiswa_Ortu	48181
		Negara	2
		Propinsi	33
		Kota	131
		Pekerjaan	51
4	KHS	Mashasiswa	48181
		Mata_Kuliah	31549
		KHS	65535
		Nilai	9
		Fakultas	15
		Jurusan	64
		Program Studi	64
		Jenjang	5
		Seleksi	27
5	Keuangan	Mahasiswa	48181
		Keuangan	96262
		Keterangan	2
		Item	3

Fragmen tersebut masing-masing diletakkan ke 5 situs yang berbeda dengan nomor situs yang sesuai nomor fragmennya. Setiap situs terhubung dengan jaringan komputer.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.2.5 Skema Basis Data Terdistribusi

Dalam penelitian ini dibentuk lima *virtual server* sesuai dengan topologi jaringan yang telah didefinisikan di awal bab 4. Empat diantara server tersebut akan bertindak sebagai situs lokal dan satu *virtual server* sebagai situs utama. Langkah pertama dilakukan konfigurasi federasi basis data pada setiap situs kemuadian dilakukan proses katalog *node* pada basis data situs lokal hingga pembuatan *nick name* untuk setiap tabel yang terdapat pada federasi basis data. Kerangka federasi basis data dari pembentukan lima *virtual server* / lima situs ditunjukkan dalam gambar 4.1.



Gambar 4.8 Basis Data Terdistribusi

Dalam gambar 4.8 setiap basis data yang terdapat di masing-masing situs lokal akan menyimpan data hasil fragmentasi vertikal yang ada pada table 4.2 menjadi lima situs.

4.3 Pemilihan *Query*

Untuk pemilihan *query* dalam penelitian ini dirancang 5 model *query* yang berbeda untuk mengakses 1 data yang sama. Sehingga akan dilakukan implementasi dan analisis rancangan *query* tersebut untuk memilih dari 5 *query* yang dirancang mana yang memiliki biaya akses terkecil dan paling optimal.

Dalam penelitian ini akan dilakukan uji coba terhadap 3 *query* untuk akses data yang berbeda dimana masing-masing *query* memiliki 5 model *query* yang berbeda untuk menghasilkan data yang sama. *Query* yang dirancang dalam prosesnya akan melibatkan beberapa situs yang ada sehingga untuk melibatkan keseluruhan situs dirancang 3 *query* yang berbeda. *Query* yang akan dioptimasi adalah *query* untuk mendapatkan data :

1. Menampilkan Kartu Hasil Studi Mahasiswa yang meliputi nim, nama, jenjang, fakultas, jurusan, program studi, semester, tahun akademik, kode mata kuliah, nama mata kuliah, sks dan nilai mahasiswa.

2. Menampilkan data keuangan mahasiswa yang meliputi nim, nama, jenjang/fakultas, jurusan, program studi, semester, transaksi, status, waktu, debet dan kredit.
3. Menampilkan data orang tua mahasiswa dan data biodata mahasiswa yang meliputi data nim, nama, jenjang, fakultas, jurusan, program studi, semester, no hp, alamat kos, nama ayah, nama ibu, alamat, kode pos, Negara, propinsi, kota, pekerjaan ayah, gaji, pendidikan ayah, pekerjaan ibu, gaji dan pendidikan ibu.

4.3.1 *Query* 1

Untuk perancangan model *query* 1, *query* ini digunakan untuk menampilkan hasil seperti yang telah dijelaskan pada pemilihan *query* nomor 1 yaitu menampilkan data yang ada pada kartu hasil studi mahasiswa. Sehingga pada proses ini memerlukan data yang berada pada tabel MAHASISWA, FAKULTAS, JURUSAN, PROGRAM STUDI, JENJANG, MATA_KULIAh, NILAI dan KHS. Tabel MAHASISWA memiliki relasi dengan tabel PROGRAM STUDI dan SELEKSI terletak pada atribut kode program studi dan kode seleksi, tabel PROGRAM STUDI memiliki relasi dengan JURUSAN, FAKULTAS dan JENJANG terletak pada atribut kode jurusan, kode fakultas dan kode jenjang, tabel MAHASISWA memiliki relasi dengan tabel KHS terletak pada atribut nim, tabel MATA_KULIAH memiliki relasi dengan tabel KHS terletak pada atribut kode mata kuliah dan tabel KHS memiliki relasi dengan tabel NILAI pada atribut kode nilai.

a. *Relational Calculus Tuple*

[x1.nim, x1.nama,x1.jenjang, x1.fakultas, x1.jurusan, x1.prog_studi, x1.tahun_akademik, x1.semester, x1.kode_mk, x1.nama_mk, x1.sks, x1.nilai |

(∀m)(MAHASISWA (m)(m.nim =x1.nim ∧ m.nama = x1.nama ∧

(∀p)(PROG_STUDI(p)(m.k_fakultas, m.k_jurusan, m.k_prog_studi, m.k_jenjang = p.k_fakultas, p.k_jurusan, p.k_prog_studi, p.k_jenjang ∧ p.prog_studi = x1.prog_studi ∧

(∀jr)(JURUSAN(jr)(p.k_jurusan = j.k_jurusan ∧ p.k_fakultas = jr.k_fakultas ∧ j.jurusan = x1.jurusan ∧

(∀f)(FAKULTAS(f)(j.k_fakultas = f.k_fakultas ∧ f.fakultas = x1.fakultas)))) ∧

(∀jn)(JENJANG(jn)(p.k_jenjang = jn.k_jenjang ∧ j.jenjang = x1.jenjang)))) ∧

(∀k)(KHS(k)(m.nim = k.nim ∧ k.tahun_akademik = x1.tahun_akademik ∧ k.is_ganjil = x1.semester ∧

(∀mk)(MATA_KULIAH(mk)(mk.id_mk = k.id_mk ∧ mk.k_mata_kuliah = x1.kode_mk ∧ mk.nama = x1.nama_mk ∧ mk.sks = x1.sks)) ∧

(∀n)(NILAI(n)(k.k_nilai = n.k_nilai ∧ n.huruf = x1.nilai))))))]

b. Relational Calculus Domain

{nim: x1 , nama: x7, jenjang: x19, fakultas: x18, jurusan: x16, prog_studi: x13, semester: x22, tahun_akademik: x21, kode_mk: x31, nama_mk: x33, sks: x34, nilai: x37 |

(∀x1,∀x2,∀x3,∀x4,∀x5,∀x6)(MAHASISWA(nim: x1, k_fakultas: x2, k_jenjang: x3, k_jurusan: x4, k_prog_studi: x5, k_seleksi: x6, nama: x7, angkatan: x8) ∧

(∀x9,∀x10,∀x11,∀x12)(PROG_STUDI(k_prog_studi: x9, k_jurusan: x10, k_jenjang: x11, k_fakultas: x12, prog_studi: x13) ∧ x2 = x9 ∧

(∀x14,∀x15)(JURUSAN(k_jurusan: x14, k_fakultas: x15, jurusan: x16) ∧ x10 = x14 ∧

(∀17)(FAKULTAS(k_fakultas: x17, fakultas: x18))) ∧ x15 = x17 ∧

(∀x18)(JENJANG(k_jenjang: x18, jenjang: x19))) ∧ xx11 = x18 ∧

(∀x20,∀x21,∀x22,∀x23,∀x24,)(KHS(nim: x20, tahun_akademik: x21, is_ganjil: x22, is_pendek: x23, id_mk: x24, k_nilai: x25) ∧ x1 = x20 ∧

(∀x26)(MATA_KULIAH(id_mk: x26, k_fakultas: x27, k_jenjang: x28, k_jurusan: x29, k_prog_studi: x30, k_mata_kuliah: x31, thn_kurikulum: x32, nama: x33, sks: x34) ∧ x24 = x26 ∧

(∀x35)(NILAI(k_nilai: x35, angka: x36, huruf: x37, min_nilai: x38, max_nilai: x39) ∧ x25 = x35))))}

4.3.1.1 *Query* 1 Model 1

Pada model *query* 1, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MATA_KULIAH (MK), NILAI (N) dan KHS (KHS) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 2 *site* yakni *site* Mahasiswa dan *site* KHS. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa, tabel Jurusan, tabel Prog_studi, tabel Fakultas dan tabel Jenjang. Pada *site* KHS tabel yang terlibat yakni tabel KHS, tabel Nilai dan tabel Mata_kuliah. Berikut adalah aljabar relational dari model akses *query* 1.

report ← ρR(nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←ρR(tahun_akademik, semester, id_mk, nim, k_nilai) (tahun_akademik, is_ganjil, id_mk, nim, k_nilai $^{(KHS)}$)

result←π(nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, semester, kode_mk, nama_mk, sks, nilai) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim KHS ∧ report1 ⋈ id_mk MATA_KULIAH ∧ report1 ⋈ k_nilai NILAI)

4.3.1.2 *Query* 1 Model 2

Pada model *query* 2, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MATA_KULIAH (MK), NILAI (N) dan KHS (KHS) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 2 *site* yakni *site* Mahasiswa dan *site* KHS. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa. Pada *site* KHS tabel yang terlibat yakni tabel KHS, tabel Nilai, tabel Mata_kuliah, tabel Jurusan, tabel Prog_studi, tabel Fakultas dan tabel Jenjang. Berikut adalah aljabar relational dari model akses *query* 2.

report ← ρR(nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←ρR(tahun_akademik, semester, id_mk, nim, k_nilai) (tahun_akademik, is_ganjil, id_mk, nim, k_nilai $^{(KHS)}$)

result←π(nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, semester, kode_mk, nama_mk, sks, nilai) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim KHS ∧ report1 ⋈ id_mk MATA_KULIAH ∧ report1 ⋈ k_nilai NILAI)

4.3.1.3 *Query* 1 Model 3

Pada model *query* 3, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MATA_KULIAH (MK), NILAI (N) dan KHS (KHS) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 2 *site* yakni *site* Mahasiswa dan *site* KHS. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa, tabel Jurusan, tabel Fakultas dan tabel Jenjang. Pada *site* KHS tabel yang terlibat yakni tabel KHS, tabel Nilai, tabel Prog_studi dan tabel Mata_kuliah. Berikut adalah aljabar relational dari model akses *query* 3.

report ← ρR(nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←ρR(tahun_akademik, semester, id_mk, nim, k_nilai) (tahun_akademik, is_ganjil, id_mk, nim, k_nilai $^{(KHS)}$)

result←π(nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, semester, kode_mk, nama_mk, sks, nilai) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim KHS ∧ report1 ⋈ id_mk MATA_KULIAH ∧ report1 ⋈ k_nilai NILAI)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.3.1.4 *Query* 1 Model 4

Pada model *query* 4, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MATA_KULIAH (MK), NILAI (N) dan KHS (KHS) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 2 *site* yakni *site* Mahasiswa dan *site* KHS. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa dan tabel Prog_studi. Pada *site* KHS tabel yang terlibat yakni tabel KHS, tabel Nilai, tabel Mata_kuliah tabel Fakultas, tabel Jurusan dan tabel Jenjang. Berikut adalah aljabar relational dari model akses *query* 4.

report ← ρR(nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←ρR(tahun_akademik, semester, id_mk, nim, k_nilai) (tahun_akademik, is_ganjil, id_mk, nim, k_nilai $^{(KHS)}$)

result←π(nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, semester, kode_mk, nama_mk, sks, nilai) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim KHS ∧ report1 ⋈ id_mk MATA_KULIAH ∧ report1 ⋈ k_nilai NILAI)

4.3.1.5 *Query* 1 Model 5

Pada model *query* 5, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MATA_KULIAH (MK), NILAI (N) dan KHS (KHS) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk equi join diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 2 *site* yakni *site* Mahasiswa dan *site* KHS. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa, tabel Jurusan dan tabel Fakultas. Pada *site* KHS tabel yang terlibat yakni tabel KHS, tabel Nilai, tabel Prog_studi, tabel Jenjang dan tabel Mata_kuliah. Berikut adalah aljabar relational dari model akses *query* 5.

report ← ρR(nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←ρR(tahun_akademik, semester, id_mk, nim, k_nilai) (tahun_akademik, is_ganjil, id_mk, nim, k_nilai $^{(KHS)}$)

result←π(nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, semester, kode_mk, nama_mk, sks, nilai) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim KHS ∧ report1 ⋈ id_mk MATA_KULIAH ∧ report1 ⋈ k_nilai NILAI)

4.3.2 *Query* 2

Perancangan model *query* 2 , query ini digunakan untuk menampilkan hasil seperti yang telah dijelaskan pada pemilihan *query* yaitu menampilkan data keungan mahasiswa. Sehingga pada proses ini memerlukan data yang berada pada tabel MAHASISWA, FAKULTAS, JURUSAN, PROGRAM STUDI, JENJANG, SELEKSI, ITEM, TRANSAKSI dan KEUANGAN. Tabel MAHASISWA memiliki relasi dengan tabel PROGRAM STUDI dan SELEKSI terletak pada atribut kode program studi dan kode seleksi, tabel PROGRAM STUDI memiliki relasi dengan JURUSAN, FAKULTAS dan JENJANG terletak pada atribut kode jurusan, kode fakultas dan kode jenjang, tabel MAHASISWA memiliki relasi dengan tabel KEUANGAN terletak pada atribut NIM, tabel KEUANGAN memiliki relasi dengan tabel ITEM dan TRANSAKSI pada atribut kode item dan kode transaksi.

a.Relational Calculus Tuple

[x1.nim, x1.nama,x1.jenjang, x1.fakultas, x1.jurusan, x1.prog_studi, x1.tahun_akademik, x1.semester,x1.transaksi, x1.status, x1.waktu, x1.debet, x1.kredit |

(∀m)(MAHASISWA (m)(m.nim =x1.nim ∧ m.nama = x1.nama ∧

(∀p)(PROG_STUDI(p)(m.k_fakultas, m.k_jurusan, m.k_prog_studi, m.k_jenjang = p.k_fakultas, p.k_jurusan, p.k_prog_studi, p.k_jenjang ∧ p.prog_studi = x1.prog_studi ∧

(∀jr)(JURUSAN(jr)(p.k_jurusan = j.k_jurusan ∧ p.k_fakultas = jr.k_fakultas ∧ j.jurusan = x1.jurusan ∧

(∀f)(FAKULTAS(f)(j.k_fakultas = f.k_fakultas ∧ f.fakultas = x1.fakultas)))) ∧

(∀jn)(JENJANG(jn)(p.k_jenjang = jn.k_jenjang ∧ j.jenjang = x1.jenjang)))) ∧

(∀k)(KEUANGAN(k)(m.nim = k.nim ∧ k.tgl_transaksi = x1.waktu ∧ k.tahun_akademik = x1.tahun_akademik ∧ k.is_ganjil = x1.semester ∧ k.debet = x1.debet ∧ k.kredit = x1.kredit ∧

(∀t)(TRANSAKSI(t)(k.k_transaksi = t.k_transaksi ∧ t.transaksi = x1.status)) ∧

(∀i)(ITEM(i)(i.k_item = k.item ∧ i.item = x1.transaksi))))))]

b. *Relational Calculus Domain*

{nim: x1 , nama: x7, jenjang: x19, fakultas: x18, jurusan: x16, prog_studi: x13, semester: x23, tahun_akademik: x22, transaksi: x31, status: x29, waktu: x21, debet: x26 , kredit: x27|

(∀x1,∀x2,∀x3,∀x4,∀x5,∀x6)(MAHASISWA(nim: x1, k_fakultas: x2, k_jenjang: x3, k_jurusan: x4, k_prog_studi: x5, k_seleksi: x6, nama: x7, angkatan: x8) ∧

(∀x9,∀x10,∀x11,∀x12)(PROG_STUDI(k_prog_studi: x9, k_jurusan: x10, k_jenjang: x11, k_fakultas: x12, prog_studi: x13) ∧ x2 = x9 ∧

(∀x14,∀x15)(JURUSAN(k_jurusan: x14, k_fakultas: x15, jurusan: x16) ∧ x10 = x14 ∧

(∀17)(FAKULTAS(k_fakultas: x17, fakultas: x18))) ∧ x15 = x17 ∧

(∀x18)(JENJANG(k_jenjang: x18, jenjang: x19))) ∧ x11 = x18 ∧

(∀x20,∀x21,∀x22,∀x23,∀x24,∀x25)(KEUANGAN(nim: x20, tgl_transaksi: x21, tahun: x22, is_ganjil: x23, k_item: x24, k_transaksi: x25, debet: x26, kredit: x27) ∧ x1 = x20 ∧

(∀x28)(TRANSAKSI(k_transaksi: x28, transaksi: x29) ∧ x25 = x28

(∀x30)(ITEM(k_item: x30, item: 31) ∧ x24 = x30))))}

4.3.2.1 *Query* 2 Model 1

Pada model *query* 1, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), KEUANGAN (K), ITEM (I) dan TRANSAKSI (T) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk equi join diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 2 *site* yakni *site* Mahasiswa dan *site* Keuangan. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa, tabel Jurusan, tabel Prog_studi, tabel Fakultas dan tabel Jenjang. Pada *site* Keuangan tabel yang terlibat yakni tabel Keuangan, tabel Item dan tabel Transaksi. Berikut adalah aljabar relational dari model akses *query* 1.

report ← $\rho_{R(nim,\ nama,\ k_jenjang,\ k_fakultas,\ k_jurusan,\ k_prog_studi)}$ (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←$\rho_{R\ (tahun_akademik,\ semester,\ k_transaksi,\ k_status,\ waktu,\ debet,\ kredit,\ nim)}$ (tahun_akademik, is_ganjil, k_item, k_transaksi, tgl_transaksi, debet, kredit, nim $^{(KEUANGAN)}$)

result←$\pi_{(nim,\ nama,\ jenjang,\ fakultas,\ jurusan,\ prog_studi,\ tahun_akademik,\ semester,\ transaksi,\ status,\ waktu,\ debet,\ kredit)}$ (report $\bowtie_{k_prog_studi,k_fakultas,k_jurusan,k_jenjang}$ PROG_STUDI ∧ PROG_STUDI $\bowtie_{k_jenjang}$ JENJANG ∧ PROG_STUDI $\bowtie_{k_jurusan}$ JURUSAN ∧ JURUSAN $\bowtie_{k_fakultas}$ FAKULTAS ∧ report $\bowtie_{nim}$ report1 ∧ report1 $\bowtie_{nim}$ KEUANGAN ∧ report1 $\bowtie_{k_item}$ ITEM ∧ report1 $\bowtie_{k_transaksi}$ TRANSAKSI)

4.3.2.2 *Query* 2 Model 2

Pada model *query* 2, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), KEUANGAN (K), ITEM (I) dan TRANSAKSI (T) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses selection ditulis dalam klausa WHERE. Melibatkan 3 *site* yakni *site* Mahasiswa, site KHS dan *site* Keuangan. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa. Pada *site* KHS tabel yang diakses adalah tabel Jurusan, tabel Prog_studi, tabel Fakultas dan tabel Jenjang. Pada *site* Keuangan tabel yang terlibat yakni tabel Keuangan, tabel Item dan tabel Transaksi. Berikut adalah aljabar relational dari model akses *query* 2.

report ← ρR(nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←ρR (tahun_akademik, semester, k_transaksi, k_status, waktu, debet, kredit, nim) (tahun_akademik, is_ganjil, k_item, k_transaksi, tgl_transaksi, debet, kredit, nim $^{(KEUANGAN)}$)

result←π (nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, semester, transaksi, status, waktu, debet, kredit) (report ⋈ k_prog_studi,k_fakultas,k_jurusan,k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim KEUANGAN ∧ report1 ⋈ k_item ITEM ∧ report1 ⋈ k_transaksi TRANSAKSI)

4.3.2.3 *Query* 2 Model 3

Pada model *query* 3, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), KEUANGAN (K), ITEM (I) dan TRANSAKSI (T) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 3 *site* yakni *site* Mahasiswa *site* KHS dan *site* Keuangan. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa dan tabel Prog_studi. Pada *site* KHS tabel yang diakses adalah tabel Jurusan, tabel Fakultas dan tabel Jenjang. Pada *site* Keuangan tabel yang terlibat yakni tabel Keuangan, tabel Item dan tabel Transaksi. Berikut adalah aljabar relational dari model akses *query* 3.

report ← ρR(nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←ρR (tahun_akademik, semester, k_transaksi, k_status, waktu, debet, kredit, nim) (tahun_akademik, is_ganjil, k_item, k_transaksi, tgl_transaksi, debet, kredit, nim $^{(KEUANGAN)}$)

result←π (nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, semester, transaksi, status, waktu, debet, kredit) (report ⋈ k_prog_studi,k_fakultas,k_jurusan,k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim KEUANGAN ∧ report1 ⋈ k_item ITEM ∧ report1 ⋈ k_transaksi TRANSAKSI)

4.3.2.4 *Query* 2 Model 4

Pada model *query* 4, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), KEUANGAN (K), ITEM (I) dan TRANSAKSI (T) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 3 *site* yakni *site* Mahasiswa, *site* KHS dan *site* Keuangan. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa, tabel Jurusan, tabel Fakultas dan tabel Jenjang. Pada *site* KHS tabel yang diakses adalah tabel Prog_studi. Pada *site* Keuangan tabel yang terlibat yakni

tabel Keuangan, tabel Item dan tabel Transaksi. Berikut adalah aljabar relational dari model akses *query* 4.

report ← ρR(nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←ρR (tahun_akademik, semester, k_transaksi, k_status, waktu, debet, kredit, nim) (tahun_akademik, is_ganjil, k_item, k_transaksi, tgl_transaksi, debet, kredit, nim $^{(KEUANGAN)}$)

result←π (nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, semester, transaksi, status, waktu, debet, kredit) (report ⋈ k_prog_studi,k_fakultas,k_jurusan,k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim KEUANGAN ∧ report1 ⋈ k_item ITEM ∧ report1 ⋈ k_transaksi TRANSAKSI)

4.3.2.5 *Query* 2 Model 5

Pada model *query* 5, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), KEUANGAN (K), ITEM (I) dan TRANSAKSI (T) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 2 *site* yakni *site* Mahasiswa dan *site* Keuangan. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa, tabel Jurusan, tabel Prog_studi, tabel Fakultas dan tabel Jenjang. Pada *site* Keuangan tabel yang terlibat yakni tabel Keuangan, tabel Mahasiswa, tabel Item dan tabel Transaksi. Berikut adalah aljabar relational dari model akses *query* 5.

report ← ρR(nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←ρR (tahun_akademik, semester, k_transaksi, k_status, waktu, debet, kredit, nim) (tahun_akademik, is_ganjil, k_item, k_transaksi, tgl_transaksi, debet, kredit, nim $^{(KEUANGAN)}$)

result←π (nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, semester, transaksi, status, waktu, debet, kredit) (report ⋈ k_prog_studi,k_fakultas,k_jurusan,k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim KEUANGAN ∧ report1 ⋈ k_item ITEM ∧ report1 ⋈ k_transaksi TRANSAKSI)

4.3.3 *Query* 3

Perancangan model akses *query* 1, *query* ini digunakan untuk menampilkan hasil seperti yang telah dijelaskan pada pemilihan *query* yaitu menampilkan data mahasiswa, biodata mahasiswa dan orang tua mahasiswa. Sehingga pada proses ini memerlukan data yang berada pada tabel MAHASISWA, FAKULTAS, JURUSAN, PROGRAM STUDI, JENJANG, SELEKSI, MAHASISWA_BIODATA dan MAHASISWA_ORTU. Tabel MAHASISWA memiliki relasi dengan tabel PROGRAM STUDI dan SELEKSI terletak pada atribut kode program studi dan kode seleksi, tabel PROGRAM STUDI memiliki relasi dengan JURUSAN, FAKULTAS dan JENJANG

terletak pada atribut kode jurusan, kode fakultas dan kode jenjang, tabel MAHASISWA memiliki relasi dengan tabel MAHASISWA_BIODATA dan MAHASIWA_ORTU yang terletak pada atribut nim.

a. Relational Calculus Tuple

[x1.nim, x1.nama,x1.jenjang, x1.fakultas, x1.jurusan, x1.prog_studi, x1.no_hp, x1.alamat_kos, x1.nama_ayah, x1.nama_ibu, x1.alamat, x1.negara, x1.propinsi, x1.kota, x1.pekerjaan_ayah, x1.gaji, x1.pekerjaan_ibu, x1.gaji |

(∀m)(MAHASISWA (m)(m.nim =x1.nim ∧ m.nama = x1.nama ∧

(∀p)(PROG_STUDI(p)(m.k_fakultas, m.k_jurusan, m.k_prog_studi, m.k_jenjang = p.k_fakultas, p.k_jurusan, p.k_prog_studi, p.k_jenjang ∧ p.prog_studi = x1.prog_studi ∧

(∀jr)(JURUSAN(jr)(p.k_jurusan = j.k_jurusan ∧ p.k_fakultas = jr.k_fakultas ∧ j.jurusan = x1.jurusan ∧

(∀f)(FAKULTAS(f)(j.k_fakultas = f.k_fakultas ∧ f.fakultas = x1.fakultas)))) ∧

(∀jn)(JENJANG(jn)(p.k_jenjang = jn.k_jenjang ∧ j.jenjang = x1.jenjang)))) ∧

(∀mb)(MAHASISWA_BIODATA(mb)(mb.nim = m.nim ∧ mb.no_hp = x1.n_hp ∧ mb.alamat_kos = x1.alamat_kos)) ∧

(∀mo)(MAHASISWA_ORTU(mo)(mo.nim = m.nim ∧ mo.nama_ayah = x1.nama_ayah ∧ mo.nama_ibu = x1.nama_ibu ∧ mo.alamat = x1.alamat ∧ mo.gaji = x1.gaji ∧ mo.gaji = x1.gaji ∧

(∀k)(KOTA(k)(mo.k_kota = k.kota ∧ k.kota = x1.kota ∧

(∀pr)(PROPINSI(pr)(k.k_propinsi = pr.k_propinsi ∧ pr.propinsi = x1.propinsi ∧

(∀n)(NEGARA(n)(mo.k_negara = n.k_negara ∧ n.negara = x1.negara)))))) ∧

(∀pk)(PEKERJAAN(pk)(mo.k_pekerjaan_ayah = pk.k_pekerjaan ∧ mo.k_pekerjaan_ibu = pk.k_pekerjaan ∧ pk.pekerjaan = x1.pekerjaan_ayah ∧ pk.pekerjaan = x1.pekerjaan_ibu))))))]

b. Relational Calculus Domain

{nim: x1 , nama: x7, jenjang: x19, fakultas: x18, jurusan: x16, prog_studi: x13, no_hp: x29, alamat_kos: x28, nama_ayah: x33, nama_ibu: x36, alamat: x42, negara: x55, propinsi: x53, kota: x50, pekerjaan_ayah: x57, gaji: x35, pekerjaan_ibu: x57, gaji: x38|

(∀x1,∀x2,∀x3,∀x4,∀x5,∀x6)(MAHASISWA(nim: x1, k_fakultas: x2, k_jenjang: x3, k_jurusan: x4, k_prog_studi: x5, k_seleksi: x6, nama: x7, angkatan: x8) ∧

(∀x9,∀x10,∀x11,∀x12)(PROG_STUDI(k_prog_studi: x9, k_jurusan: x10, k_jenjang: x11, k_fakultas: x12, prog_studi: x13) ∧ x2 = x9 ∧

(∀x14,∀x15)(JURUSAN(k_jurusan: x14, k_fakultas: x15, jurusan: x16) ∧ x10 = x14 ∧

(∀17)(FAKULTAS(k_fakultas: x17, fakultas: x18))) ∧ x15 = x17 ∧

(∀x18)(JENJANG(k_jenjang: x18, jenjang: x19))) ∧ x11 = x18 ∧

(∀x)(MAHASISWA_BIODATA (nim: x20, k_negara: x21, k_propinsi: x22, k_kota: x23, k_sekolah: x24, tmp_lahir: x25, tgl_lahir: x26, alamat_asal: x27, alamat_kos: x28, no_hp: x29, no_telp_rmh: x30, no_telp_kos: x31)) ∧ x20 = x1 ∧

(∀x42) (MAHASISWA_ORTU (nim: x32, nama_ayah: x33, k_pekerjaan_ayah: x34, penghasilan_ayah: x35, nama_ibu: x36, k_pekerjaan_ibu: x37, penghasilan_ibu: x38, k_negara: x39, k_propinsi: x40, k_kota: x41, alamat: x42, n_telpon: x43, no_hp: x46) ∧ x32 = x1

(∀)(KOTA(k_kota: x47, k_negara: x48, k_propinsi: x49, kota: x50) ∧ x41 = x47 ∧

()(PROPINSI(k_propinsi: x51, k_negara: x52, propinsi: x53) ∧ x40 = x51 ∧

()(NEGARA(k_negara: x54, negara: x55) ∧ x39 = x54 ∧

()(PEKERJAAN(k_pekerjaan: x56, pekerjaan: x57) ∧ x34 = x56 ∧ x37 = x56))))))}

4.3.3.1 *Query* 3 Model 1

Pada model *query* 1, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MAHASISWA_BIODATA (MB), MAHASISWA_ORTU (MO), NEGARA (N), PROPINSI (PR) dan KOTA (K) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 3 *site* yakni *site* Mahasiswa, *site* Mhs_bio dan *site* Mhs_ortu. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa, tabel Jurusan, tabel Fakultas, tabel Prog_studi dan tabel Jenjang. Pada *site* Mhs_bio tabel yang diakses adalah Mhs_bio. Pada *site* Mhs_ortu tabel yang terlibat yakni tabel Mhs_ortu, tabel Kota, tabel Propinsi, tabel Negara, tabel Pekerjaan dan tabel Pekerjaan. Berikut adalah aljabar relational dari model akses *query* 1.

report ← ρR (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←Pr (nim, no_hp, alamat_kos) (nim, no_hp, alamat_kos $^{(MAHASISWA_BIODATA)}$)

report2←ρR (nim, nama_ayah, nama_ibu, alamat, k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji) (nim, nama_ayah, nama_ibu, alamat, k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji $^{(MAHASISWA_ORTU)}$)

result←π (nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_ayah, nama_ibu, alamat, negara, propinsi, kota, pekerjaan_ayah, gaji, pekerjaan_ibu, gaji) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim MAHASISWA_BIODATA∧ report ⋈ nim report2 ∧ report2 ⋈ nim MAHASISWA_ORTU ∧ report2 ⋈ k_kota KOTA ∧ KOTA ⋈ k_propinsi PROPINSI ∧ PROPINSI ⋈ k_negara NEGARA ∧ report2 ⋈ k_pekerjaan PEKERJAAN, report ⋈ k_pekerjaan PEKERJAAN)

4.3.3.2 *Query* 3 Model 2

Pada model *query* 2, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MAHASISWA_BIODATA (MB), MAHASISWA_ORTU (MO), NEGARA (N), PROPINSI (PR) dan KOTA (K) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 3 *site* yakni *site* Mahasiswa, *site* Mhs_bio dan *site* Mhs_ortu. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa, tabel Jurusan, tabel Fakultas, tabel Prog_studi dan tabel Jenjang. Pada *site* Mhs_bio tabel yang diakses adalah Mhs_bio, tabel Kota, tabel Propinsi dan tabel Negara. Pada *site* Mhs_ortu tabel yang terlibat yakni tabel Mhs_ortu, tabel Pekerjaan dan tabel Pekerjaan. Berikut adalah aljabar relational dari model akses *query* 2.

report ← ρR (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←Pr (nim, no_hp, alamat_kos) (nim, no_hp, alamat_kos $^{(MAHASISWA_BIODATA)}$)

report2←ρR (nim, nama_ayah, nama_ibu, alamat, k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji) (nim, nama_ayah, nama_ibu, alamat, k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji $^{(MAHASISWA_ORTU)}$)

result←π (nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_ayah, nama_ibu, alamat, negara, propinsi, kota, pekerjaan_ayah, gaji, pekerjaan_ibu, gaji) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim MAHASISWA_BIODATA∧ report ⋈ nim report2 ∧ report2 ⋈ nim MAHASISWA_ORTU ∧ report2 ⋈ k_kota KOTA ∧ KOTA ⋈ k_propinsi PROPINSI ∧ PROPINSI ⋈ k_negara NEGARA ∧ report2 ⋈ k_pekerjaan PEKERJAAN, report ⋈ k_pekerjaan PEKERJAAN)

4.3.3.3 *Query* 3 Model 3

Pada model *query* 3, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MAHASISWA_BIODATA (MB), MAHASISWA_ORTU (MO), NEGARA (N), PROPINSI (PR) dan KOTA (K) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis

dalam klausa WHERE. Melibatkan 4 *site* yakni *site* Mahasiswa, *site* KHS, *site* Mhs_bio dan *site* Mhs_ortu. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa. Pada *site* KHS tabel yang diakses adalah tabel Jurusan, tabel Fakultas, tabel Prog_studi dan tabel Jenjang. Pada *site* Mhs_bio tabel yang diakses adalah Mhs_bio , tabel Kota, tabel Propinsi dan tabel Negara. Pada *site* Mhs_ortu tabel yang terlibat yakni tabel Mhs_ortu, tabel Pekerjaan dan tabel Pekerjaan. Berikut adalah aljabar relational dari model akses *query* 3.

report ← ρR (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←Pr (nim, no_hp, alamat_kos) (nim, no_hp, alamat_kos $^{(MAHASISWA_BIODATA)}$)

report2←ρR (nim, nama_ayah, nama_ibu, alamat, k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji) (nim, nama_ayah, nama_ibu, alamat, k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji $^{(MAHASISWA_ORTU)}$)

result←π (nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_ayah, nama_ibu, alamat, negara, propinsi, kota, pekerjaan_ayah, gaji, pekerjaan_ibu, gaji) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim MAHASISWA_BIODATA∧ report ⋈ nim report2 ∧ report2 ⋈ nim MAHASISWA_ORTU ∧ report2 ⋈ k_kota KOTA ∧ KOTA ⋈ k_propinsi PROPINSI ∧ PROPINSI ⋈ k_negara NEGARA ∧ report2 ⋈ k_pekerjaan PEKERJAAN, report ⋈ k_pekerjaan PEKERJAAN)

4.3.3.4 *Query* 3 Model 4

Pada model *query* 4, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MAHASISWA_BIODATA (MB), MAHASISWA_ORTU (MO), NEGARA (N), PROPINSI (PR) dan KOTA (K) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 4 *site* yakni *site* Mahasiswa, *site* KHS, *site* Mhs_bio dan *site* Mhs_ortu. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa dan tabel Prog_studi. Pada *site* KHS tabel yang diakses adalah tabel Jurusan, tabel Fakultas, dan tabel Jenjang. Pada *site* Mhs_bio tabel yang diakses adalah Mhs_bio , tabel Kota, tabel Propinsi dan tabel Negara. Pada *site* Mhs_ortu tabel yang terlibat yakni tabel Mhs_ortu, tabel Pekerjaan dan tabel Pekerjaan. Berikut adalah aljabar relational dari model akses *query* 4.

report ← ρR (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi $^{(MAHASISWA)}$)

report1←Pr (nim, no_hp, alamat_kos) (nim, no_hp, alamat_kos $^{(MAHASISWA_BIODATA)}$)

report2←ρR (nim, nama_ayah, nama_ibu, alamat, k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji) (nim, nama_ayah, nama_ibu, alamat,

k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji (MAHASISWA_ORTU))

result←π (nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_ayah, nama_ibu, alamat, negara, propinsi, kota, pekerjaan_ayah, gaji, pekerjaan_ibu, gaji) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim MAHASISWA_BIODATA∧ report ⋈ nim report2 ∧ report2 ⋈ nim MAHASISWA_ORTU ∧ report2 ⋈ k_kota KOTA ∧ KOTA ⋈ k_propinsi PROPINSI ∧ PROPINSI ⋈ k_negara NEGARA ∧ report2 ⋈ k_pekerjaan PEKERJAAN, report ⋈ k_pekerjaan PEKERJAAN)

4.3.3.5 *Query* 3 Model 5

Pada model *query* 5, proses *selection* dari tabel MAHASISWA (M), FAKULTAS (F), JURUSAN (J), PROGRAM_STUDI (P), JENJANG (JN), MAHASISWA_BIODATA (MB), MAHASISWA_ORTU (MO), NEGARA (N), PROPINSI (PR) dan KOTA (K) dilakukan dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Melibatkan 3 *site* yakni *site* Mahasiswa, *site* Mhs_bio dan *site* Mhs_ortu. Pada *site* Mahasiwa tabel yang diakses adalah tabel Mahasiswa, tabel Jurusan, tabel Fakultas, tabel Prog_studi dan tabel Jenjang. Pada *site* Mhs_bio tabel yang diakses adalah Mhs_bio, tabel Kota dan tabel Propinsi. Pada *site* Mhs_ortu tabel yang terlibat yakni tabel Mhs_ortu, tabel Negara, tabel Pekerjaan dan tabel Pekerjaan. Berikut adalah aljabar relational dari model akses *query* 5.

report ← ρR (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi) (nim, nama, k_jenjang, k_fakultas, k_jurusan, k_prog_studi (MAHASISWA))

report1←Pr (nim, no_hp, alamat_kos) (nim, no_hp, alamat_kos (MAHASISWA_BIODATA))

report2←ρR (nim, nama_ayah, nama_ibu, alamat, k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji) (nim, nama_ayah, nama_ibu, alamat, k_negara, k_propinsi, k_kota, k_pekerjaan_ayah, gaji, k_pekerjaan_ibu, gaji (MAHASISWA_ORTU))

result←π (nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_ayah, nama_ibu, alamat, negara, propinsi, kota, pekerjaan_ayah, gaji, pekerjaan_ibu, gaji) (report ⋈ k_prog_studi, k_fakultas, k_jurusan, k_jenjang PROG_STUDI ∧ PROG_STUDI ⋈ k_jenjang JENJANG ∧ PROG_STUDI ⋈ k_jurusan JURUSAN ∧ JURUSAN ⋈ k_fakultas FAKULTAS ∧ report ⋈ nim report1 ∧ report1 ⋈ nim MAHASISWA_BIODATA∧ report ⋈ nim report2 ∧ report2 ⋈ nim MAHASISWA_ORTU ∧ report2 ⋈ k_kota KOTA ∧ KOTA ⋈ k_propinsi PROPINSI ∧ PROPINSI ⋈ k_negara NEGARA ∧ report2 ⋈ k_pekerjaan PEKERJAAN, report ⋈ k_pekerjaan PEKERJAAN)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.4 Penerapan *Teacher Learner Based Optimization*

Langkah-langkah penerapan metode TLBO adalah sebagai berikut :

1. Mengidentifikasi parameter yang digunakan pada penelitian

   Penerapan metode TLBO diawali dengan inisialisasi parameter optimasi. Menentukan parameter untuk penerapan algoritme TLBO. Terdapat dua parameter yang digunakan yakni parameter heuristik dan parameter meta-heuristik.

   1) Parameter Meta-heuristik adalah parameter yang dapat diterapkan untuk berbagai permasalahan karena parameter meta-heuristik bersifat *problem independent*. Parameter metaheuristik yang digunakan sebagai berikut :

      a. Jumlah populasi (Pn), dalam penelitian ini kita memiliki 5 populasi untuk setiap *query* yang dipilih. Terdapat 3 rancangan *query* yang dipilih, sehingga terdapat 15 populasi yang akan diuji.

      b. Jumlah generasi (Gn), dalam penelitian ini terdapat 2 kali *update* populasi atau pembuatan generasi baru. Untuk *update* populasi 01 dilakukan dengan cara menggunakan guru terbaik dari *query plan* yang akan dikalkulasi dengan setiap *query* yang ada pada *query plan*. Untuk *update* populasi 02 dilakukan dengan cara saling berinteraksi antar *query plan* pada *update* populasi 01 sehingga kan menghasilkan *update* populasi 02.

   2) Parameter heuristik adalah parameter yang hanya memang dirancang untuk permasalahan yang sedang diteliti karena parameter heuristik bersifat *problem dependent* sehingga untuk satu permasalahan penelitian dengan permasalahan penelitian yang lain berbeda. Parameter heuristik yang digunakan mengacu pada jurnal yang diacu terdapat pada bab 2 sub bab 2.9.3 desain parameter heuristik yaitu :

      *a. Query Affinity Cost (*QAC*)*

      *b. Query Localization Cost (*QLC*)*

      *c. Local Processing Cost (*LPC*)*

   Selanjutnya ukuran populasi mendefinisikan jumlah *learner*. *Learner* disini adalah *query plan*.

2. Menginisialisasi populasi atau QEP berdasar query yang dipilih

   Pada tahap ini dipilih data yang akan diakses oleh *query*. Selanjutnya akan mendefinisikan 5 *query plan* untuk mengakses data tersebut. Sebelumnya dibuat *Relation Site Matrix* (RSM) untuk menunjukkan relasi pada setiap *site* yang ditunjukkan pada tabel 4.3.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.3 Matrik relasi dan situs

SITE/ RELASI	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19
S1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
S2	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
S3	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
S4	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
S5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Keterangan :

R1 : Tabel Mahasiswa
R2 : Tabel Fakultas
R3 : Tabel Jurusan
R4 : Tabel Jenjang
R5 : Tabel Seleksi
R6 : Tabel Program_Studi
R7 : Tabel Mahasiswa_Biodata
R8 : Tabel Sekolah
R9 : Tabel Kota
R10 : Tabel Propinsi
R11 : Tabel Negara
R12 : Tabel Mahasiswa_Ortu
R13 : Tabel Pekerjaan
R14 : Tabel Nilai
R15 : Tabel KHS
R16 : Tabel Mata_Kuliah
R17 : Tabel Keuangan
R18 : Tabel Item
R19 : Tabel Transaksi
S1 : Site Mahasiswa
S2 : Site Mahasiswa Biodata
S3 : Site Biodata Orang Tua Mahasiswa
S4 : Site KHS Mahasiswa
S5 : Site Keuangan

Dari RSM yang telah dibuat, selanjutnya inisialisasi *query plan*. Terdapat 3 *query* yang akan digunakan dan setiap *query* memiliki 5 QEP yang secara rinci dijelaskan pada tabel 4.4 untuk *query* 1, pada tabel 4.5 untuk *query* 2 dan pada tabel 4.6 untuk *query* 1. Pada setiap tabel tersebut QEP ditulis dengan format pada QEP terdapat berapa relasi, dan setiap relasi dituliskan nomor *site* yang terdapat relasi yang terlibat tersebut.

Tabel 4.4 *Query Plan Query* 1

Population (Query Plan)	
[R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18]	
1	[1 1 1 1 1 4 4 4]
2	[1 4 4 4 4 4 4 4]
3	[1 1 1 1 4 4 4 4]
4	[1 1 4 4 4 4 4 4]
5	[1 1 1 4 4 4 4 4]

Tabel 4.5 *Query Plan Query* 2

Population (Query Plan)	
[R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18]	
1	[1 1 1 1 1 5 5 5]
2	[1 4 4 4 4 5 5 5]
3	[1 1 4 4 4 5 5 5]
4	[1 1 1 1 4 5 5 5]
5	[1 1 1 1 1 5 5 5 5]

Tabel 4.6 *Query Plan Query* 3

Population (Query Plan)	
[R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18]	
1	[1 1 1 1 1 2 3 3 3 3 3 3]
2	[1 1 1 1 1 2 2 2 2 3 3 3]
3	[1 2 2 2 2 3 3 3 4 4 4 4]
4	[1 1 2 2 2 3 3 3 3 4 4 4]
5	[1 1 1 1 1 2 2 2 3 3 3 3]

3. Menghitung nilai parameter heuristik pada setiap *query plan*

Setiap QEP pada *query* 1, *query* 2 dan *query* 3 dihitung dengan parameter heuristik yakni QAC, QLC dan LPC. Untuk perhitungan parameter QAC menggunakan persamaan 2.13, untuk parameter QLC menggunakan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

persamaan 2.14 dan parameter LPC menggunakan persamaan 2.15, 2.16, 2.17 dan 2.18.

4. Membuat tabel dengan kolom menunjukkan parameter (QAC, QLC dan LPC) dan baris menunjukkan nilai parameter untuk setiap QEP.
5. Menerapkan metode yang digunakan pada penelitian ini yakni TLBO

Fase ini dibagi menjadi dua tahap yakni pada tahap *teacher* dari populasi data *testing* yang ada kita pilih *learner* yang paling baik untuk menjadi *teacher* dan digunakan untuk mengupdate populasi. Selanjutnya pada tahap kedua yakni tahap *learner*, siswa atau *learner* meningkatkan pengetahuan mereka dengan berinteraksi dengan siswa lain untuk menghasilkan populasi terupdate yang kedua.

Fase Teacher

i. Menentukan nilai mean dari setiap kolom yang ada.
ii. Menentukan *Difference Factor (TF)* dari setiap kolom (QAC, QLC, LPC) dengan menggunakan persamaan 2.10 dan 2.11.
iii. Menentukan *best teacher query plan* dengan *fitness* terkecil. Nilai *fitness* adalah nilai yang mampu mewakili ketiga parameter, sehingga disini nilai *fitness* adalah nilai rata-rata setiap kolom (QAC, QLC, LPC) untuk masing-masing baris (QEP).
iv. *Update* populasi 1 berdasarkan *TF* dengan menggunakan persamaan 2.12.

Fase Learner

i. Menentukan nilai *fitness* untuk populasi 1. Nilai fitness adalah nilai yang mampu mewakili ketiga parameter, sehingga disini nilai *fitness* adalah nilai rata-rata setiap kolom (QAC, QLC, LPC) untuk masing-masing baris (QEP).
ii. Melakukan interaksi *query plan* 1 dengan *query plan* lain dst. Untuk meningkatkan murid yang lemah. Proses interaksi antar *learner* dengan simulasi (1 dengan 2, 3, 4, 5 atau 2 dengan 3, 4, 5 dst), dan proses interaksi antar *learner* menggunakan persamaan yang ada dalam gambar 2.22.
iii. Dari hasil perhitungan menghasilkan *update* populasi ke 2 berdasar interaksi antar *learner*

BAB 5 IMPLEMENTASI

Pada bab ini akan dijelaskan implementasi dari penelitian yang dilakukan. Proses implementasi ini membutuhkan media penunjang seperti perangkat lunak, perangkat keras hingga implementasi *query* agar dapat dilakukan pengujian terhadap objek penelitian.

5.1 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam implementasi penelitian ini adalah sebagai berikut :

1. Sistem Operasi Windows 7 64 bit.
2. Sistem Operasi *centos* 6.5 64bit.
3. VMware Workstation versi 11.
4. IBM DB2 expc 64 bit
5. IBM Data Studio
6. Microsoft Exel
7. DB Designer

5.2 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam implementasi penelitian ini adalah sebagai berikut :

1. Komputer Dekstop core
2. Memori DDR3 8 GB

5.3 Implementasi *Virtual Server (Site)*

Dalam penelitian ini, dibentuk *virtual server* dengan menggunakan aplikasi VMware 10. Spesifikasi yang diterapkan pada setiap *virtual server* yang dibuat adala sebagai berikut :

1. Sistem Operasi Centos 64 bit.
2. Single Core.
3. Memori (RAM) 1GB.
4. Disk 100 GB.
5. Network connection NAT.

Virtual server yang dibuat sebanyak lima *virtual server* dan dijalankan menggunakan arsitektur jaringan *network address translation* dengan tujuan agar setiap virtual juga dapat terkoneksi ke jaringan internet untuk mengunduh paket tambahan installasi IBM DB2 expc. Setelah seluruh *virtual* terbentuk kemudian

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

dikonfigurasi lebih lanjut agar dapat saling berkomunikasi antar *virtual*. Dalam proses untuk saling menghubungkan akan terbentuk *IP address* yang digunakan oleh masing-masing *virtual* sebagai *site* yang ditunjukkan pada tabel 5.1.

Tabel 5.1 Daftar IP *Adrress Virtual Server*

***Virtual* Servel**	**IP Address**
Site 1	192.168.10.129
Site 2	192.168.10.130
Site 3	192.168.10.131
Site 4	192.168.10.132
Site 5	192.168.10.133

5.4 Implementasi Basis Data

Dalam penelitian ini, dilakukan implementasi basis data dengan menggunakan DBMS IBM DB2 expc 64 bit. Setelah DBMS terinstall pada masing-masing *site*, selanjutnya dilakukan pembangunan basis data. Dalam pembangunannya dibentuk beberapa tabel sebagai pengelompokan data dan tabel yang dibentuk antara lain tabel fakultas, tabel jurusan, tabel prog_studi, tabel jenjang, tabel seleksi, tabel mahasiswa, tabel item, tabel transaksi, tabel keuangan, tabel nilai, tabel khs, tabel mata_kuliah, tabel pekerjaan, tabel negara, tabel propinsi, tabel kota, tabel sekolah, tabel mahasiswa_ortu, dan tabel mahasiswa_biodata.

Dalam implementasi basis data, dilakukan implementasi *query create* tabel pada setiap *site* karena kebutuhan tabel pada setiap *site* berbeda-beda. Pada keselurhan *site* terdapat satu tabel yang pasti dimiliki yaitu tabel mahasiswa yang merupakan tabel *primary* atau relasi yang kuat pada skema.

5.4.1 Implementasi *Site*

Dalam basis data terdistribusi, pada *site* 1 merupakan basis data yang menyimpan data mahasiswa. Data mahasiswa tersebut meliputi beberapa tabel yaitu tabel fakultas, tabel jurusan, tabel prog_studi, tabel jenjang, tabel seleksi dan tabel mahasiswa.

Dalam basis data terdistribusi, pada *site* 2 merupakan basis data yang menyimpan data biodata mahasiswa. Data biodata mahasiswa tersebut meliputi beberapa tabel yaitu tabel mahasiswa, tabel negara, tabel propinsi, tabel kota, tabel sekolah dan tabel mahasiswa_biodata.

Dalam basis data terdistribusi, pada *site* 3 merupakan basis data yang menyimpan data biodata orang tua mahasiswa. Data biodata orang tua mahasiswa tersebut meliputi beberapa tabel yaitu tabel mahasiswa, tabel pekerjaan, tabel negara, tabel propinsi, tabel kota dan tabel mahasiswa_ortu.

Dalam basis data terdistribusi, pada *site* 4 merupakan basis data yang menyimpan data KHS mahasiswa. Data KHS mahasiswa tersebut meliputi

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

beberapa tabel yaitu tabel fakultas, tabel jurusan, tabel prog_studi, tabel jenjang, tabel seleksi, tabel mahasiswa, tabel nilai, tabel KHS dan tabel mata_kuliah.

Dalam basis data terdistribusi, pada *site* 5 merupakan basis data yang menyimpan data keuangan mahasiswa. Data keuangan mahasiswa tersebut meliputi beberapa tabel yaitu tabel mahasiswa, tabel item, tabel transaksi dan tabel keuangan. Pada tabel 5.2 adalah persebaran tabel pada setiap *site*.

Tabel 5.2 Persebaran Tabel pada setiap *Site*

Nama Tabel	*Site* 1 Mahsiswa	*Site* 2 Mahasiswa Biodata	*Site* 3 Mahasiswa Ortu	*Site* 4 KHS	*Site* 5 Keuangan
Tabel Mahasiswa	✓	✓	✓	✓	✓
Tabel Fakultas	✓	-	-	✓	-
Tabel Jurusan	✓	-	-	✓	-
Tabel Jenjang	✓	-	-	✓	-
Tabel Prog_studi	✓	-	-	✓	-
Tabel Seleksi	✓	-	-	✓	-
Tabel Mata_Kuliah	-	-	-	✓	
Tabel Item	-	-	-	-	✓
Tabel Transaksi	-	-	-	-	✓
Tabel Keuangan	-	-	-	-	✓
Tabel KHS	-	-		✓	-
Tabel Mahasiswa_ortu	-	-	✓	-	-
Tabel Mahasiswa_biodata	-	✓	-	-	-
Tabel Pekerjaan	-	-	✓	-	-
Tabel Kota	-	✓	✓	-	-
Tabel Negara	-	✓	✓	-	-
Tabel Propinsi	-	✓	✓	-	-
Tabel Sekolah	-	✓	-	-	-
Tabel Nilai	-	-	-	✓	

5.4.2 Implementas Tabel

5.4.2.1 Tabel Fakultas

Dalam basis data akan dibentuk tabel fakultas. Tabel ini memiliki dua atribut yaitu k_fakultas yang menyimpan kode fakultas dan fakultas yang berisi data nama fakultas. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.1.

```
1  CREATE TABLE FAKULTAS (
2    K_FAKULTAS VARCHAR(3) NOT NULL,
3    FAKULTAS VARCHAR(100) NULL,
4    PRIMARY KEY(K_FAKULTAS)
5  )
6  IN REGTSPACE
7  INDEX IN IDXTSPACE
8  LONG IN LONGTSPACE;
```

Gambar 5.1 *Source Create Table* FAKULTAS

Berikut adalah keterangan dari *query* pembentukan tabel fakultas yang ditunjukkan dalam gambar 5.1 :

Baris 1 : pembukaan perintah membuat tabel dengan nama FAKULTAS.

Baris 2 : K_FAKULTAS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : FAKULTAS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (100) dan kondisi boleh kosong (*null*).

Baris 4 : perintah untuk menjadikan atribut K_FAKULTAS sebagai kunci utama (*primary key*).

Baris 5 : tanda penutup untuk pembentukan atribut tabel.

Baris 6 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 7 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.2 Tabel Jurusan

Dalam basis data akan dibentuk tabel jurusan. Tabel ini memiliki tiga atribut yaitu k_jurusan yang menyimpan kode jurusan, jurusan yang berisi data nama jurusan dan k_fakultas yang berisi kode fakultas. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.2.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
1   CREATE TABLE JURUSAN (
2     K_JURUSAN VARCHAR(3) NOT NULL,
3     K_FAKULTAS VARCHAR(3) NOT NULL,
4     JURUSAN VARCHAR(45),
5     PRIMARY KEY(K_JURUSAN, K_FAKULTAS),
6     FOREIGN KEY (K_FAKULTAS) REFERENCES FAKULTAS (K_FAKULTAS)
7   )
8   IN REGTSPACE
9   INDEX IN IDXTSPACE
10  LONG IN LONGTSPACE;
```

Gambar 5.2 *Source Create Table* JURUSAN

Berikut adalah keterangan dari *query* pembentukan tabel jurusan yang ditunjukkan dalam gambar 5.2 :

Baris 1 : pembukaan perintah membuat tabel dengan nama JURUSAN.

Baris 2 : K_JURUSAN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : K_FAKULTAS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : JURUSAN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (100) dan kondisi boleh kosong (*null*).

Baris 5 : perintah untuk menjadikan atribut K_JURUSAN dan K_FAKULTAS sebagai kunci utama (*primary key*).

Baris 6 : perintah referensi data pada atribut K_FAKULTAS sebagai *foreign key* yang mengarah pada tabel FAKULTAS.

Baris 7 : tanda penutup untuk pembentukan atribut tabel.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 9 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 10 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.3 Tabel Program Studi

Dalam basis data akan dibentuk tabel prog_studi. Tabel ini memiliki lima atribut yaitu k_prog_studi yang menyimpan kode prog_studi, prog_studi yang berisi data nama program studi, k_jurusan yang berisi data kode jurusan, k_jenjang yang berisi data kode jenjang dan k_fakultas yang berisi data kode fakultas. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.3.

```
1   CREATE TABLE PROG_STUDI (
2     K_PROG_STUDI VARCHAR(3) NOT NULL,
3     K_JURUSAN VARCHAR(3) NOT NULL,
4     K_JENJANG VARCHAR(3) NOT NULL,
5     K_FAKULTAS VARCHAR(3) NOT NULL,
6     PROG_STUDI VARCHAR(45),
7     PRIMARY KEY(K_PROG_STUDI, K_JURUSAN, K_JENJANG, K_FAKULTAS),
8     FOREIGN KEY (K_FAKULTAS, K_JURUSAN) REFERENCES JURUSAN
    (K_FAKULTAS, K_JURUSAN),
9     FOREIGN KEY (K_JENJANG) REFERENCES JENJANG (K_JENJANG)
10  )
11  IN REGTSPACE
12  INDEX IN IDXTSPACE
13  LONG IN LONGTSPACE;
```

Gambar 5.3 *Source Create Table* PROG_STUDI

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.3 :

Baris 1 : pembukaan perintah membuat tabel dengan nama PROG_STUDI.

Baris 2 : K_PROG_STUDI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : K_JURUSAN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : K_JENJANG adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : K_FAKULTAS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 6 : PROG_STUDI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi boleh kosong (*null*).

Baris 7 : perintah untuk menjadikan atribut K_JENJANG, K_PROG_STUDI, K_JURUSAN dan K_FAKULTAS sebagai kunci utama (*primary key*).

Baris 8 : perintah referensi data pada atribut K_FAKULTAS dan K_JURUSAN sebagai *foreign key* yang mengarah pada tabel JURUSAN.

Baris 9 : perintah referensi data pada atribut K_JENJANG sebagai *foreign key* yang mengarah pada tabel JENJANG.

Baris 10 : tanda penutup untuk pembentukan atribut tabel.

Baris 11 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 12 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 13 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.4 Tabel Jenjang

Dalam basis data akan dibentuk tabel jenjang. Tabel ini memiliki dua atribut yaitu k_jenjang yang menyimpan kode jenjang dan jenjang yang berisi data nama jenjang. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.4.

```
1  CREATE TABLE JENJANG (
2    K_JENJANG VARCHAR(3) NOT NULL,
3    JENJANG VARCHAR(25),
4    PRIMARY KEY(K_JENJANG)
5  )
6  IN REGTSPACE
7  INDEX IN IDXTSPACE
8  LONG IN LONGTSPACE;
```

Gambar 5.4 *Source Create Table* JENJANG

Berikut adalah keterangan dari *query* pembentukan tabel fakultas yang ditunjukkan dalam gambar 5.4 :

Baris 1 : pembukaan perintah membuat tabel dengan nama JENJANG.

Baris 2 : K_JENJANG adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : JENJANG adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (25) dan kondisi boleh kosong (*null*).

Baris 4 : perintah untuk menjadikan atribut K_JENJANG sebagai kunci utama (*primary key*).

Baris 5 : tanda penutup untuk pembentukan atribut tabel.

Baris 6 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 7 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.5 Tabel Seleksi

Dalam basis data akan dibentuk tabel seleksi. Tabel ini memiliki dua atribut yaitu k_seleksi yang menyimpan kode seleksi dan seleksi yang berisi data nama seleksi. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.5.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
1  CREATE TABLE SELEKSI (
2    K_SELEKSI VARCHAR(3) NOT NULL,
3    SELEKSI VARCHAR(45),
4    PRIMARY KEY(K_SELEKSI)
5  )
6  IN REGTSPACE
7  INDEX IN IDXTSPACE
8  LONG IN LONGTSPACE;
```

Gambar 5.5 *Source Create Table* SELEKSI

Berikut adalah keterangan dari *query* pembentukan tabel fakultas yang ditunjukkan dalam gambar 5.5 :

Baris 1 : pembukaan perintah membuat tabel dengan nama SELEKSI.

Baris 2 : K_SELEKSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : SELEKSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi boleh kosong (*null*).

Baris 4 : perintah untuk menjadikan atribut K_SELEKSI sebagai kunci utama (*primary key*).

Baris 5 : tanda penutup untuk pembentukan atribut tabel.

Baris 6 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 7 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.6 Tabel Mahasiswa

Dalam basis data akan dibentuk tabel mahasiswa. Tabel ini memiliki delapan atribut yaitu nim yang menyimpan kode mahasiswa, nama yang berisi data nama mahasiswa, angkatan yang berisi data angkatan, k_fakultas yang berisi data kode fakultas, k_jurusan yang berisi data kode jurusan, k_prog_studi yang berisi kode program studi, k_jenjang yang berisi kode jenjang dan k_seleksi yang berisi kode seleksi dari mahasiswa. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.6.

```
1  CREATE TABLE MAHASISWA (
2    NIM VARCHAR(15) NOT NULL,
3    K_FAKULTAS VARCHAR(3) NOT NULL,
4    K_JENJANG VARCHAR(3) NOT NULL,
5    K_JURUSAN VARCHAR(3) NOT NULL,
6    K_PROG_STUDI VARCHAR(3) NOT NULL,
7    K_SELEKSI VARCHAR(3) NOT NULL,
8    NAMA VARCHAR(100),
9    ANGKATAN SMALLINT,
10   PRIMARY KEY(NIM),
11   FOREIGN KEY (K_FAKULTAS, K_JURUSAN, K_JENJANG, K_PROG_STUDI)
     REFERENCES JURUSAN (K_FAKULTAS, K_JURUSAN, K_JENJANG,
     K_PROG_STUDI),
12   FOREIGN KEY (K_SELEKSI) REFERENCES SELEKSI (K_SEKLEKSI)
13 )
14 IN REGTSPACE
15 INDEX IN IDXTSPACE
16 LONG IN LONGTSPACE;
```

Gambar 5.6 *Source Create Table* MAHASISWA

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.6 :

Baris 1 : pembukaan perintah membuat tabel dengan nama MAHASISWA.

Baris 2 : NIM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (15) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : K_FAKULTAS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : K_JENJANG adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : K_JURUSAN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 6 : K_PROG_STUDI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 7 : K_SELEKSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 8 : NAMA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (100) .

Baris 9 : ANGKATAN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *smallint* .

Baris 10 : perintah untuk menjadikan atribut NIM sebagai kunci utama (*primary key*).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 11 : perintah referensi data pada atribut K_FAKULTAS, K_JENJANG, K_PROG_STUDI dan K_JURUSAN sebagai *foreign key* yang mengarah pada tabel JURUSAN.

Baris 12 : perintah referensi data pada atribut K_SELEKSI sebagai *foreign key* yang mengarah pada tabel SELEKSI.

Baris 13 : tanda penutup untuk pembentukan atribut tabel.

Baris 14 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 15 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 16 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.7 Tabel Negara

Dalam basis data akan dibentuk tabel negara. Tabel ini memiliki dua atribut yaitu k_negara yang menyimpan data kode negara dan negara yang berisi data nama negara. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.7.

```
1  CREATE TABLE NEGARA (
2    K_NEGARA VARCHAR(3) NOT NULL,
3    NEGARA VARCHAR(45) ,
4    PRIMARY KEY(K_NEGARA)
5  )
6  IN REGTSPACE
7  INDEX IN IDXTSPACE
8  LONG IN LONGTSPACE;
```

Gambar 5.7 *Source Create Table* NEGARA

Berikut adalah keterangan dari *query* pembentukan tabel fakultas yang ditunjukkan dalam gambar 5.7 :

Baris 1 : pembukaan perintah membuat tabel dengan nama NEGARA.

Baris 2 : K_NEGARA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : NEGARA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi boleh kosong (*null*).

Baris 4 : perintah untuk menjadikan atribut K_NEGARA sebagai kunci utama (*primary key*).

Baris 5 : tanda penutup untuk pembentukan atribut tabel.

Baris 6 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 7 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.8 Tabel Propinsi

Dalam basis data akan dibentuk tabel propinsi. Tabel ini memiliki tiga atribut yaitu k_propinsi yang menyimpan data kode propinsi, k_negara yang berisi data kode negara dan propinsi yang berisi data nama propinsi. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.8.

```
1  CREATE TABLE PROPINSI (
2    K_PROPINSI VARCHAR(3) NOT NULL,
3    K_NEGARA VARCHAR(3) NOT NULL,
4    PROPINSI VARCHAR(100) NULL,
5    PRIMARY KEY(K_PROPINSI, K_NEGARA),
6    FOREIGN KEY (K_NEGARA) REFERENCES NEGARA (K_NEGARA)
7  )
8  IN REGTSPACE
9  INDEX IN IDXTSPACE
10 LONG IN LONGTSPACE;
```

Gambar 5.8 *Source Create Table* PROPINSI

Berikut adalah keterangan dari *query* pembentukan tabel jurusan yang ditunjukkan dalam gambar 5.8 :

Baris 1 : pembukaan perintah membuat tabel dengan nama PROPINSI.

Baris 2 : K_PROPINSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : K_NEGARA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : PROPINSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (100) dan kondisi boleh kosong (*null*).

Baris 5 : perintah untuk menjadikan atribut K_PROPINSI dan K_NEGARA sebagai kunci utama (*primary key*).

Baris 6 : perintah referensi data pada atribut K_NEGARA sebagai *foreign key* yang mengarah pada tabel NEGARA.

Baris 7 : tanda penutup untuk pembentukan atribut tabel.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 9 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 10 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.9 Tabel Kota

Dalam basis data akan dibentuk tabel kota. Tabel ini memiliki empat atribut yaitu k_kota yang menyimpan dat kode kota, k_propinsi yang berisi data kode propinsi, k_negara yang berisi data kode negara dan kota yang berisi data nama kota. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.9.

```
1   CREATE TABLE KOTA (
2     K_KOTA VARCHAR(3) NOT NULL,
3     K_NEGARA VARCHAR(3) NOT NULL,
4     K_PROPINSI VARCHAR(3) NOT NULL,
5     KOTA VARCHAR(100) NULL,
6     PRIMARY KEY(K_KOTA, K_NEGARA, K_PROPINSI),
7     FOREIGN KEY (K_NEGARA, K_PROPINSI) REFERENCES PROPINSI
    (K_NEGARA, K_PROPINSI)
8   )
9   IN REGTSPACE
10  INDEX IN IDXTSPACE
11  LONG IN LONGTSPACE;
```

Gambar 5.9 *Source Create Table* KOTA

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.9 :

Baris 1 : pembukaan perintah membuat tabel dengan nama KOTA.

Baris 2 : K_KOTA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : K_NEGARA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : K_PROPINSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : KOTA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (100) dan kondisi boleh kosong (*null*).

Baris 6 : perintah untuk menjadikan atribut K_KOTA, K_NEGARA dan K_PROPINSI sebagai kunci utama (*primary key*).

Baris 7 : perintah referensi data pada atribut K_NEGARA dan K_PROPINSI sebagai *foreign key* yang mengarah pada tabel JURUSAN.

Baris 8 : tanda penutup untuk pembentukan atribut tabel.

Baris 9 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 10 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 11 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.10 Tabel Sekolah

Dalam basis data akan dibentuk tabel sekolah. Tabel ini memiliki empat atribut yaitu k_sekolah yang menyimpan data kode sekolah, k_kota yang berisi data kode kota, k_propinsi yang berisi data kode propinsi dan k_negara yang berisi data kode negara. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.10.

```
1   CREATE TABLE SEKOLAH (
2     K_SEKOLAH VARCHAR(9) NOT NULL,
3     K_KOTA VARCHAR(3) NOT NULL,
4     K_PROPINSI VARCHAR(3) NOT NULL,
5     K_NEGARA VARCHAR(3) NOT NULL,
6     SEKOLAH VARCHAR (50),
7     PRIMARY KEY(K_SEKOLAH, K_KOTA, K_PROPINSI, K_NEGARA),
8     FOREIGN KEY (K_NEGARA, K_PROPINSI, K_KOTA) REFERENCES KOTA
    (K_NEGARA, K_PROPINSI, K_KOTA)
9   )
10  IN REGTSPACE
11  INDEX IN IDXTSPACE
12  LONG IN LONGTSPACE;
```

Gambar 5.10 *Source Create Table* SEKOLAH

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.10 :

Baris 1 : pembukaan perintah membuat tabel dengan nama SEKOLAH.

Baris 2 : K_SEKOLAH adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (9) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : K_KOTA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : K_PROPINSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : K_NEGARA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 6 : SEKOLAH adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi boleh kosong (*null*).

Baris 7 : perintah untuk menjadikan atribut K_SEKOLAH, K_PROPINSI, K_KOTA dan K_NEGARA sebagai kunci utama (*primary key*).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 8 : perintah referensi data pada atribut K_NEGARA, K_PROPINSI dan K_KOTA sebagai *foreign key* yang mengarah pada tabel KOTA.

Baris 9 : tanda penutup untuk pembentukan atribut tabel.

Baris 10 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 11 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 12 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.11 Tabel Biodata Mahasiswa

Dalam basis data akan dibentuk tabel mahasiswa_biodata. Tabel ini memiliki dua belas atribut yaitu nim yang menyimpan data kode mahasiswa, k_kota yang berisi data kode kota, k_propinsi yang berisi data kode propinsi, k_negara yang berisi data kode negara, k_sekolah yang berisi data kode sekolah, tmp_lahir yang berisi data tempat lahir, tgl_lahir yang berisi data tanggal lahir, alamat_asal yang berisi data alamat asal, alamat_kos yang berisi data alamat kos, no_hp yang berisi data nomor handphone, no_telp_rmh yang berisi data nomor telepon rumah dan no_telp_kos yang berisi data nomor telepon kos. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.11.

```
1  CREATE TABLE MAHASISWA_BIODATA (
2    NIM VARCHAR(15) NOT NULL,
3    K_NEGARA VARCHAR(3) NOT NULL,
4    K_PROPINSI VARCHAR(3) NOT NULL,
5    K_KOTA VARCHAR(3) NOT NULL,
6    K_SEKOLAH VARCHAR(9) NOT NULL,
7    TMP_LAHIR VARCHAR(50),
8    TGL_LAHIR DATE ,
9    ALAMAT_ASAL VARCHAR(60) ,
10   ALAMAT_KOS VARCHAR(100) ,
11   NO_HP VARCHAR(25),
12   NO_TELP_RMH VARCHAR(25) ,
13   NO_TELP_KOS VARCHAR(25) ,
14   PRIMARY KEY(NIM),
15   FOREIGN KEY (NIM) REFERENCES MAHASISWA (NIM),
16   FOREIGN KEY (K_NEGARA, K_PROPINSI, K_KOTA, K_SEKOLAH)
     REFERENCES KOTA (K_NEGARA, K_PROPINSI, K_KOTA, K_SEKOLAH)
17 )
18 IN REGTSPACE
19 INDEX IN IDXTSPACE
20 LONG IN LONGTSPACE;
```

Gambar 5.11 *Source Create Table* MAHASISWA_BIODATA

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.11 :

Baris 1 : pembukaan perintah membuat tabel dengan nama MAHASISWA_BIODATA.

Baris 2 : NIM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (15) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : K_NEGARA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : K_PROPINSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : K_KOTA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 6 : K_SEKOLAH adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (9) dan kondisi tidak boleh kosong (*not null*).

Baris 7 : TMP_LAHIR adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi tidak boleh kosong (*not null*).

Baris 8 : TGL_LAHIR adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi tidak boleh kosong (*not null*).

Baris 9 : ALAMAT_ASAL adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (60).

Baris 10 : ALAMAT_KOS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (100).

Baris 11 : NO_HP adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (25) .

Baris 12 : NO_TELP_RMH adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (25) .

Baris 13 : NO_TELP_KOS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (25) .

Baris 14 : perintah untuk menjadikan atribut NIM sebagai kunci utama (*primary key*).

Baris 15 : perintah referensi data pada atribut NIM sebagai *foreign key* yang mengarah pada tabel MAHASISWA.

Baris 16 : perintah referensi data pada atribut K_NEGARA, K_PROPINSI, K_KOTA dan K_SEKOLAH sebagai *foreign key* yang mengarah pada tabel JURUSAN.

Baris 17 : tanda penutup untuk pembentukan atribut tabel

Baris 18 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 19 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 20 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.12 Tabel Mahasiswa 1

Dalam basis data akan dibentuk tabel mahasiswa1. Tabel ini memiliki tiga atribut yaitu nim yang menyimpan kode mahasiswa, nama yang berisi data nama mahasiswa dan angkatan yang berisi data angkatan. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.12.

```
1  CREATE TABLE MAHASISWA (
2    NIM VARCHAR(15) NOT NULL,
3    NAMA VARCHAR(100),
4    ANGKATAN SMALLINT,
5    PRIMARY KEY(NIM),
6  )
7  IN REGTSPACE
8  INDEX IN IDXTSPACE
9  LONG IN LONGTSPACE;
```

Gambar 5.12 *Source Create Table* MAHASISWA

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.12 :

Baris 1 : pembukaan perintah membuat tabel dengan nama MAHASISWA.

Baris 2 : NIM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (15) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : NAMA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (100) .

Baris 4 : ANGKATAN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *smallint* .

Baris 5 : perintah untuk menjadikan atribut NIM sebagai kunci utama (*primary key*).

Baris 6 : tanda penutup untuk pembentukan atribut tabel.

Baris 7 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 9 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5.4.2.13 Tabel Pekerjaan

Dalam basis data akan dibentuk tabel pekerjaan. Tabel ini memiliki dua atribut yaitu k_pekerjaan yang menyimpan data kode pekerjaan dan pekerjaan yang berisi data nama pekerjaan. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.13.

```
1  CREATE TABLE PEKERJAAN (
2    K_PEKERJAAN VARCHAR(3) NOT NULL,
3    PEKERJAAN VARCHAR(60) NULL,
4    PRIMARY KEY(K_PEKERJAAN)
5  )
6  IN REGTSPACE
7  INDEX IN IDXTSPACE
8  LONG IN LONGTSPACE;
```

Gambar 5.13 *Source Create Table* PEKERJAAN

Berikut adalah keterangan dari *query* pembentukan tabel fakultas yang ditunjukkan dalam gambar 5.13 :

Baris 1 : pembukaan perintah membuat tabel dengan nama PEKERJAAN.

Baris 2 : K_PEKERJAAN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : PEKERJAAN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi boleh kosong (*null*).

Baris 4 : perintah untuk menjadikan atribut K_PEKERJAAN sebagai kunci utama (*primary key*).

Baris 5 : tanda penutup untuk pembentukan atribut tabel.

Baris 6 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 7 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.14 Tabel Orang Tua Mahasiswa

Dalam basis data akan dibentuk tabel mahasiswa_ortu. Tabel ini memiliki tiga belas atribut yaitu nim yang menyimpan data kode mahasiswa, nama_ayah yang berisi data nama ayah, k_pekerjaan_ayah yang berisi data kode pekerjaan ayah, penghasilan_ayah yang berisi data penghasilan ayah, nama_ibu yang berisi data nama ibu, k_pekerjaan_ibu yang berisi data kode pekerjaan ibu, penghasilan_ibu yang berisi data penghasilan ibu, k_kota yang menyimpan data kode kota, k_propinsi yang berisi data kode propinsi, k_negara yang berisi data kode negara, alamat yang berisi data alamat, no_telpon yang berisi data nomor telepon dan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

no_hp yang berisi data nomor hp. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.14.

```
1  CREATE TABLE MAHASISWA_ORTU (
2    NIM VARCHAR(15) NOT NULL,
3    NAMA_AYAH VARCHAR(45) NOT NULL,
4    K_PEKERJAAN_AYAH VARCHAR(3) NOT NULL,
5    PENGHASILAN_AYAH DOUBLE NOT NULL,
6    NAMA_IBU VARCHAR(45) NOT NULL,
7    K_PEKERJAAN_IBU VARCHAR(3) NOT NULL,
8    PENGHASILAN_IBU DOUBLE NOT NULL,
9    K_PROPINSI VARCHAR(3) NOT NULL,
10   K_NEGARA VARCHAR(3) NOT NULL,
11   K_KOTA VARCHAR(3) NOT NULL,
12   ALAMAT VARCHAR(60) NULL,
13   NO_TELPON VARCHAR(25) NULL,
14   NO_HP VARCHAR(25) NULL,
15   PRIMARY KEY(NIM),
16   FOREIGN KEY (NIM) REFERENCES MAHASISWA (NIM),
17   FOREIGN KEY (K_NEGARA, K_PROPINSI, K_KOTA) REFERENCES KOTA
     (K_NEGARA, K_PROPINSI, K_KOTA),
18   FOREIGN KEY (K_PEKERJAAN_AYAH) REFERENCES PEKERJAAN
     (K_PEKERJAAN),
19   FOREIGN KEY (K_PEKERJAAN_IBU) REFERENCES PEKERJAAN
     (K_PEKERJAAN)
20 )
21 IN REGTSPACE
22 INDEX IN IDXTSPACE
23 LONG IN LONGTSPACE;
```

Gambar 5.14 *Source Create Table* MAHASISWA_ORTU

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.14 :

Baris 1 : pembukaan perintah membuat tabel dengan nama MAHASISWA_BIODATA.

Baris 2 : NIM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (15) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : K_NEGARA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : K_PROPINSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : K_KOTA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 6 : K_SEKOLAH adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (9) dan kondisi tidak boleh kosong (*not null*).

Baris 7 : TMP_LAHIR adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi tidak boleh kosong (*not null*).

Baris 8 : TGL_LAHIR adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi tidak boleh kosong (*not null*).

Baris 9 : ALAMAT_ASAL adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (60).

Baris 10 : ALAMAT_KOS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (100).

Baris 11 : NO_HP adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (25) .

Baris 12 : NO_TELP_RMH adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (25) .

Baris 13 : NO_TELP_KOS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (25) .

Baris 14 : perintah untuk menjadikan atribut NIM sebagai kunci utama (*primary key*).

Baris 15 : perintah referensi data pada atribut NIM sebagai *foreign key* yang mengarah pada tabel MAHASISWA.

Baris 16 : perintah referensi data pada atribut K_NEGARA, K_PROPINSI, K_KOTA dan K_SEKOLAH sebagai *foreign key* yang mengarah pada tabel JURUSAN.

Baris 17 : tanda penutup untuk pembentukan atribut tabel.

Baris 18 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 19 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 20 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.15 Tabel KHS

Dalam basis data akan dibentuk tabel KHS. Tabel ini memiliki enam atribut yaitu tahun_akademik yang menyimpan data tahun akademik, is_ganjil yang berisi data semester ganjil/genap, is_pendek yang berisi data semester pendek atau bukan, nim yang berisi data kode mahasiswa, id_mk yang berisi data mata kuliah dan k_nilai yang berisi data kode nilai. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.15.

```
1  CREATE TABLE KHS (
2    TAHUN_AKADEMIK SMALLINT NOT NULL,
3    IS_GANJIL CHAR(1) NOT NULL,
4    IS_PENDEK CHAR(1) NOT NULL,
5    ID_MK INTEGER NOT NULL,
6    NIM VARCHAR(15) NOT NULL,
7    K_NILAI VARCHAR(2) NOT NULL,
8    PRIMARY KEY(TAHUN_AKADEMIK, IS_GANJIL, IS_PENDEK, ID_MK, NIM),
9    FOREIGN KEY (NIM) REFERENCES MAHASISWA (NIM),
10   FOREIGN KEY (K_NILAI) REFERENCES NILAI (K_NILAI),
11   FOREIGN KEY (ID_MK) REFERENCES MATA_KULIAH (ID_MK)
12 )
13 IN REGTSPACE
14 INDEX IN IDXTSPACE
15 LONG IN LONGTSPACE;
```

Gambar 5.15 *Source Create Table* KHS

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.15 :

Baris 1 : pembukaan perintah membuat tabel dengan nama KHS.

Baris 2 : TAHUN_AKADEMIK adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *smallint* dan kondisi tidak boleh kosong (*not null*).

Baris 3 : IS_GANJUL adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *char* (1) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : IS_PENDEK adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *char* (1) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : ID_MK adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *integer* dan kondisi tidak boleh kosong (*not null*).

Baris 6 : NIM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (15) dan kondisi tidak boleh kosong (*not null*).

Baris 7 : K_NILAI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (2) dan kondisi tidak boleh kosong (*not null*).

Baris 8 : perintah untuk menjadikan atribut TAHUN_AKADEMIK, IS_GANJIL, IS_PENDEK dan NIM sebagai kunci utama (*primary key*).

Baris 9 : perintah referensi data pada atribut NIM sebagai *foreign key* yang mengarah pada tabel MAHASISWA.

Baris 10 : perintah referensi data pada atribut K_NILAI sebagai *foreign key* yang mengarah pada tabel NILAI.

Baris 11 : perintah referensi data pada atribut ID_MK sebagai *foreign key* yang mengarah pada tabel MATA_KULIAH.

Baris 12 : tanda penutup untuk pembentukan atribut tabel.

Baris 13 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 14 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 15 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.16 Tabel Nilai

Dalam basis data akan dibentuk tabel nilai. Tabel ini memiliki lima atribut yaitu k_nilai yang menyimpan data kode nilai, angka yang berisi data angka dari nilai, huruf yang berisi data huruf dari nilai, min_nilai yang berisi data minimum nilai dan max_nilai yang berisi data maximum nilai. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.16.

```
1  CREATE TABLE NILAI (
2    K_NILAI VARCHAR(2) NOT NULL,
3    ANGKA DECIMAL NULL,
4    HURUF VARCHAR NULL,
5    MIN_NILAI DECIMAL NULL,
6    MAX_NILAI DECIMAL NULL,
7    PRIMARY KEY(K_NILAI)
8  )
9  IN REGTSPACE
10 INDEX IN IDXTSPACE
11 LONG IN LONGTSPACE;
```

Gambar 5.16 *Source Create Table* NILAI

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.16 :

Baris 1 : pembukaan perintah membuat tabel dengan nama NILAI.

Baris 2 : ANGKA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *decimal* dan kondisi boleh kosong (*null*).

Baris 3 : HURUF adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *decimal* dan kondisi boleh kosong (*null*).

Baris 4 : MIN_NILAI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *decimal* dan kondisi boleh kosong (*null*).

Baris 5 : MAX_NILAI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *decimal* dan kondisi boleh kosong (*null*).

Baris 6 : PROG_STUDI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (45) dan kondisi boleh kosong (*null*).

Baris 7 : perintah untuk menjadikan atribut K_NILAI sebagai kunci utama (*primary key*).

Baris 8 : tanda penutup untuk pembentukan atribut tabel.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 9 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 10 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 11 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.17 Tabel Mata Kuliah

Dalam basis data akan dibentuk tabel mata_kuliah. Tabel ini memiliki Sembilan atribut yaitu nim yang menyimpan kode mahasiswa, k_fakultas yang berisi data kode fakultas, k_jurusan yang berisi data kode jurusan, k_prog_studi yang berisi kode program studi, k_jenjang yang berisi kode jenjang, k_mata_kuliah yang berisi data kode mata kuliah, thn_kurikulum yang berisi data tahun kurikulum, nama yang berisi data nama mata kuliah dan sks yang berisi data jumlah sks mata kuliah. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.17.

```
1  CREATE TABLE MATA_KULIAH (
2    ID_MK INTEGER NOT NULL,
3    K_FAKULTAS VARCHAR(3) NOT NULL,
4    K_JENJANG VARCHAR(3) NOT NULL,
5    K_JURUSAN VARCHAR(3) NOT NULL,
6    K_PROG_STUDI VARCHAR(3) NOT NULL,
7    K_MATA_KULIAH VARCHAR(10) NOT NULL,
8    THN_KURIKULUM SMALLINT NOT NULL,
9    NAMA VARCHAR(45) NULL,
10   SKS SMALLINT NULL,
11   PRIMARY KEY(ID_MK),
12   FOREIGN KEY (K_FAKULTAS, K_JURUSAN, K_JENJANG, K_PROG_STUDI)
     REFERENCES PROG_STUDI (K_FAKULTAS, K_JURUSAN, K_JENJANG,
     K_PROG_STUDI)
13 )
14 IN REGTSPACE
15 INDEX IN IDXTSPACE
16 LONG IN LONGTSPACE;
```

Gambar 5.17 *Source Create Table* MATA_KULIAH

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.17 :

Baris 1 : pembukaan perintah membuat tabel dengan nama MATA_KULIAH.

Baris 2 : ID_MK adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *integer* dan kondisi tidak boleh kosong (*not null*).

Baris 3 : K_FAKULTAS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 4 : K_JENJANG adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : K_JURUSAN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 6 : K_PROG_STUDI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 7 : K_MATA_KULIAH adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 8 : THN_KURIKULUM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *smallint* dan kondisi tidak boleh kosong (*not null*).

Baris 9 : NAMA adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar (45).*

Baris 10 : SKS adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *smallint.*

Baris 11 : perintah untuk menjadikan atribut ID_MK sebagai kunci utama (*primary key*).

Baris 12 : perintah referensi data pada atribut K_FAKULTAS, K_JENJANG, K_PROG_STUDI dan K_JURUSAN sebagai *foreign key* yang mengarah pada tabel JURUSAN.

Baris 13 : tanda penutup untuk pembentukan atribut tabel.

Baris 14 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 15 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 16 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.18 Tabel Item

Dalam basis data akan dibentuk tabel item. Tabel ini memiliki dua atribut yaitu k_item yang menyimpan kode item dan item yang berisi data nama item. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.18.

```
1  CREATE TABLE ITEM (
2    K_ITEM VARCHAR(3) NOT NULL,
3    ITEM VARCHAR(30),
4    PRIMARY KEY(K_ITEM)
5  )
6  IN REGTSPACE
7  INDEX IN IDXTSPACE
8  LONG IN LONGTSPACE;
```

Gambar 5.18 *Source Create Table Item*

Berikut adalah keterangan dari *query* pembentukan tabel fakultas yang ditunjukkan dalam gambar 5.18 :

Baris 1 : pembukaan perintah membuat tabel dengan nama ITEM.

Baris 2 : K_ITEM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : ITEM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (30) dan kondisi boleh kosong (*null*).

Baris 4 : perintah untuk menjadikan atribut K_ITEM sebagai kunci utama (*primary key*).

Baris 5 : tanda penutup untuk pembentukan atribut tabel.

Baris 6 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 7 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.19 Tabel Transaksi

Dalam basis data akan dibentuk tabel transaksi. Tabel ini memiliki dua atribut yaitu k_transaksi yang menyimpan kode transaksi dan transaksi yang berisi data nama transaksi. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.19.

```
1  CREATE TABLE TRANSAKSI (
2    K_TRANSAKSI VARCHAR(3) NOT NULL,
3    TRANSAKSI VARCHAR(30) ,
4    PRIMARY KEY(K_TRANSAKSI)
5  )
6  IN REGTSPACE
```

Gambar 5.19 *Source Create Table Transaksi*

Berikut adalah keterangan dari *query* pembentukan tabel fakultas yang ditunjukkan dalam gambar 5.19 :

Baris 1 : pembukaan perintah membuat tabel dengan nama TRANSAKSI.

Baris 2 : K_TRANSAKSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : TRANSAKSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (100) dan kondisi boleh kosong (*null*).

Baris 4 : perintah untuk menjadikan atribut K_TRANSAKSI sebagai kunci utama (*primary key*).

Baris 5 : tanda penutup untuk pembentukan atribut tabel.

Baris 6 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 7 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 8 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.4.2.20 Tabel Keuangan

Dalam basis data akan dibentuk tabel fakultas. Tabel ini memiliki delapan atribut yaitu tgl_transaksi yang menyimpan data tanggal transaksi, tahun yang berisi data tahun akademik, is_ganjil yang berisi data semester ganjil/genap, k_item yang berisi data kode item, k_transaksi yang berisi data kode transaksi, nim yang berisi data nim, debet yang berisi data debet dan kredit yang berisi data kredit. *Query* pembentukan tabel ditunjukkan seperti dalam gambar 5.20.

```
1  CREATE TABLE KEUANGAN (
2    TGL_TRANSAKSI TIMESTAMP NOT NULL,
3    TAHUN SMALLINT NOT NULL,
4    IS_GANJIL CHAR(1) NOT NULL,
5    K_ITEM VARCHAR(3) NOT NULL,
6    K_TRANSAKSI VARCHAR(3) NOT NULL,
7    NIM VARCHAR(15) NOT NULL,
8    DEBET DOUBLE ,
9    KREDIT DOUBLE ,
10   PRIMARY KEY(TGL_TRANSAKSI, TAHUN, IS_GANJIL, K_ITEM,
   K_TRANSAKSI, NIM),
11   FOREIGN KEY (NIM) REFERENCES MAHASISWA (NIM),
12   FOREIGN KEY (K_ITEM) REFERENCES ITEM (K_ITEM),
13   FOREIGN KEY (K_TRANSAKSI) REFERENCES TRANSAKSI (K_TRANSAKSI)
14 )
15 IN REGTSPACE
16 INDEX IN IDXTSPACE
17 LONG IN LONGTSPACE;
```

Gambar 5.20 *Source Create Table* Keuangan

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.20 :

Baris 1 : pembukaan perintah membuat tabel dengan nama KEUANGAN.

Baris 2 : TGL_TRANSAKSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *timestamp* dan kondisi tidak boleh kosong (*not null*).

Baris 3 : TAHUN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *smallint* dan kondisi tidak boleh kosong (*not null*).

Baris 4 : IS_GANJIL adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *char* (1) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : K_ITEM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 6 : K_TRANSAKSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 7 : NIM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (15) dan kondisi tidak boleh kosong (*not null*).

Baris 8 : DEBET adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *double.*

Baris 9 : KREDIT adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *double.*

Baris 10 : perintah untuk menjadikan atribut K_TRANSAKSI, TAHUN, IS_GANJIL, K_ITEM dan NIM sebagai kunci utama (*primary key*).

Baris 11 : perintah referensi data pada atribut NIM sebagai *foreign key* yang mengarah pada tabel MAHASISWA.

Baris 12 : perintah referensi data pada atribut K_ITEM sebagai *foreign key* yang mengarah pada tabel ITEM.

Baris 13 : perintah referensi data pada atribut K_TRANSAKSI sebagai *foreign key* yang mengarah pada tabel TRANSAKSI.

Baris 14 : tanda penutup untuk pembentukan atribut tabel.

Baris 15 : pengaturan penyimpanan data pada *tablespace*, IN REGTSPACE data regular pada *tablespace* REGTSPACE.

Baris 16 : pengaturan penyimpanan data pada *tablespace*, INDEX IN IDXTSPACE data *index* pada *tablespace* IDXSPACE.

Baris 17 : pengaturan penyimpanan data pada *tablespace*, LONG IN LONGTSPACE data regular pada *tablespace* LONGTSPACE dan *statement terminator.*

5.5 Implementasi Data

Pada penelitian ini, digunakan data *dummy* yang dibuat dengan mengacu pada jumlah mahasiswa aktif di Universitas Brawijaya. Dalam data yang digunakan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

meruapakan data dengan jumlah mahasiswa jenjang S1. Pembuatan data dalam penelitian ini memanfaatkan Microsoft Excel 2013 sebagai media untuk membuat data pendukung. Beberapa fungsi yang dimanfaatkan pada Microsoft Excel tercantum pada tabel 5.3.

Tabel 5.3 Formula Excel

No	Formula	Penjelasan
1	=RAND()	Mengembalikan nilai acak lebih besar dari atau sama dengan 0 sampai kurang dari 1
2	&	Berfungsi untuk menggabungkan string dan data antar sel
3	=[sel]+1	Berfungsi untuk membuat nomor secara *increment*

Fungsi-fungsi pada tabel 5.3 digunakan untuk membuat data *dummy* yang digunakan pada basis data.

5.5.1 Tabel MASUK

Tabel MASUK merupakan tabel yang berfungsi untuk membentuk data *dummy* pada tabel KEUANGAN. Tabel ini digunakan untuk membentuk data tanggal dan waktu transaksi keuangan pada tabel keuangan. Sedangkan pembentukan data dilakukan dengan memanfaatkan fitur storage procedure yang terdapat pada BD2. Pembentukan tabel MASUK yang ditunjukkan dalam gambar 5.21.

```
1  CREATE TABLE MASUK (
2        KE INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY (START
3  WITH 1 INCREMENT BY 1),
4        WAKTU TIMESTAMP,
5        PRIMARY KEY (KE)
6  )@
```

Gambar 5.21 *Source Create Table* MASUK

Berikut adalah keterangan dari *query* pembentukan tabel fakultas yang ditunjukkan dalam gambar 5.21 :

Baris 1 : pembukaan perintah membuat tabel dengan nama MASUK.

Baris 2 : KE adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *integer* dan kondisi tidak boleh kosong (*not null*) serta akan degenerate secara otomatis mulai 1 dan akan ditambah 1 untuk nilai seterusnya.

Baris 3 : WAKTU adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *timestamp*).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 4 : perintah untuk menjadikan atribut KE sebagai kunci utama (*primary key*).

Baris 5 : tanda penutup untuk pembentukan atribut tabel dan *statement terminator.*

5.5.2 Tabel K_TEMP

Tabel K_TEMP merupakan tabel yang berfungsi untuk membentuk data *dummy* pada tabel KEUANGAN. Tabel ini digunakan sebagai tabel yang digabungkan dengan tabel MASUK untuk membentuk data pada tabel KEUANGAN. Pembentukan tabel K_TEMP yang ditunjukkan dalam gambar 5.22.

```
1  CREATE TABLE K_TEMP (
2        NIM VARCHAR (15) NOT NULL,
3        TAHUN INTEGER NOT NULL,
4        IS_GANJIL VARCHAR(1) NOT NULL,
5        K_TRANSAKSI VARCHAR (3) NOT NULL,
6        K_ITEM VARCHAR (3) NOT NULL,
7        DEBET BIGINT NOT NULL,
8        KREDIT BIGINT NOT NULL,
9        KE INTEGER NOT NULL,
10       PRIMARY KEY (NIM, TAHUN, IS_GANJIL, K_TRANSAKSI, K_ITEM,
   KE),
11       FOREIGN KEY (NIM) REFERENCES MAHASISWA (NIM),
12       FOREIGN KEY (K_TRANSAKSI) REFERENCES TRANSAKSI
   (K_TRANSAKSI),
13       FOREIGN KEY (K_ITEM) REFERENCES ITEM (K_ITEM),
14       FOREIGN KEY (KE) REFERENCES MASUK (KE)
15 )@
```

Gambar 5.22 *Source Create Table* K_TEMP

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.22 :

Baris 1 : pembukaan perintah membuat tabel dengan nama K_TEMP.

Baris 2 : NIM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (15) dan kondisi tidak boleh kosong (*not null*).

Baris 3 : TAHUN adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *smallint* dan kondisi tidak boleh kosong (*not null*).

Baris 4 : IS_GANJIL adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *char* (1) dan kondisi tidak boleh kosong (*not null*).

Baris 6 : K_TRANSAKSI adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 5 : K_ITEM adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *varchar* (3) dan kondisi tidak boleh kosong (*not null*).

Baris 7 : DEBET adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *double.*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 8 : KREDIT adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *double.*

Baris 9 : KE adalah nama atribut yang akan dibentuk pada tabel, dengan tipe data *integer* dan kondisi tidak boleh kosong (*not null*).

Baris 10 : perintah untuk menjadikan atribut K_TRANSAKSI, TAHUN, IS_GANJIL, K_ITEM, KE dan NIM sebagai kunci utama (*primary key*).

Baris 11 : perintah referensi data pada atribut NIM sebagai *foreign key* yang mengarah pada tabel MAHASISWA.

Baris 12 : perintah referensi data pada atribut K_TRANSAKSI sebagai *foreign key* yang mengarah pada tabel TRANSAKSI.

Baris 13 : perintah referensi data pada atribut K_ITEM sebagai *foreign key* yang mengarah pada tabel ITEM.

Baris 14 : perintah referensi data pada atribut KE sebagai *foreign key* yang mengarah pada tabel MASUK.

Baris 15 : tanda penutup untuk pembentukan atribut tabel dan *statement terminator.*

5.5.3 Generate Timestamp

Setelah tabel terbentuk kemudian dilakukan pembuatan data pada tabel masuk. Pembuatan data pada tabel masuk dilakukan secara otomatis oleh sistem. Pada tabel masuk, data dari atribut ke diisi secara *auto increment* saat terdapat data masuk dan pada atribut waktu diisi dengan waktu sistem (*current timestamp*). Proses pembuatan pada tabel masuk dilakukan dengan memanfaatkan *storage procedure*. Pembuatan data pada tabel masuk ditunjukkan seperti dalam gambar 5.23.

```
1  CREATE OR REPLACE PROCEDURE PROC_TS (IN P_START INT, IN P_eND
   INT)
2  SPECIFIC PROC_TS
3  LANGUAGE SQL
4  TS:BEGIN
5        DECLARE V_CURRENT INTEGER;
6        SET V_CURRENT = P_START;
7        WHILE (V_CURRENT <= P_END) DO
8              INSERT INTO MASUK (WAKTU) VALUES (CURRENT TIMESTAMP);
9              SET V_CURRENT = V_CURRENT + 1;
10       END WHILE;
11 END TS@
```

Gambar 5.23 *Source Create Procedure* PROC_TS

Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.23 :

Baris 1 : perintah untuk membentuk atau mengganti prosedur jika telah ada dengan nama yang sama dan mebentu prosedur bila belum ada dengan nama prosedur PROC_TS dengan 2 parameter masukan yakni p_START dan P_END

Baris 2 : menspesifikan prosedur yang dibuat dengan nama PROC_TS.

Baris 3 : menspesifikan Bahasa yang dipakai dalam prosedur.

Baris 4 : penanda *statement query* pada prosedur dimulai.

Baris 6 : deklarasi variable dengan nama V_CURRENT dengan tipe data *integer*.

Baris 5 : perintah untuk mengatur bahwa variable V_CURRENT sama dengan variable parameter masukan P_START.

Baris 7 : inisialisasi proses perulangan dengan DO WHILE dengan kondisi batasan nilai V_CURRENT sama dengan P_END untuk pengehntian perulangan. .

Baris 8 : perintah INSERT yang ditunjukkan ke tabel MASUK pada atribut waktu dengan nilai data berupa *current timestamp* yaitu waktu sistem saat itu.

Baris 9 : *statement* penambahan nilai perulangan pada variable V_CURRENT diamana ketiak berulang V_CURRENT akan ditambah 1.

Baris 10 : *statement* yang menandakan akhir dari perulangan.

Baris 11 : *statement* yang mengakhiri atau ditutupnya perinah pada *storage procedure*.

Dari *storage procedure* yang dibuat di atas, *query* tersebut adalah untuk melakukan proses INSERT ke tabel MASUK. Proses INSERT dapat dilakukan berulang sesuai dengan parameter masukan yang ditetapkan ketika pemanggilan *storage procedure* tersebut. Perintah pemanggilan prosedur ditunjukkan dalam gambar 5.24.

```
1  CALL PROC_TS(1, 108480)@
```

Gambar 5.24 *Source Call Procedure* PROC_TS

5.5.4 Drop Tabel K_TEMP dan Tabel MASUK

Setelah data pada tabel KEUANGAN telah terbentuk, maka langkah selanjutnya adalah menghapus tabel K_TEMP dan tabel MASUK. Penghapusan kedua tabel tersebut karena kedua tabel hanya sebagai sarana untuk mendapatkan data. Penghapusan tabel K_TEMP dan tabel MASUK dapat dilihat dalam gambar 5.25.

```
1  DROP TABLE K_TEMP@
2  DROP TABLE MASUK@
```

Gambar 5.25 *Source Drop Table*

Dari gambar ditunjukkan di atas dignakan untuk menghapus table K_TEMP dan MASUK. Penghapusan dilakukan karena kedua tabel tersebut hanya digunakan

untuk membantu proses *generate* data keuangan, sehingga setelah proses selesai akn dihapus karena tidak digunakan untuk proses selanjutnya.

Baris 1 : perintah untuk menhapus table K_TEMP dan *statement terminator.*

Baris 2 : perintah untuk menhapus table MASUK dan *statement terminator.*

5.6 Implementasi Federasi Basis Data

Pada penelitian ini, dibangun sebuah basis data terdistribusi untuk melakukan penelitian. Basis data yang dibangun terdiri dari lima server yang dibentuk dalam *VMware* sebagai *software virtualization*. Dari lima *site*, terdapat satu *site* yang berperan sebagai *site* utama. Dalam pembangunannya, perlu dikonfigurasi pada setiap *site* agar federasi basis data dapat dilakukan. Pada pembangunan federasi basis data perlu dilakukan konfigurasi basis data, *catalog* basis data hingga pembentukan jaringan beserta penamaan inisialisasi tabel yang akan di *remote* pada setiap *site*nya.

5.6.1 Konfigurasi Basis Data

Pada tahap awal basis data terbentuk, servis federasi basis data masih belum difungsikan. Sehingga perlu mengaktifkan terlebih dahulu fungsi pada DBMS di masing-masing *site*. Perintah aktifasi fungsi federasi basis data seperti yang ditunjukkan dalam gambar 5.26. Perintah aktifasi federasi basis data dijalankan pada terminal db2 server dengan tujuan untuk menyalakan fungsi federasi basis data. Perintah diimpelementasikan pada keseluruhan server basis data yang dibuat yakni sebanyak lima *site* basis data. Dengan mengaktifkan fungsi federasi basis data ini memungkinkan setiap *site* dapat berkomunikasi sebagai federasi basis data yang dikatalogkan.

```
1  DB2 UPDATE DBM CFG USING FEDERATED YES
```

Gambar 5.26 *Source update dbm cfg*

5.6.2 *Catalog Node*

Setelah dilakukan aktifasi fungsi federasi basis data pada setiap *site*, langkah selanjutnya adalah dilakukan proses catalog node. Fungsi dari *catalog node* adalah melakukan referensi *node* atau jaringan agar DBMS dapat terhubung pada basis data lain yang tergabung pada sebuah jaringan di *site* yang berbeda. Perintah *catalog node* seperti yang ditunjukkan dalam gambar 5.27.

```
1  DB2 CATALOG TCPIP NODE SERVER1 REMOTE 172.21.4.122 SERVER 50000
2  DB2 CATALOG TCPIP NODE SERVER2 REMOTE 172.21.4.133 SERVER 50000
3  DB2 CATALOG TCPIP NODE SERVER3 REMOTE 172.21.4.144 SERVER 50000
4  DB2 CATALOG TCPIP NODE SERVER4 REMOTE 172.21.4.155 SERVER 50000
```

Gambar 5.27 *Source catalog node*

Perintah *catalog node* yang berjalan di terminal db2 server pada centos. Perintah tersebut merupakan perintah *catalog* jaringan yang dilakukan pada *site* pusat. Perintah tersebut melakukan penandaan pada jaringan TCP/IP dengan IP

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

tujuan (*site*) yang akan difederasikan dengan membentuk sebagai *node* dan mengakses basis data melalui *port* yang digunakan pada *site* untuk komunikasi basis data. Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.27 :

Baris 1 : perintah membuat *catalog node* atau jaringan melalui terminal db2 server pada centos dengan SERVER1 sebagai nama *node* yang melakukan *remote* pada IP 172.21.4.122 melalui *port* 50000.

Baris 2 : perintah membuat *catalog node* atau jaringan melalui terminal db2 server pada centos dengan SERVER2 sebagai nama *node* yang melakukan *remote* pada IP 172.21.4.133 melalui *port* 50000.

Baris 3 : perintah membuat *catalog node* atau jaringan melalui terminal db2 server pada centos dengan SERVER3 sebagai nama *node* yang melakukan *remote* pada IP 172.21.4.144 melalui *port* 50000.

Baris 4 : perintah membuat *catalog node* atau jaringan melalui terminal db2 server pada centos dengan SERVER4 sebagai nama *node* yang melakukan *remote* pada IP 172.21.4.155 melalui *port* 50000.

5.6.3 *Catalog* Basis Data

Catalog basis data merupakan tahap pemberian tanda terhadap basis data pada *site* lain yang tergabung pada federasi basis data. Proses pendaftaran nama basis data dihubungkan dengan *catalog* yang telah dibuat sebelumnya. Dengan proses katalog basis data ini, *site* utama dapat terhubung dengan basis data lain yang terdapat pada *site* lain yang tergabung dalam federasi basis data. Bentuk perintah yang digunakan berjalan pada terminal db2 linux dapat dilihat seperti dalam gambar 5.28.

```
1  DB2 CATALOG DB MHSBIO AT NODE SERVER1
2  DB2 CATALOG DB MHSORTU AT NODE SERVER2
3  DB2 CATALOG DB KHS AT NODE SERVER3
4  DB2 CATALOG DB UANG AT NODE SERVER4
```

Gambar 5.28 *Source catalog database*

Perintah *catalog* basis data di atas dijalankan pada terminal db2 server *site* pusat pada centos. Perintah tersebut membentuk katalog basis data pada *site* pusat dengan nama masing-masing basis data yang akan dihubungkan dan didaftarkan sesuai dengan *node* server yang ada. Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.28 :

Baris 1 : perintah membentuk *catalog* basis data melalui terminal db2 server pada centos dengan nama db MHSBIO yang dapat diakses melalui node SERVER1, sehingga site pusat dapat mengakses basis data pada *remote site.*

Baris 2 : perintah membentuk *catalog basis* data melalui terminal db2 server pada centos dengan nama db MHSORTU yang dapat diakses

melalui node SERVER1, sehingga site pusat dapat mengakses basis data pada *remote site.*

Baris 3 : perintah membentuk *catalog* basis data melalui terminal db2 server pada centos dengan nama db KHS yang dapat diakses melalui node SERVER1, sehingga site pusat dapat mengakses basis data pada *remote site.*

Baris 4 : perintah membentuk catalog basis data melalui terminal db2 server pada centos dengan nama db MHSUANG yang dapat diakses melalui node SERVER1, sehingga site pusat dapat mengakses basis data pada *remote site.*

5.6.4 Membuat *Wrapper*

Wrapper merupakan sebuah jembatan komunikasi yang digunakan. Fungsi dari *wrapper* sendiri merupakan sebagai media protokol yang menghubungkan *site* utama dengan *site* lain yang tergabung pada federasi basis data. Pada penelitian ini dibuat *wrapper* dengan nama DRDA sehingga jika dipantau dari segi komunikasi jaringan protokol yang digunakan adalah DRDA. Proses pembuatan *wrapper* dapat dilakukan dengan dua acara, yaitu dapat melakukan eksekusi perintah langsung melalui terminal linux atau juga memanfaatkan aplikasi pendukung seperti data studio. Perintah yang digunakan untuk membuat wrapper seperti dalam gambar 5.29.

```
1  DB2 CREATE WRAPPER DRDA
```

Gambar 5.29 *Source create wrapper*

5.6.5 Membuat *Server* Federasi Basis Data

Pada tahap ini, dibuat server federasi basis data dengan bantuan data studio. Pembuatan server basis data berfungsi untuk memberikan tanda pada DBMS site pusat agar dapat terhubung ke server pada site lain melalui *wrapper* yang telah dibuat dengan autentikasi nama pengguna dan sandi basis data yang dituju. Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.30 :

```
1  CREATE SERVER SERVER1 TYPE DB2/UDB VERSION 10.5 WRAPPER DRDA
   AUTHORIZATION "DB2INST1" PASSWORD "RAHASIA" OPTIONS (DBNAME
   'MHSBIO');
2  CREATE SERVER SERVER2 TYPE DB2/UDB VERSION 10.5 WRAPPER DRDA
   AUTHORIZATION "DB2INST1" PASSWORD "RAHASIA" OPTIONS (DBNAME
   'MHSORTU');
3  CREATE SERVER SERVER3 TYPE DB2/UDB VERSION 10.5 WRAPPER DRDA
   AUTHORIZATION "DB2INST1" PASSWORD "RAHASIA" OPTIONS (DBNAME
   'KHS');
4  CREATE SERVER SERVER4 TYPE DB2/UDB VERSION 10.5 WRAPPER DRDA
   AUTHORIZATION "DB2INST1" PASSWORD "RAHASIA" OPTIONS (DBNAME
   'MHSUANG');
```

Gambar 5.30 *Source create server*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Perintah pembuatan server federasi basis data dalam eksekusinya memanfaatkan bantuan data studio sebagai media eksekusi perintah. Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.30 :

Baris 1 : perintah pembuatan server federasi basis data melalui data studio dengan nama sesuai katalog SERVER1 tipe DBMS DB2/UDB versi 10.5, menggunakan *wrapper* DRDA dengan otoritas pengguna db2inst1 dan *password* rahasia pada node SERVER1 dan nama basis data yang dituju adalah database MHSBIO.

Baris 2 : perintah pembuatan server federasi basis data melalui data studio dengan nama sesuai katalog SERVER2 tipe DBMS DB2/UDB versi 10.5, menggunakan *wrapper* DRDA dengan otoritas pengguna db2inst1 dan *password* rahasia pada node SERVER1 dan nama basis data yang dituju adalah database MHSORTU.

Baris 3 : perintah pembuatan server federasi basis data melalui data studio dengan nama sesuai katalog SERVER3 tipe DBMS DB2/UDB versi 10.5, menggunakan *wrapper* DRDA dengan otoritas pengguna db2inst1 dan *password* rahasia pada node SERVER1 dan nama basis data yang dituju adalah database KHS.

Baris 4 : perintah pembuatan server federasi basis data melalui data studio dengan nama sesuai katalog SERVER4 tipe DBMS DB2/UDB versi 10.5, menggunakan *wrapper* DRDA dengan otoritas pengguna db2inst1 dan *password* rahasia pada node SERVER1 dan nama basis data yang dituju adalah database MHSUANG.

5.6.6 Pembuatan *User Mapping*

Pada tahap ini dilakukan pembuatan *user mapping* pada *site* pusat. Pembuatan *user mapping* berguna untuk memetakan *user* pada *site* federasi. Dalam pemetaannya dilakukan untuk *user* pada *site* pusat. Perintah pembuatan *user mapping* ditunjukkan dalam gambar 5.31.

```
1  CREATE USER MAPPING FOR "DB2INST1" SERVER SERVER1 OPTIONS ( ADD
   REMOTE_AUTHID 'DB2INST1', ADD REMOTE_PASSWORD 'RAHASIA' );
2  CREATE USER MAPPING FOR "DB2INST1" SERVER SERVER2 OPTIONS ( ADD
   REMOTE_AUTHID 'DB2INST1', ADD REMOTE_PASSWORD 'RAHASIA' );
3  CREATE USER MAPPING FOR "DB2INST1" SERVER SERVER3 OPTIONS ( ADD
   REMOTE_AUTHID 'DB2INST1', ADD REMOTE_PASSWORD 'RAHASIA' );
4  CREATE USER MAPPING FOR "DB2INST1" SERVER SERVER4 OPTIONS ( ADD
   REMOTE_AUTHID 'DB2INST1', ADD REMOTE_PASSWORD 'RAHASIA' );
```

Gambar 5.31 *Source user mapping*

Perintah *user mapping* digunakan untuk membentuk pemetaan *user* pada *site* utama. Perintah tersebut dibuat untuk memetakan pengguna pada server tujuan dengan *user* lokal beserta *password*nya. Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.31 :

Baris 1 : Perintah pembuatan *user mapping* untuk *user* yang digunakan pada *site* utama yaitu MHS pada SERVER1 dengan tambahan *remote authentication id* merupakan *user* lokal yang digunakan yaitu db2inst1 dan *remote password* adalah 'rahasia'.

Baris 2 : Perintah pembuatan *user mapping* untuk *user* yang digunakan pada *site* utama yaitu MHS pada SERVER2 dengan tambahan *remote authentication id* merupakan *user* lokal yang digunakan yaitu db2inst1 dan *remote password* adalah 'rahasia'.

Baris 3 : Perintah pembuatan *user mapping* untuk *user* yang digunakan pada *site* utama yaitu MHS pada SERVER3 dengan tambahan *remote authentication id* merupakan *user* lokal yang digunakan yaitu db2inst1 dan *remote password* adalah 'rahasia'.

Baris 4 : Perintah pembuatan *user mapping* untuk *user* yang digunakan pada *site* utama yaitu MHS pada SERVER4 dengan tambahan *remote authentication id* merupakan *user* lokal yang digunakan yaitu db2inst1 dan *remote password* adalah 'rahasia'.

5.6.7 Pembuatan *Nick Name* Tabel

Pada tahap ini dilakukan pembuatan *nick name* tabel yang merupakan penginisialisasi tabel pada *site* lokal. Pengguna *nick name* ini berfungsi pada pemanggilan data dari tabel-tabel yang tersimpan pada basis data di *site* lokal. Perintah pembuatan *nick name* ditunjukkan dalam gambar 5.32.

```
1   CREATE NICKNAME MHSBIO.MAHASISWA FOR SERVER1.DB2INST1.MAHASISWA;
2   CREATE NICKNAME MHSBIO.NEGARA FOR SERVER1.DB2INST1.NEGARA;
3   CREATE NICKNAME MHSBIO.PROPINSI FOR SERVER1.DB2INST1.PROPINSI;
4   CREATE NICKNAME MHSBIO.KOTA FOR SERVER1.DB2INST1.KOTA;
5   CREATE NICKNAME MHSBIO.SEKOLAH FOR SERVER1.DB2INST1.SEKOLAH;
6   CREATE NICKNAME MHSBIO.MAHASISWA_BIODATA FOR
    SERVER1.DB2INST1.MAHASISWA_BIODATA;

7   CREATE NICKNAME MHSORTU.NEGARA FOR SERVER2.DB2INST1.NEGARA;
8   CREATE NICKNAME MHSORTU.PROPINSI FOR SERVER2.DB2INST1.PROPINSI;
9   CREATE NICKNAME MHSORTU.KOTA FOR SERVER2.DB2INST1.KOTA;
10  CREATE NICKNAME MHSORTU.PEKERJAAN FOR SERVER2.DB2INST1.PEKERJAAN;
11  CREATE NICKNAME MHSORTU.MAHASISWA FOR SERVER2.DB2INST1.MAHASISWA;
12  CREATE NICKNAME MHSORTU.MAHASISWA_ORTU FOR
    SERVER2.DB2INST1.MAHASISWA_ORTU;

13  CREATE NICKNAME KHS.FAKULTAS FOR SERVER3.DB2INST1.FAKULTAS;
14  CREATE NICKNAME KHS.JURUSAN FOR SERVER3.DB2INST1.JURUSAN;
15  CREATE NICKNAME KHS.PROG_STUDI FOR SERVER3.DB2INST1.PROG_STUDI;
16  CREATE NICKNAME KHS.JENJANG FOR SERVER3.DB2INST1.JENJANG;
17  CREATE NICKNAME KHS.SELEKSI FOR SERVER3.DB2INST1.SELEKSI;
18  CREATE NICKNAME KHS.MAHASISWA FOR SERVER3.DB2INST1.MAHASISWA;
19  CREATE NICKNAME KHS.NILAI FOR SERVER3.DB2INST1.NILAI;
20  CREATE NICKNAME KHS.MATA_KULIAH FOR SERVER3.DB2INST1.MATA_KULIAH;
21  CREATE NICKNAME KHS.KHS FOR SERVER3.DB2INST1.KHS;

22  CREATE NICKNAME MHSUANG.MAHASISWA FOR SERVER4.DB2INST1.MAHASISWA;
23  CREATE NICKNAME MHSUANG.ITEM FOR SERVER4.DB2INST1.ITEM;
24  CREATE NICKNAME MHSUANG.TRANSAKSI FOR SERVER4.DB2INST1.TRANSAKSI;
25  CREATE NICKNAME MHSUANG.KEUANGAN FOR SERVER4.DB2INST1.KEUANGAN;
```

Gambar 5.32 ***Source create nick name***

Perintah pembuatan *nick name* tabel pada *site* utama yang tersimpan pada masing-masing *site* lokal. Pada pembuatannya dibuat nama inisialisasi tabel dengan skema seperti yang digunakan pada *site* utama yang ditujukan untuk tabel yang tersimpan pada *site* lokal dengan server dan skema yang digunakan pada *site* lokal. Berikut adalah keterangan dari *query* pembentukan tabel program studi yang ditunjukkan dalam gambar 5.32 :

Baris 1 : perintah pembuatan *nick name* dengan skema MHSBIO dan nama inisial sesuai tabel yaitu tabel MAHASISWA yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER1.

Baris 2 : perintah pembuatan *nick name* dengan skema MHSBIO dan nama inisial sesuai tabel yaitu tabel NEGARA yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER1.

Baris 3 : perintah pembuatan *nick name* dengan skema MHSBIO dan nama inisial sesuai tabel yaitu tabel PROPINSI yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER1.

Baris 4 : perintah pembuatan *nick name* dengan skema MHSBIO dan nama inisial sesuai tabel yaitu tabel KOTA yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER1.

Baris 5 : perintah pembuatan *nick name* dengan skema MHSBIO dan nama inisial sesuai tabel yaitu tabel SEKOLAH yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER1.

Baris 6 : perintah pembuatan *nick name* dengan skema MHSBIO dan nama inisial sesuai tabel yaitu tabel MAHASISWA_BIODATA yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER1.

Baris 7 : perintah pembuatan *nick name* dengan skema MHSORTU dan nama inisial sesuai tabel yaitu tabel NEGARA yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER2.

Baris 8 : perintah pembuatan *nick name* dengan skema MHSORTU dan nama inisial sesuai tabel yaitu tabel PROPINSI yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER2.

Baris 9 : perintah pembuatan *nick name* dengan skema MHSORTU dan nama inisial sesuai tabel yaitu tabel KOTA yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER2.

Baris 10 : perintah pembuatan *nick name* dengan skema MHSORTU dan nama inisial sesuai tabel yaitu tabel PEKERJAAN yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER2.

Baris 11 : perintah pembuatan *nick name* dengan skema MHSORTU dan nama inisial sesuai tabel yaitu tabel MAHASISWA yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER2.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 12 : perintah pembuatan *nick name* dengan skema MHSORTU dan nama inisial sesuai tabel yaitu tabel MAHASISWA_ORTU yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER2.

Baris 13 : perintah pembuatan *nick name* dengan skema KHS dan nama inisial sesuai tabel yaitu tabel FAKULTAS yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER3.

Baris 14 : perintah pembuatan *nick name* dengan skema KHS dan nama inisial sesuai tabel yaitu tabel JURUSAN yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER3.

Baris 15 : perintah pembuatan *nick name* dengan skema KHS dan nama inisial sesuai tabel yaitu tabel PROG_STUDI yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER3.

Baris 16 : perintah pembuatan *nick name* dengan skema KHS dan nama inisial sesuai tabel yaitu tabel JENJANG yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER3.

Baris 17 : perintah pembuatan *nick name* dengan skema KHS dan nama inisial sesuai tabel yaitu tabel SELEKSI yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER3.

Baris 18 : perintah pembuatan *nick name* dengan skema KHS dan nama inisial sesuai tabel yaitu tabel MAHASISWA yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER3.

Baris 19 : perintah pembuatan *nick name* dengan skema KHS dan nama inisial sesuai tabel yaitu tabel NILAI yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER3.

Baris 20 : perintah pembuatan *nick name* dengan skema KHS dan nama inisial sesuai tabel yaitu tabel MATA_KULIAH yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER3.

Baris 21 : perintah pembuatan *nick name* dengan skema KHS dan nama inisial sesuai tabel yaitu tabel KHS yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER3.

Baris 22 : perintah pembuatan *nick name* dengan skema MHSUANG dan nama inisial sesuai tabel yaitu tabel KHS yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER4.

Baris 23 : perintah pembuatan *nick name* dengan skema MHSUANG dan nama inisial sesuai tabel yaitu tabel KHS yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER4.

Baris 24 : perintah pembuatan *nick name* dengan skema MHSUANG dan nama inisial sesuai tabel yaitu tabel KHS yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER4.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 25 : perintah pembuatan *nick name* dengan skema MHSUANG dan nama inisial sesuai tabel yaitu tabel KHS yang terdapat pada *site* lokal dengan skema db2inst1 pada SERVER4.

5.7 Implementasi *Import* Data

Dalam penelitian ini, dilakukan fragmentasi data yang dilakukan secara vertikal. Seluruh data yang telah dibuat diimport ke dalam basis data yang terletak pada setiap *site*. Dikarenakan fragmentasi dilakukan secara vertikal sehingga data di setiap *site* berbeda satu sama lain. Setiap *site* mangandung seluruh jumlah Mahasiswa di Universitas Brawijaya, tetapi data yang disimpan mengenai hal yang berbeda. Pada tahap *import* data dijalankan sintak *load* pada terminal linux pada masing-masing *site*. Sebelumnya dilakukan *share folder* pada *vmware* agar dapat mengakses folder data yang akan di*import*.

5.7.1 Fragmen 1 (*Site* 1)

Pada fragmen 1 menyimpan data mahasiswa yang meliputi tabel Fakultas, tabel Jurusan, tabel Jenjang, tabel Program Studi, tabel Seleksi dan tabel Mahasiswa. Proses memasukkan data ke tabel pada *database* dilakukan dengan memanfaatkan perintah *load* berfungsi untuk memuat (*loading*) data ke dalam tabel pada terminal linux, dan data yang dimasukkan disimpan dalam format csv. Berikut adalah sintak proses *load* data yang dilakukan pada *site* 1 ditunjukkan dalam gambar 5.33.

```
1   db2 "load from /mnt/hgfs/import/FAKULTAS.csv OF DEL savecount
    30000 insert into db2inst1.fakultas"
2   db2 SET INTEGRITY FOR db2inst1.fakultas ALL IMMEDIATE UNCHECKED;

3   db2 "load from /mnt/hgfs/import/JURUSAN.csv OF DEL savecount
    30000 insert into db2inst1.jurusan"
4   db2 SET INTEGRITY FOR db2inst1.jurusan ALL IMMEDIATE UNCHECKED;

5   db2 "load from /mnt/hgfs/import/JENJANG.csv OF DEL savecount
    30000 insert into db2inst1.jenjang"
6   db2 SET INTEGRITY FOR db2inst1.jenjang ALL IMMEDIATE UNCHECKED;

7   db2 "load from /mnt/hgfs/import/SELEKSI.csv OF DEL savecount
    30000 insert into db2inst1.seleksi"
8   db2 SET INTEGRITY FOR db2inst1.seleksi ALL IMMEDIATE UNCHECKED;

9   db2 "load from /mnt/hgfs/import/PRODI.csv OF DEL savecount 30000
    insert into db2inst1.prog_studi"
10  db2 SET INTEGRITY FOR db2inst1.prog_studi ALL IMMEDIATE
    UNCHECKED;

11  db2 "load from /mnt/hgfs/import/MAHASISWA.csv OF DEL savecount
    30000 insert into db2inst1.mahasiswa"
12  db2 SET INTEGRITY FOR db2inst1.mahasiswa ALL IMMEDIATE UNCHECKED;
```

Gambar 5.33 Perintah *Load* Data *Site* 1

Berikut adalah penjelasan penjelasan perintah-perintah dalam gambar 5.33:

Baris 1 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file

path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* FAKULTAS.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Fakultas.

Baris 2 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Faklutas .

Baris 3 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* JURUSAN.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Jurusan.

Baris 4 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Jurusan .

Baris 5 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* JENJANG.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Jenjang.

Baris 6 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Jenjang.

Baris 7 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* SELEKSI.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Seleksi.

Baris 8 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Seleksi.

Baris 9 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* PRODI.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Prog_Studi.

Baris 10 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Prog_Studi.

Baris 11 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama

file yang di*load* MAHASISWA.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Mahasiswa.

Baris 12 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Mahasiswa.

5.7.2 Fragmen 2 (*Site* 2)

Pada fragmen 2 menyimpan data biodata mahasiswa yang meliputi tabel Negara, tabel Propinsi, tabel Kota, tabel Sekolah, tabel Mahasiwa dan tabel Mahasiswa_biodata. Proses memasukkan data ke tabel pada *database* dilakukan dengan memanfaatkan perintah *load* berfungsi untuk memuat (*loading*) data ke dalam tabel pada terminal linux, dan data yang dimasukkan disimpan dalam format csv. Berikut adalah sintak proses *load* data yang dilakukan pada *site* 2 ditunjukkan dalam gambar 5.34.

```
1   db2 "load from /mnt/hgfs/import/NEGARA.csv OF DEL savecount 30000
    insert into db2inst1.negara"
2   db2 SET INTEGRITY FOR db2inst1.negara ALL IMMEDIATE UNCHECKED;

3   db2 "load from /mnt/hgfs/import/PROPINSI.csv OF DEL savecount
    30000 insert into db2inst1.propinsi"
4   db2 SET INTEGRITY FOR db2inst1.propinsi ALL IMMEDIATE UNCHECKED;

5   db2 "load from /mnt/hgfs/import/KOTA.csv OF DEL savecount 30000
    insert into db2inst1.kota"
6   db2 SET INTEGRITY FOR db2inst1.kota ALL IMMEDIATE UNCHECKED;

7   db2 "load from /mnt/hgfs/import/SEKOLAH.csv OF DEL savecount
    30000 insert into db2inst1.sekolah"
8   db2 SET INTEGRITY FOR db2inst1.sekolah ALL IMMEDIATE UNCHECKED;

9   db2 "load from /mnt/hgfs/import/MAHASISWA1.csv OF DEL savecount
    30000 insert into db2inst1.mahasiswa"
10  db2 SET INTEGRITY FOR db2inst1.mahasiswa ALL IMMEDIATE UNCHECKED;

11  db2 "load from /mnt/hgfs/import/MHS_BIODATA.csv OF DEL savecount
    30000 insert into db2inst1.mahasiswa_biodata"
12  db2 SET INTEGRITY FOR db2inst1.mahasiswa_biodata ALL IMMEDIATE
    UNCHECKED;
```

Gambar 5.34 Perntah *Load* Data *Site* 2

Berikut adalah penjelasan penjelasan perintah-perintah dalam gambar 5.34:

Baris 1 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* NEGARA.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Negara.

Baris 2 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Negara .

Baris 3 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* PROPINSI.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Propinsi.

Baris 4 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Propinsi.

Baris 5 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* KOTA.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Kota.

Baris 6 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Kota.

Baris 7 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* SEKOLAH.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Sekolah.

Baris 8 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Sekolah.

Baris 9 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* MAHASISWA1.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Mahasiswa.

Baris 10 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Mahasiswa.

Baris 11 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* MHS_BIODATA.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Mahasiswa_biodata.

Baris 12 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Mahasiswa_biodata.

5.7.3 Fragmen 3 (*Site* 3)

Pada fragmen 3 menyimpan data orang tua mahasiswa yang meliputi tabel Pekerjaan, Tabel Negara, tabel Propinsi, tabel Kota, tabel Mahasiwa dan tabel Mahasiswa_ortu. Proses memasukkan data ke tabel pada *database* dilakukan dengan memanfaatkan perintah *load* berfungsi untuk memuat (*loading*) data ke dalam tabel pada terminal linux, dan data yang dimasukkan disimpan dalam format csv. Berikut adalah sintak proses *load* data yang dilakukan pada *site* 3 ditunjukkan dalam gambar 5.35.

```
1   db2 "load from /mnt/hgfs/import/NEGARA.csv OF DEL savecount 30000
    insert into db2inst1.negara"
2   db2 SET INTEGRITY FOR db2inst1.negara ALL IMMEDIATE UNCHECKED;

3   db2 "load from /mnt/hgfs/import/PROPINSI.csv OF DEL savecount
    30000 insert into db2inst1.propinsi"
4   db2 SET INTEGRITY FOR db2inst1.propinsi ALL IMMEDIATE UNCHECKED;

5   db2 "load from /mnt/hgfs/import/KOTA.csv OF DEL savecount 30000
    insert into db2inst1.kota"
6   db2 SET INTEGRITY FOR db2inst1.kota ALL IMMEDIATE UNCHECKED;

7   db2 "load from /mnt/hgfs/import/PEKERJAAN.csv OF DEL savecount
    30000 insert into db2inst1.pekerjaan"
8   db2 SET INTEGRITY FOR db2inst1.pekerjaan ALL IMMEDIATE UNCHECKED;

9   db2 "load from /mnt/hgfs/import/MAHASISWA1.csv OF DEL savecount
    30000 insert into db2inst1.mahasiswa"
10  db2 SET INTEGRITY FOR db2inst1.mahasiswa ALL IMMEDIATE UNCHECKED;

11  db2 "load from /mnt/hgfs/import/MHS_ORTU.csv OF DEL savecount
    30000 insert into db2inst1.mahasiswa_ortu"
12  db2 SET INTEGRITY FOR db2inst1.mahasiswa_ortu ALL IMMEDIATE
    UNCHECKED;
```

Gambar 5.35 Perntah *Load* Data *Site* 3

Berikut adalah penjelasan penjelasan perintah-perintah dalam gambar 5.35:

Baris 1 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* NEGARA.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Negara.

Baris 2 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Negara .

Baris 3 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* PROPINSI.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Propinsi.

Baris 4 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Propinsi.

Baris 5 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* KOTA.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Kota.

Baris 6 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Kota.

Baris 7 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* PEKERJAAN.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Pekerjaan.

Baris 8 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Pekerjaan.

Baris 9 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* MAHASISWA1.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Mahasiswa.

Baris 10 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Mahasiswa.

Baris 11 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* MHS_ORTU.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Mahasiswa_ortu.

Baris 12 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Mahasiswa_ortu.

5.7.4 Fragmen 4 (*Site* 4)

Pada fragmen 4 menyimpan data KHS mahasiswa yang meliputi tabel Pekerjaan, Tabel Negara, tabel Propinsi, tabel Kota, tabel Mahasiwa dan tabel Mahasiswa_ortu. Proses memasukkan data ke tabel pada *database* dilakukan

dengan memanfaatkan perintah *load* berfungsi untuk memuat (*loading*) data ke dalam tabel pada terminal linux, dan data yang dimasukkan disimpan dalam format csv. Berikut adalah sintak proses *load* data yang dilakukan pada *site* 3 ditunjukkan dalam gambar 5.36.

```
1   db2 "load from /mnt/hgfs/import/FAKULTAS.csv OF DEL savecount
    30000 insert into db2inst1.fakultas"
2   db2 SET INTEGRITY FOR db2inst1.fakultas ALL IMMEDIATE UNCHECKED;

3   db2 "load from /mnt/hgfs/import/JURUSAN.csv OF DEL savecount
    30000 insert into db2inst1.jurusan"
4   db2 SET INTEGRITY FOR db2inst1.jurusan ALL IMMEDIATE UNCHECKED;

5   db2 "load from /mnt/hgfs/import/JENJANG.csv OF DEL savecount
    30000 insert into db2inst1.jenjang"
6   db2 SET INTEGRITY FOR db2inst1.jenjang ALL IMMEDIATE UNCHECKED;

7   db2 "load from /mnt/hgfs/import/SELEKSI.csv OF DEL savecount
    30000 insert into db2inst1.seleksi"
8   db2 SET INTEGRITY FOR db2inst1.seleksi ALL IMMEDIATE UNCHECKED;

9   db2 "load from /mnt/hgfs/import/PRODI.csv OF DEL savecount 30000
    insert into db2inst1.prog_studi"
10  db2 SET INTEGRITY FOR db2inst1.prog_studi ALL IMMEDIATE
    UNCHECKED;

11  db2 "load from /mnt/hgfs/import/MAHASISWA.csv OF DEL savecount
    30000 insert into db2inst1.mahasiswa"
12  db2 SET INTEGRITY FOR db2inst1.mahasiswa ALL IMMEDIATE UNCHECKED;

13  db2 "load from /mnt/hgfs/import/MATA_KULIAH.csv OF DEL savecount
    30000 insert into db2inst1.mata_kuliah"
14  db2 SET INTEGRITY FOR db2inst1.mata_kuliah ALL IMMEDIATE
    UNCHECKED;

15  db2 "load from /mnt/hgfs/import/NILAI.csv OF DEL savecount 30000
    insert into db2inst1.nilai"
16  db2 SET INTEGRITY FOR db2inst1.nilai ALL IMMEDIATE UNCHECKED;

17  db2 "load from /mnt/hgfs/import/KHS.csv OF DEL savecount 30000
    insert into db2inst1.khs"
18  db2 SET INTEGRITY FOR db2inst1.khs ALL IMMEDIATE UNCHECKED;
```

Gambar 5.36 Perntah *Load* Data *Site* 4

Berikut adalah penjelasan penjelasan perintah-perintah dalam gambar 5.36:

Baris 1 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* FAKULTAS.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Fakultas.

Baris 2 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Faklutas .

Baris 3 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama

file yang di*load* JURUSAN.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Jurusan.

Baris 4 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Jurusan .

Baris 5 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* JENJANG.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Jenjang.

Baris 6 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Jenjang.

Baris 7 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* SELEKSI.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Seleksi.

Baris 8 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Seleksi.

Baris 9 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* PRODI.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Prog_Studi.

Baris 10 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Prog_Studi.

Baris 11 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* MAHASISWA.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Mahasiswa.

Baris 12 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Mahasiswa.

Baris 13 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama

file yang di*load* MATA_KULIAH.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Mata_kuliah.

Baris 14 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Mata_kuliah.

Baris 15 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* NILAI.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Nilai.

Baris 16 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Nilai.

Baris 17 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* KHS.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel KHS.

Baris 18 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel KHS.

5.7.5 Fragmen 5 (*Site* 5)

Pada fragmen 5 menyimpan data keuangan yang meliputi tabel Pekerjaan, Tabel Item, tabel Transaksi, tabel Mahasiswa dan tabel Keuangan. Proses memasukkan data ke tabel pada *database* dilakukan dengan memanfaatkan perintah *load* berfungsi untuk memuat (*loading*) data ke dalam tabel pada terminal linux, dan data yang dimasukkan disimpan dalam format csv. Berikut adalah sintak proses *load* data yang dilakukan pada *site* 5 ditunjukkan dalam gambar 5.37.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
1   db2 "load from /mnt/hgfs/import/ITEM.csv OF DEL savecount 30000
    insert into db2inst1.item"
2   db2 SET INTEGRITY FOR db2inst1.item ALL IMMEDIATE UNCHECKED;

3   db2 "load from /mnt/hgfs/import/TRANSAKSI.csv OF DEL savecount
    30000 insert into db2inst1.transaksi"
4   db2 SET INTEGRITY FOR db2inst1.transaksi ALL IMMEDIATE UNCHECKED;

5   db2 "load from /mnt/hgfs/import/MAHASISWA1.csv OF DEL savecount
    30000 insert into db2inst1.mahasiswa"
6   db2 SET INTEGRITY FOR db2inst1.mahasiswa ALL IMMEDIATE UNCHECKED;

7   db2 "load from /mnt/hgfs/import/KEUANGAN_DBP.csv OF DEL savecount
    30000 insert into db2inst1.keuangan"
8   db2 SET INTEGRITY FOR db2inst1.keuangan ALL IMMEDIATE UNCHECKED;

9   db2 "load from /mnt/hgfs/import/KEUANGAN_SPP.csv OF DEL savecount
    30000 insert into db2inst1.keuangan"
10  db2 SET INTEGRITY FOR db2inst1.keuangan ALL IMMEDIATE UNCHECKED;
```

Gambar 5.37 Perntah *Load* Data *Site* 5

Berikut adalah penjelasan penjelasan perintah-perintah dalam gambar 5.37:

Baris 1 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* ITEM.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Item.

Baris 2 : merupakan perintah SET INTEGRITY pdada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Item.

Baris 3 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* TRANSAKSI.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Transaksi.

Baris 4 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Transaksi.

Baris 5 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* MAHASISWA1.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Mahasiswa.

Baris 6 : merupakan perintah SET INTEGRITY pada tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Mahasiswa.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 7 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* KEUANGAN_DBP.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Keuangan.

Baris 8 : merupakan perintah SET INTEGRITY paa tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Keuangan.

Baris 9 : merupakan perintah memasukkan data dengan perintah *load*, data yang di*load* disimpan dalam bentuk csv dan terdapat lokasi file path yakni *share folder* pada *vmware* /mnt/hgfs/import dan nama file yang di*load* KEUANGAN_SPP.csv dilakukan *savecount* agar *log* tidak penuh pada *record* 30000 dan dilakukan INSERT ke tabel Keuangan.

Baris 10 : merupakan perintah SET INTEGRITY pdaa tabel Fakultas untuk mengecek konsistensi dari PK dan FK yang *direference* oleh tabel Keuangan.

5.8 Implementasi Model Akses *Query*

Pada tahap ini diimplementasi mode akses *query* yang didesain. 3 *query* akses data yang didesain dimana setiap *query* akses data meimiki lima model akses *query*. Ketiga *query* tersebut memiliki model akses *query* yaitu dengan menggunakan *inner join*, *outer join* dan *sub query*. Berikut adalah model akses *query* yang dirancang.

5.8.1 *Query* 1 Model 1

Pada *query* 1 model 1, *query select* melibatkan dua *site* yakni *site* 1 yang menyimpan data mahasiswa dan *site* 4 yang menyimpan data KHS mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa, tabel fakultas, tabel jursan, tabel jenjang dan tabel prog_studi. Sedangkan pada *site* 4 meliputi tabel KHS, nilai dan mata_kuliah. *Query* didesain dengan model simple *query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 1 ditunjukkan dalam gambar 5.38.

```
1   SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
          P.PROG_STUDI, K.TAHUN_AKADEMIK, K.IS_GANJIL AS SEMESTER,
          MK.K_MATA_KULIAH AS KODE_MK, MK.NAMA AS NAMA_MK, MK.SKS,
          N.HURUF AS NILAI
2   FROM MAHASISWA M, FAKULTAS F, JURUSAN JR, PROG_STUDI P, JENJANG
          J, KHS.KHS K, KHS.MATA_KULIAH MK, KHS.NILAI N
3   WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
          JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
           M.K_JENJANG = J.K_JENJANG AND M.NIM = K.NIM AND K.ID_MK =
          MK.ID_MK AND K.K_NILAI = N.K_NILAI;
```

Gambar 5.38 *Query* 1 Model 1

Berikut adalah penjelasan dari dalam gambar 5.38 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, k_mata_kuliah, nama_mk, sks dan nilai.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site.* Relasi yang terlibat antara lain Mahasiswa, Prog_studi, Jenjang, Jurusan, Fakultas dan data pada site remote yang mengakses *nick name* KHS yakni relasi KHS, Nilai dan Mata_kuliah.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.2 *Query* 1 Model 2

Pada *query* 1 model 2, *query select* melibatkan dua *site* yakni *site* 1 yang menyimpan data mahasiswa dan *site* 4 yang menyimpan data KHS mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa. Sedangkan pada *site* 4 meliputi tabel fakultas, tabel jursan, tabel jenjang, tabel prog_studi, KHS, nilai dan mata_kuliah. *Query* didesain dengan model simple *query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 2 ditunjukkan dalam gambar 5.39.

```
1   SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
          P.PROG_STUDI, K.TAHUN_AKADEMIK,
           K.IS_GANJIL AS SEMESTER, MK.K_MATA_KULIAH AS KODE_MK,
          MK.NAMA AS NAMA_MK, MK.SKS, N.HURUF AS NILAI
2   FROM MAHASISWA M, KHS.FAKULTAS F, KHS.JURUSAN JR, KHS.PROG_STUDI
          P, KHS.JENJANG J, KHS.KHS K, KHS.MATA_KULIAH MK, KHS.NILAI N
3   WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
          JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
           M.K_JENJANG = J.K_JENJANG AND M.NIM = K.NIM AND K.ID_MK =
          MK.ID_MK AND K.K_NILAI = N.K_NILAI;
```

Gambar 5.39 *Query* 1 Model 2

Berikut adalah penjelasan dari dalam gambar 5.39 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, k_mata_kuliah, nama_mk, sks dan nilai.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa dan data pada site remote yang mengakses *nick name* KHS yakni relasi Jenjang, Prog_studi, Jurusan, Fakultas, KHS, Nilai dan Mata_kuliah.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.3 *Query* 1 Model 3

Pada *query* 1 model 3, *query select* melibatkan dua *site* yakni *site* 1 yang menyimpan data mahasiswa dan *site* 4 yang menyimpan data KHS mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa, tabel fakultas, tabel jursan dan tabel jenjang. Sedangkan pada *site* 4 meliputi tabel prog_studi, KHS, nilai dan mata_kuliah. *Query* didesain dengan model simple *query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 3 ditunjukkan dalam gambar 5.40.

```
1  SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
       P.PROG_STUDI, K.TAHUN_AKADEMIK,
        K.IS_GANJIL AS SEMESTER, MK.K_MATA_KULIAH AS KODE_MK,
       MK.NAMA AS NAMA_MK, MK.SKS, N.HURUF AS NILAI
2  FROM MAHASISWA M, FAKULTAS F, JURUSAN JR, KHS.PROG_STUDI P,
       JENJANG J, KHS.KHS K, KHS.MATA_KULIAH MK, KHS.NILAI N
3  WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
       JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
        M.K_JENJANG = J.K_JENJANG AND M.NIM = K.NIM AND K.ID_MK =
       MK.ID_MK AND K.K_NILAI = N.K_NILAI;
```

Gambar 5.40 *Query* 1 Model 3

Berikut adalah penjelasan dari dalam gambar 5.40 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, k_mata_kuliah, nama_mk, sks dan nilai.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site.* Relasi yang terlibat antara lain Mahasiswa, Jenjang, Jurusan, Fakultas dan data pada site remote yang mengakses *nick name* KHS yakni relasi KHS, Prog_studi, Nilai dan Mata_kuliah.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.4 *Query* 1 Model 4

Pada *query* 1 model 4, *query select* melibatkan dua *site* yakni *site* 1 yang menyimpan data mahasiswa dan *site* 4 yang menyimpan data KHS mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa dan tabel prog_studi. Sedangkan pada *site* 4 meliputi , tabel fakultas, tabel jursan, tabel jenjang, KHS, nilai dan mata_kuliah. *Query* didesain dengan model simple *query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 4 ditunjukkan dalam gambar5.41.

```
1  SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
       P.PROG_STUDI, K.TAHUN_AKADEMIK,
        K.IS_GANJIL AS SEMESTER, MK.K_MATA_KULIAH AS KODE_MK,
       MK.NAMA AS NAMA_MK, MK.SKS, N.HURUF AS NILAI
2  FROM MAHASISWA M, KHS.FAKULTAS F, KHS.JURUSAN JR, PROG_STUDI P,
       KHS.JENJANG J, KHS.KHS K, KHS.MATA_KULIAH MK, KHS.NILAI N
3  WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
       JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
        M.K_JENJANG = J.K_JENJANG AND M.NIM = K.NIM AND K.ID_MK =
       MK.ID_MK AND K.K_NILAI = N.K_NILAI;
```

Gambar 5.41 *Query* 1 Model 4

Berikut adalah penjelasan dari dalam gambar 5.41 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, k_mata_kuliah, nama_mk, sks dan nilai.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa, Prog_studi dan data pada site remote yang mengakses *nick name* KHS yakni relasi KHS, Jenjang, Jurusan, Fakultas, Nilai dan Mata_kuliah.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.5 *Query* 1 Model 5

Pada *query* 1 model 5, *query select* melibatkan dua *site* yakni *site* 1 yang menyimpan data mahasiswa dan *site* 4 yang menyimpan data KHS mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa, tabel fakultas dan tabel jursan. Sedangkan pada *site* 4 meliputi tabel jenjang, tabel prog_studi, KHS, nilai dan mata_kuliah. *Query* didesain dengan model simple *query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 5 ditunjukkan dalam gambar 5.42.

```
1   SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
          P.PROG_STUDI, K.TAHUN_AKADEMIK, K.IS_GANJIL AS SEMESTER,
          MK.K_MATA_KULIAH AS KODE_MK, MK.NAMA AS NAMA_MK, MK.SKS,
          N.HURUF AS NILAI
2   FROM MAHASISWA M, FAKULTAS F, JURUSAN JR, KHS.PROG_STUDI P,
          KHS.JENJANG J, KHS.KHS K, KHS.MATA_KULIAH MK, KHS.NILAI N
3   WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
          JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
           M.K_JENJANG = J.K_JENJANG AND M.NIM = K.NIM AND K.ID_MK =
          MK.ID_MK AND K.K_NILAI = N.K_NILAI;
```

Gambar 5.42 *Query* 1 Model 5

Berikut adalah penjelasan dari dalam gambar 5.42 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, k_mata_kuliah, nama_mk, sks dan nilai.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa, Jurusan, Fakultas dan data pada site remote yang mengakses *nick name* KHS yakni relasi Prog_studi, Jenjang, KHS, Nilai dan Mata_kuliah.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.6 *Query* 2 Model 1

Pada *query* 2 model 1, *query select* melibatkan dua *site* yakni *site* 1 yang menyimpan data mahasiswa dan *site* 5 yang menyimpan data keuangan mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa, tabel fakultas, tabel jursan, tabel jenjang dan tabel prog_studi. Sedangkan pada *site* 5 meliputi tabel keuangan, item dan transaksi. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 1 ditunjukkan dalam gambar 5.43.

```
1   SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
          P.PROG_STUDI, K.TAHUN AS TAHUN_AKADEMIK,
           K.IS_GANJIL AS SEMESTER, I.ITEM AS TRANSAKSI, T.TRANSAKSI
          AS STATUS, K.TGL_TRANSAKSI AS WAKTU, K.DEBET, K.KREDIT
2   FROM MAHASISWA M, FAKULTAS F, JURUSAN JR, PROG_STUDI P, JENJANG
          J, KEUANGAN.KEUANGAN K, KEUANGAN.ITEM I, KEUANGAN.TRANSAKSI
          T
3   WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
          JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
           M.K_JENJANG = J.K_JENJANG AND M.NIM = K.NIM AND K.K_ITEM =
          I.K_ITEM AND K.K_TRANSAKSI = T.K_TRANSAKSI;
```

Gambar 5.43 *Query* 2 Model 1

Berikut adalah penjelasan dari dalam gambar 5.43 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, item, transaksi, tgl_transaksi, debet dan kredit.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa, Prog_studi, Jenjang, Jurusan, Fakultas dan data pada site remote yang mengakses *nick name* KEUANGAN yakni relasi Keuangan, Item dan Transaksi.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.7 *Query* 2 Model 2

Pada *query* 2 model 2, *query select* melibatkan tiga *site* yakni *site* 1 yang menyimpan data mahasiswa, *site* 4 yang menyimpan data KHS mahasiswa dan *site* 5 yang menyimpan data keuangan mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa. Pada *site* 4 tabel yang diakses adalah tabel fakultas, tabel jursan, tabel jenjang dan tabel prog_studi. Sedangkan pada *site* 5 meliputi tabel keuangan, item dan transaksi. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 2 ditunjukkan dalam gambar 5.44.

```
1   SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
          P.PROG_STUDI, K.TAHUN AS TAHUN_AKADEMIK,
           K.IS_GANJIL AS SEMESTER, I.ITEM AS TRANSAKSI, T.TRANSAKSI
          AS STATUS, K.TGL_TRANSAKSI AS WAKTU, K.DEBET, K.KREDIT
2   FROM MAHASISWA M, KHS.FAKULTAS F, KHS.JURUSAN JR, KHS.PROG_STUDI
          P, KHS.JENJANG J, KEUANGAN.KEUANGAN K, KEUANGAN.ITEM I,
          KEUANGAN.TRANSAKSI T
3   WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
          JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
           M.K_JENJANG = J.K_JENJANG AND M.NIM = K.NIM AND K.K_ITEM =
          I.K_ITEM AND K.K_TRANSAKSI = T.K_TRANSAKSI;
```

Gambar 5.44 *Query* 2 Model 2

Berikut adalah penjelasan dari dalam gambar 5.44 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, item, transaksi, tgl_transaksi, debet dan kredit.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa dan pada site remote yang mengakses *nick name* KHS yakni relasi Prog_studi, Jenjang, Jurusan, Fakultas dan data pada site remote kedua yang mengakses *nick name* KEUANGAN yakni relasi Keuangan, Item dan Transaksi.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.8 *Query* 2 Model 3

Pada *query* 2 model 3, *query select* melibatkan tiga *site* yakni *site* 1 yang menyimpan data mahasiswa, *site* 4 yang menyimpan data KHS mahasiswa dan *site* 5 yang menyimpan data keuangan mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa dan tabel prog_studi. Pada *site* 4 tabel yang diakses adalah tabel fakultas, tabel jursan dan tabel jenjang. Sedangkan pada *site* 5 meliputi tabel keuangan, item dan transaksi. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 3 ditunjukkan dalam gambar 5.45.

```
1  SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
       P.PROG_STUDI, K.TAHUN AS TAHUN_AKADEMIK,
        K.IS_GANJIL AS SEMESTER, I.ITEM AS TRANSAKSI, T.TRANSAKSI
       AS STATUS, K.TGL_TRANSAKSI AS WAKTU, K.DEBET, K.KREDIT
2  FROM MAHASISWA M, KHS.FAKULTAS F, KHS.JURUSAN JR, PROG_STUDI P,
       KHS.JENJANG J, KEUANGAN.KEUANGAN K, KEUANGAN.ITEM I,
       KEUANGAN.TRANSAKSI T
3  WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
       JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
        M.K_JENJANG = J.K_JENJANG AND M.NIM = K.NIM AND K.K_ITEM =
       I.K_ITEM AND K.K_TRANSAKSI = T.K_TRANSAKSI;
```

Gambar 5.45 *Query* 2 Model 3

Berikut adalah penjelasan dari dalam gambar 5.45 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, item, transaksi, tgl_transaksi, debet dan kredit.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa, Prog_studi dan pada site remote yang mengakses *nick name* KHS yakni relasi Jenjang, Jurusan, Fakultas dan data pada site remote kedua yang mengakses *nick name* KEUANGAN yakni relasi Keuangan, Item dan Transaksi.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.9 *Query* 2 Model 4

Pada *query* 2 model 4, *query select* melibatkan tiga *site* yakni *site* 1 yang menyimpan data mahasiswa, *site* 4 yang menyimpan data KHS mahasiswa dan *site* 5 yang menyimpan data keuangan mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa, tabel fakultas, tabel jursan dan tabel jenjang. Pada *site* 4 tabel yang diakses adalah tabel prog_studi. Sedangkan pada *site* 5 meliputi tabel keuangan, item dan transaksi. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 4 ditunjukkan dalam gambar 5.46.

```
1  SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
       P.PROG_STUDI, K.TAHUN AS TAHUN_AKADEMIK,
        K.IS_GANJIL AS SEMESTER, I.ITEM AS TRANSAKSI, T.TRANSAKSI
       AS STATUS, K.TGL_TRANSAKSI AS WAKTU, K.DEBET, K.KREDIT
2  FROM MAHASISWA M, FAKULTAS F, JURUSAN JR, KHS.PROG_STUDI P,
       JENJANG J, KEUANGAN.KEUANGAN K, KEUANGAN.ITEM I,
       KEUANGAN.TRANSAKSI T
3  WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
       JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
        M.K_JENJANG = J.K_JENJANG AND M.NIM = K.NIM AND K.K_ITEM =
       I.K_ITEM AND K.K_TRANSAKSI = T.K_TRANSAKSI;
```

Gambar 5.46 *Query* 2 Model 4

Berikut adalah penjelasan dari dalam gambar 5.46 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, item, transaksi, tgl_transaksi, debet dan kredit.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa, Jenjang, Jurusan, Fakultas, dan pada site remote yang mengakses *nick name* KHS yakni relasi Prog_studi dan data pada site remote kedua yang mengakses *nick name* KEUANGAN yakni relasi Keuangan, Item dan Transaksi.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.10 *Query* 2 Model 5

Pada *query* 2 model 5, *query select* melibatkan dua *site* yakni *site* 1 yang menyimpan data mahasiswa dan *site* 5 yang menyimpan data keuangan mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa, tabel fakultas, tabel jursan, tabel jenjang dan tabel prog_studi. Sedangkan pada *site* 5 meliputi tabel mahasiwa, keuangan, item dan transaksi. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 5 ditunjukkan dalam gambar 5.47.

```
1  SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
       P.PROG_STUDI, K.TAHUN AS TAHUN_AKADEMIK,
        K.IS_GANJIL AS SEMESTER, I.ITEM AS TRANSAKSI, T.TRANSAKSI
       AS STATUS, K.TGL_TRANSAKSI AS WAKTU, K.DEBET, K.KREDIT
2  FROM MAHASISWA M, FAKULTAS F, JURUSAN JR, PROG_STUDI P, JENJANG
       J, KEUANGAN.MAHASISWA MHS, KEUANGAN.KEUANGAN K,
       KEUANGAN.ITEM I, KEUANGAN.TRANSAKSI T
3  WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
       JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
        M.K_JENJANG = J.K_JENJANG AND M.NIM = MHS.NIM AND  MHS.NIM
       = K.NIM AND K.K_ITEM = I.K_ITEM AND K.K_TRANSAKSI =
       T.K_TRANSAKSI;
```

Gambar 5.47 *Query* 2 Model 5

Berikut adalah penjelasan dari dalam gambar 5.47 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, tahun_akademik, is_ganjil, item, transaksi, tgl_transaksi, debet dan kredit.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa, Prog_studi, Jenjang, Jurusan, Fakultas dan data pada site remote yang mengakses *nick name* KEUANGAN yakni relasi Mahasiswa, Keuangan, Item dan Transaksi.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.11 *Query* 3 Model 1

Pada *query* 3 model 1, *query select* melibatkan *tiga* site yakni *site* 1 yang menyimpan data mahasiswa, *site* 2 yang menyimpan data biodata mahasiswa dan *site* 3 yang menyimpan data orang tua mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa, tabel fakultas, tabel jurusan, tabel jenjang dan tabel prog_studi. Pada *site* 2 meliputi tabel mahasiwa, dan mahasiswa_biodata. Sedangkan pada *site* 3 meliputi tabel mahasiswa, mahasiswa_ortu, negara, kota, propinsi dan pekerjaan. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 1 ditunjukkan dalam gambar 5.48.

```
1  SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
       P.PROG_STUDI, MB.NO_HP, MB.ALAMAT_KOS,
        MO.NAMA_AYAH, MO.NAMA_IBU, MO.ALAMAT, N.NEGARA,
       PO.PROPINSI, KO.KOTA, PK.PEKERJAAN AS PEKERJAAN_AYAH,
       MO.PENGHASILAN_AYAH AS GAJI,
        PKI.PEKERJAAN AS PEKERJAAN_IBU, MO.PENGHASILAN_IBU AS GAJI
2  FROM MAHASISWA M, FAKULTAS F, JURUSAN JR, PROG_STUDI P, JENJANG
       J, MHSBIO.MAHASISWA_BIODATA MB, MHSORTU.MAHASISWA_ORTU MO,
        MHSORTU.KOTA KO, MHSORTU.PROPINSI PO, MHSORTU.NEGARA N,
       MHSORTU.PEKERJAAN PK, MHSORTU.PEKERJAAN PKI
3  WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
       JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
        M.K_JENJANG = J.K_JENJANG AND M.NIM = MB.NIM AND M.NIM =
       MO.NIM AND MO.K_KOTA = KO.K_KOTA AND KO.K_PROPINSI =
       PO.K_PROPINSI
        AND PO.K_NEGARA = N.K_NEGARA AND MO.K_PEKERJAAN_AYAH =
       PK.K_PEKERJAAN AND MO.K_PEKERJAAN_IBU = PKI.K_PEKERJAAN;
```

Gambar 5.48 *Query* 3 Model 1

Berikut adalah penjelasan dari dalam gambar 5.48 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_Ayah, nama_ibu, alamt, ngara, propinsi, kota, pekerjaan_ayah, gaji_Ayah, pekerjaan_ibu dan gaji_ibu.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa, Prog_studi, Jenjang, Jurusan, Fakultas dan data pada *site remote* yang mengakses *nick name* MHSBIO yakni relasi Mahasiswa_Biodata dan MHSORTU yakni relasi Mahasiswa_ortu, Pekerjaan, Negara, Propinsi dan Kota.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.12 *Query* 3 Model 2

Pada *query* 3 model 2, *query select* melibatkan *tiga* site yakni *site* 1 yang menyimpan data mahasiswa, *site* 2 yang menyimpan data biodata mahasiswa dan *site* 3 yang menyimpan data orang tua mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa, tabel fakultas, tabel jurusan, tabel jenjang dan tabel prog_studi. Pada *site* 2 meliputi negara, kota, propinsi dan mahasiswa_biodata. Sedangkan pada *site* 3 meliputi tabel mahasiswa_ortu dan pekerjaan. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 2 ditunjukkan dalam gambar 5.49.

```
1   SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
          P.PROG_STUDI, MB.NO_HP, MB.ALAMAT_KOS,
           MO.NAMA_AYAH, MO.NAMA_IBU, MO.ALAMAT, N.NEGARA,
          PO.PROPINSI, KO.KOTA, PK.PEKERJAAN AS PEKERJAAN_AYAH,
          MO.PENGHASILAN_AYAH AS GAJI,
           PKI.PEKERJAAN AS PEKERJAAN_IBU, MO.PENGHASILAN_IBU AS GAJI
2   FROM MAHASISWA M, FAKULTAS F, JURUSAN JR, PROG_STUDI P, JENJANG
          J, MHSBIO.MAHASISWA_BIODATA MB, MHSORTU.MAHASISWA_ORTU MO,
           MHSBIO.KOTA KO, MHSBIO.PROPINSI PO, MHSBIO.NEGARA N,
          MHSORTU.PEKERJAAN PK, MHSORTU.PEKERJAAN PKI
3   WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
          JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
           M.K_JENJANG = J.K_JENJANG AND M.NIM = MB.NIM AND M.NIM =
          MO.NIM AND MO.K_KOTA = KO.K_KOTA AND KO.K_PROPINSI =
          PO.K_PROPINSI
           AND PO.K_NEGARA = N.K_NEGARA AND MO.K_PEKERJAAN_AYAH =
          PK.K_PEKERJAAN AND MO.K_PEKERJAAN_IBU = PKI.K_PEKERJAAN;
```

Gambar 5.49 *Query* 3 Model 2

Berikut adalah penjelasan dari dalam gambar 5.49 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_Ayah, nama_ibu, alamt, ngara, propinsi, kota, pekerjaan_ayah, gaji_Ayah, pekerjaan_ibu dan gaji_ibu.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa, Prog_studi, Jenjang, Jurusan, Fakultas dan data pada *site remote* yang mengakses *nick name* MHSBIO yakni relasi Mahasiswa_Biodata, Negara, Propinsi dan Kota dan MHSORTU yakni relasi Mahasiswa_ortu dan Pekerjaan.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.13 *Query* 3 Model 3

Pada *query* 3 model 3, *query select* melibatkan empat *site* yakni *site* 1 yang menyimpan data mahasiswa, *site* 2 yang menyimpan data biodata mahasiswa, *site* 4 yang menyimpan data KHS mahasiswa dan *site* 3 yang menyimpan data orang tua mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa. Pada *site* 4 tabel yang diakses adalah tabel fakultas, tabel jurusan, tabel jenjang dan tabel prog_studi. Pada *site* 2 meliputi tabel negara, kota, propinsi dan mahasiswa_biodata. Sedangkan pada *site* 3 meliputi tabel mahasiswa_ortu dan pekerjaan. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses

selection ditulis dalam klausa WHERE. Bentuk *query* model 3 ditunjukkan dalam gambar 5.50.

```
1   SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
          P.PROG_STUDI, MB.NO_HP, MB.ALAMAT_KOS,
           MO.NAMA_AYAH, MO.NAMA_IBU, MO.ALAMAT, N.NEGARA,
          PO.PROPINSI, KO.KOTA, PK.PEKERJAAN AS PEKERJAAN_AYAH,
          MO.PENGHASILAN_AYAH AS GAJI,
           PKI.PEKERJAAN AS PEKERJAAN_IBU, MO.PENGHASILAN_IBU AS GAJI
2   FROM MAHASISWA M, KHS.FAKULTAS F, KHS.JURUSAN JR, KHS.PROG_STUDI
          P, KHS.JENJANG J, MHSBIO.MAHASISWA_BIODATA MB,
          MHSORTU.MAHASISWA_ORTU MO,
           MHSBIO.KOTA KO, MHSBIO.PROPINSI PO, MHSBIO.NEGARA N,
3         MHSORTU.PEKERJAAN PK, MHSORTU.PEKERJAAN PKI
    WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
          JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
           M.K_JENJANG = J.K_JENJANG AND M.NIM = MB.NIM AND M.NIM =
          MO.NIM AND MO.K_KOTA = KO.K_KOTA AND KO.K_PROPINSI =
          PO.K_PROPINSI
           AND PO.K_NEGARA = N.K_NEGARA AND MO.K_PEKERJAAN_AYAH =
          PK.K_PEKERJAAN AND MO.K_PEKERJAAN_IBU = PKI.K_PEKERJAAN;
```

Gambar 5.50 *Query* 3 Model 3

Berikut adalah penjelasan dari dalam gambar 5.50 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_Ayah, nama_ibu, alamt, ngara, propinsi, kota, pekerjaan_ayah, gaji_Ayah, pekerjaan_ibu dan gaji_ibu.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa. Pada *site remote* yang mengakses *nick name* KHS yakni relasi Prog_studi, Jenjang, Jurusan, Fakultas dan data pada *site remote* yang mengakses *nick name* MHSBIO yakni relasi Mahasiswa_Biodata, Negara, Propinsi dan Kota dan MHSORTU yakni relasi Mahasiswa_ortu dan Pekerjaan.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.14 *Query* 3 Model 4

Pada *query* 3 model 4, *query select* melibatkan empat *site* yakni *site* 1 yang menyimpan data mahasiswa, *site* 2 yang menyimpan data biodata mahasiswa, *site* 4 yang menyimpan data KHS mahasiswa dan *site* 3 yang menyimpan data orang tua mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa dan tabel prog_studi. Pada *site* 4 tabel yang diakses adalah tabel fakultas, tabel jurusan dan

tabel jenjang. Pada *site* 2 meliputi tabel negara, kota, propinsi dan mahasiswa_biodata. Sedangkan pada *site* 3 meliputi tabel mahasiswa_ortu dan pekerjaan. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 4 ditunjukkan dalam gambar 5.51.

```
1   SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
          P.PROG_STUDI, MB.NO_HP, MB.ALAMAT_KOS,
           MO.NAMA_AYAH, MO.NAMA_IBU, MO.ALAMAT, N.NEGARA,
          PO.PROPINSI, KO.KOTA, PK.PEKERJAAN AS PEKERJAAN_AYAH,
          MO.PENGHASILAN_AYAH AS GAJI,
           PKI.PEKERJAAN AS PEKERJAAN_IBU, MO.PENGHASILAN_IBU AS GAJI
2   FROM MAHASISWA M, KHS.FAKULTAS F, KHS.JURUSAN JR, PROG_STUDI P,
          KHS.JENJANG J, MHSBIO.MAHASISWA_BIODATA MB,
          MHSORTU.MAHASISWA_ORTU MO,
           MHSBIO.KOTA KO, MHSBIO.PROPINSI PO, MHSORTU.NEGARA N,
3         MHSORTU.PEKERJAAN PK, MHSORTU.PEKERJAAN PKI
    WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
          JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
           M.K_JENJANG = J.K_JENJANG AND M.NIM = MB.NIM AND M.NIM =
          MO.NIM AND MO.K_KOTA = KO.K_KOTA AND KO.K_PROPINSI =
          PO.K_PROPINSI
           AND PO.K_NEGARA = N.K_NEGARA AND MO.K_PEKERJAAN_AYAH =
          PK.K_PEKERJAAN AND MO.K_PEKERJAAN_IBU = PKI.K_PEKERJAAN;
```

Gambar 5.51 *Query* 3 Model 4

Berikut adalah penjelasan dari dalam gambar 5.52 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_Ayah, nama_ibu, alamt, ngara, propinsi, kota, pekerjaan_ayah, gaji_Ayah, pekerjaan_ibu dan gaji_ibu.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa dan Prog_studi. Pada *site remote* yang mengakses *nick name* KHS yakni relasi Jenjang, Jurusan, Fakultas dan data pada *site remote* yang mengakses *nick name* MHSBIO yakni relasi Mahasiswa_Biodata, Propinsi dan Kota dan MHSORTU yakni relasi Mahasiswa_ortu, Negara dan Pekerjaan.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

5.8.15 *Query* 3 Model 5

Pada *query* 3 model 5, *query select* melibatkan *tiga* site yakni *site* 1 yang menyimpan data mahasiswa, *site* 2 yang menyimpan data biodata mahasiswa dan

site 3 yang menyimpan data orang tua mahasiswa. Tabel yang diakses pada *site* 1 meliputi tabel mahasiswa, tabel fakultas, tabel jurusan, tabel jenjang dan tabel prog_studi. Pada *site* 2 meliputi tabel kota, propinsi dan mahasiswa_biodata. Sedangkan pada *site* 3 meliputi tabel mahasiswa_ortu, negara dan pekerjaan. *Query* didesain dengan model *simple query* menggunakan *inner join* dalam bentuk *equi join* diamana relasi antar tabel-tabel yang terlibat dalam proses *selection* ditulis dalam klausa WHERE. Bentuk *query* model 5 ditunjukkan dalam gambar 5.53.

```
1  SELECT M.NIM, M.NAMA, J.JENJANG, F.FAKULTAS, JR.JURUSAN,
       P.PROG_STUDI, MB.NO_HP, MB.ALAMAT_KOS,
        MO.NAMA_AYAH, MO.NAMA_IBU, MO.ALAMAT, N.NEGARA,
       PO.PROPINSI, KO.KOTA, PK.PEKERJAAN AS PEKERJAAN_AYAH,
       MO.PENGHASILAN_AYAH AS GAJI,
        PKI.PEKERJAAN AS PEKERJAAN_IBU, MO.PENGHASILAN_IBU AS GAJI
2  FROM MAHASISWA M, FAKULTAS F, JURUSAN JR, PROG_STUDI P, JENJANG
       J, MHSBIO.MAHASISWA_BIODATA MB, MHSORTU.MAHASISWA_ORTU MO,
        MHSBIO.KOTA KO, MHSBIO.PROPINSI PO, MHSORTU.NEGARA N,
       MHSORTU.PEKERJAAN PK, MHSORTU.PEKERJAAN PKI
3  WHERE M.K_PROG_STUDI = P.K_PROG_STUDI AND P.K_JURUSAN =
       JR.K_JURUSAN AND JR.K_FAKULTAS = F.K_FAKULTAS AND
        M.K_JENJANG = J.K_JENJANG AND M.NIM = MB.NIM AND M.NIM =
       MO.NIM AND MO.K_KOTA = KO.K_KOTA AND KO.K_PROPINSI =
       PO.K_PROPINSI
        AND PO.K_NEGARA = N.K_NEGARA AND MO.K_PEKERJAAN_AYAH =
       PK.K_PEKERJAAN AND MO.K_PEKERJAAN_IBU = PKI.K_PEKERJAAN;
```

Gambar 5.52 *Query* 3 Model 5

Berikut adalah penjelasan dari dalam gambar 5.53 di atas :

Baris 1 : merupakan klausa SELECT dimana berisi kolom-kolom yang ingin ditampilkan yang merupakan atribut dari setiap tabel yang ada pada klausa FROM. Atribut yang akan ditampilkan meliputi nim, nama, jenjang, fakultas, jurusan, prog_studi, no_hp, alamat_kos, nama_Ayah, nama_ibu, alamt, ngara, propinsi, kota, pekerjaan_ayah, gaji_Ayah, pekerjaan_ibu dan gaji_ibu.

Baris 2 : merupakan klausa FROM yang berisi relasi yang terlibat pada query SELECT. Relasi yang terlibat tidak hanya pada lokal *site* tapi juga *remote site*. Relasi yang terlibat antara lain Mahasiswa, Prog_studi, Jenjang, Jurusan, Fakultas dan data pada *site remote* yang mengakses *nick name* MHSBIO yakni relasi Mahasiswa_Biodata, Propinsi dan Kota dan MHSORTU yakni relasi Mahasiswa_ortu, Negara dan Pekerjaan.

Baris 3 : merupakan kalusa WHERE yang berisi kondisi dari hubungan data antar relasi satu dengan yang lain yang akan ditampilkan. Yang meliputi kode-kode yang mereferensi antar relasi satu dengan yang lain.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dijelaskan mengenai hasil dari proses implementasi rancangan yang telah dibuat pada bab 4 sesuai dengan permasalahan basis data terdistribusi yang diangkat. Hasil dari proses eksekusi *query* yang akan diuraikan adalah nilai dari parameter heuristik yang digunakan. Setiap *query plan* yang dirancang memiliki nilai parameter heuristik yang berbeda selanjutnya akan dibandingkan nilainya. Perbandingan nilai tersebut digunakan untuk menunjukkan hasil efisiensi dari bagaimana cara proses melakukan pemanggilan data sebagai bentuk otpimasi *query* pada basis data terdistribusi.

6.1 Pengujian *Query* 1

Query 1 merupakan sebuah *query* yang didesain untuk menghasilkan data keterangan hasil studi dari masing-masing mahasiswa pada universitas. Pada *query* 1 ini dibuat 5 model *query* sebagai alternatif *query* untuk pengambilan data. Pada pengujian *query* ini kelima model *query* yang telah terbentuk tidak dijalankan/dieksekusi pada data studio, karena parameter yang digunakan untuk pengujian *query* meggunakan data dari *query* bukan dari data hasil eksekusi *query*. Sehingga biaya *query* tidak dihitung dari waktu eksekusi *query* atau biaya-biaya yang muncul dari hasil eksekusi *query*, tetapi biaya *query* dihitung berdasar parameter yang digunakan pada peneletian ini yaitu pada pada bab 2 sub bab 2.9.2 desain parameter heuristik. Bedasarkan desain parameter tersebut terdapat 3 desain parameter yang digunakan.

6.1.1 *Query Affinity Cost*

Parameter pertama yang ada pada desain parameter heuristik yaitu QAC yang menunjukkan derajat dari keberagaman jumlah dari situs yang terlibat dalam *query plan* yang dibuat, *Query plan* yang melibatkan jumlah situs yang paling sedikit dianggap lebih baik dari alternatif lain, perhitungan nilai QAC menggunakan persamaan 2.13 dan perhitungan lengkap terdapat pada lampiran A.1. Pada *query* 1 terdapat 5 model *query* yang akan diuji untuk mendapatkan hasil dari data hasil studi mahasiswa. Nilai dari QAC setiap model *query* 1 terdapat pada tabel 6.1.

Tabel 6.1 Tabel Nilai *QAC Query* 1

Query Plan	*Query Affinity Cost*
1	0.468
2	0.218
3	0.500
4	0.375
5	0.468

Nilai dari parameter QAC berada antara 0 sampai 1, hal tersebut ditentukan agar membuat perhitungan lebih mudah dan juga mempermudah proses membandingkan antar nilai QAC *query plan* satu dengan *query plan* lain. Pada tabel 6.1 menampilkan nilai dari parameter QAC yang telah dihitung berdasarkan aspek-aspek yang ada pada 5 model *query*. Terdapat dua aspek penting yaitu isi dari *query plan* dan panjang dari *query plan*. Panjang dari *query plan* dihitung sebagai jumlah relasi yang ada pada klausa FROM dari *query* pengguna dan yang disebut isi adalah situs dari relasi tersebut. Perhitungan terdapat pada lampiran.

Untuk *query* 1 model 1 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai dan tabel KHS. Nilai QAC dari model *query* (*QP*) 1 adalah 0.468, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 1 adalah 0.468 dari 1.

Untuk *query* 1 model 2 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Jurusan, tabel Prodi, tabel Fakultas,tabel Jenjang dan tabel KHS. Nilai QAC dari model *query* (*QP*) 2 adalah 0.218, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 2 adalah 0.218 dari 1.

Untuk *query* 1 model 3 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Fakultas, tabel Jenjang dan tabel Jurusan dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Prodi dan tabel KHS. Nilai QAC dari model *query* (*QP*) 3 adalah 0.5, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 3 adalah 0.5 dari 1.

Untuk *query* 1 model 4 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan tabel Prodi dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Jurusan, tabel Fakultas,tabel Jenjang dan tabel KHS. Nilai QAC dari model *query* (QP) 4 adalah 0.375, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 4 adalah 0.375 dari 1.

Untuk *query* 1 model 5 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan dan tabel Fakultas dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Prodi, tabel Jenjang dan tabel KHS. Nilai QAC dari model *query* (*QP*) 5 adalah 0.468, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 5 adalah 0.468 dari 1.

Dari hasil perhitungan nilai setiap *QP* dan membandingkan nilai *QP* 1 dengan yang lain maka nilai dari parameter QAC yang terbaik adalah *QP* 2 dengan nilai 0.218 dari 1. Nilai terbaik adalah nilai yang paling kecil diantara keseluruhan alternatif yang ada.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6.1.2 *Query Localization Cost*

Parameter kedua yang ada pada desain parameter heuristik yaitu QLC yang menunjukkan derajat dua bentuk/model tujuan, pertama jumlah derajat dari komunikasi antara dua situs yang berbeda pada *query plan* dan kedua, hal tersebut memainkan peran penting sekali untuk menentukan situs pengontrol untuk menjawab *query* dari masing-masing *QP*, perhitungan nilai QAC menggunakan persamaan 2.14 dan perhitungan lengkap terdapat pada lampiran B.1. Pada *query* 1 terdapat 5 model *query* yang akan diuji untuk mendapatkan hasil dari data hasil studi mahasiswa. Nilai dari *QPcost* minimal (biaya komunikasi/penempatan *query plan* paling minimum) setiap model *query* 1 terdapat pada tabel 6.2.

Tabel 6.2 Tabel Nilai *QLC Query* 1

Query Plan	*Query Localization Cost*
1	0.00044
2	0.00044
3	0.00044
4	0.00044
5	0.00044

Nilai dari parameter QLC berada antara 0 sampai 1, hal tersebut ditentukan agar membuat perhitungan lebih mudah dan juga mempermudah proses membandingkan antar nilai QLC *query plan* satu dengan *query plan* lain. Pada tabel 6.2 menampilkan nilai dari parameter QLC yang telah dihitung berdasarkan aspek-aspek yang ada pada 5 model *query*.

Untuk *query* 1 model 1 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai dan tabel KHS. Nilai QLC dari model *query* (*QP*) 1 adalah 0, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 1 adalah 0 dari 1.

Untuk *query* 1 model 2 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Jurusan, tabel Prodi, tabel Fakultas,tabel Jenjang dan tabel KHS. Nilai QLC dari model *query* (*QP*) 2 adalah 0.00044, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 2 adalah 0.00044 dari 1.

Untuk *query* 1 model 3 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Fakultas, tabel Jenjang dan tabel Jurusan dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Prodi dan tabel KHS. Nilai QLC dari model

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

query (*QP*) 3 adalah 0.00044, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 3 adalah 0.00044 dari 1.

Untuk *query* 1 model 4 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan tabel Prodi dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Jurusan, tabel Fakultas,tabel Jenjang dan tabel KHS. Nilai QLC dari model *query* (*QP*) 4 adalah 0.00044, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 4 adalah 0.00044 dari 1.

Untuk *query* 1 model 5 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan dan tabel Fakultas dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Prodi, tabel Jenjang dan tabel KHS. Nilai QLC dari model *query* (*QP*) 5 adalah 0.00044, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 5 adalah 0.00044 dari 1.

Dari hasil perhitungan nilai setiap *QP* dan membandingkan nilai *QP* 1 dengan yang lain maka nilai dari parameter QLC yang terbaik adalah keseluruhan *QP* karena keseluruhan bernilai 0.00044 dari 1. Nilai terbaik adalah nilai yang paling kecil diantara keseluruhan alternatif yang ada. Hal tersebut dapat terjadi karena QLC mengambil nilai minimal dari perhitungan yaitu 0.00044.

6.1.3 *Localization Processing Cost*

Parameter ketiga yang ada pada desain parameter heuristik yaitu *query* LPC yang menunjukkan pemrosesan lokal yang secara pokok berfokus pada relasi yang disimpan pada situs lokal. Secara umum LPC bergantung pada jumlah dari jumlah akses memori atau memori yang dimabil untuk mentransfer kumpulan tupel dari *second* memori ke *main* memori. Menghitung nilai dari LPC bergantung pada dua sub-perhitungan, pertama LPC pada situs remot yang ditunjukkan pada persamaan 2.16 dan kedua berdasar pada situs kontrol yang ditunjukkan pada persamaan 2.17 dan perhitungan lengkap terdapat pada lampiran C.1. Pada *query* 1 terdapat 5 model *query* yang akan diuji untuk mendapatkan hasil dari data hasil studi mahasiswa. Nilai dari LPC setiap model *query* 1 terdapat pada tabel 6.3.

Tabel 6.3 Tabel Nilai *LPC Query* 1

Query Plan	*Query Localization Cost*
1	0.1202
2	0.1202
3	0.1202
4	0.1202
5	0.1202

Nilai dari parameter LPC berada antara 0 sampai 1, hal tersebut ditentukan agar membuat perhitungan lebih mudah dan juga mempermudah proses membandingkan antar nilai LPC *query plan* satu dengan *query plan* lain. Pada tabel 6.3 menampilkan nilai dari parameter LPC yang telah dihitung berdasarkan aspek-aspek yang ada pada 5 model *query*.

Untuk *query* 1 model 1 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai dan tabel KHS. Nilai LPC dari model *query* (*QP*) 1 adalah 0.1202, dimana biaya pemrosesan lokal pada *QP* 1 adalah 0.1202 dari 1.

Untuk *query* 1 model 2 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Jurusan, tabel Prodi, tabel Fakultas,tabel Jenjang dan tabel KHS. Nilai LPC dari model *query* (*QP*) 2 adalah 0.1202, dimana biaya pemrosesan lokal pada *QP* 2 adalah 0.1202 dari 1.

Untuk *query* 1 model 3 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Fakultas, tabel Jenjang dan tabel Jurusan dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Prodi dan tabel KHS. Nilai LPC dari model *query* (*QP*) 3 adalah 0.1202, dimana biaya pemrosesan lokal pada *QP* 3 adalah 0.1202 dari 1.

Untuk query 1 model 4 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan tabel Prodi dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Jurusan, tabel Fakultas,tabel Jenjang dan tabel KHS. Nilai LPC dari model *query* (*QP*) 4 adalah 0.1202, dimana biaya pemrosesan lokal pada *QP* 4 adalah 0.1202 dari 1.

Untuk *query* 1 model 5 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan dan tabel Fakultas dan situs KHS dimana relasi yang terlibat adalah tabel Mata_kuliah, tabel Nilai, tabel Prodi, tabel Jenjang dan tabel KHS. Nilai LPC dari model *query* (*QP*) 5 adalah 0.1202, dimana biaya pemrosesan lokal pada *QP* 5 adalah 0.1202 dari 1.

Dari hasil perhitungan nilai setiap *QP* dan membandingkan nilai *QP* 1 dengan yang lain maka nilai dari parameter LPC yang terbaik adalah *QP* 1 dengan nilai 0.1202 dari 1. Nilai terbaik adalah nilai yang paling kecil diantara keseluruhan alternatif yang ada.

6.2 Pengujian Query 2

Query 2 merupakan sebuah *query* yang didesain untuk menghasilkan data keuangan dari masing-masing mahasiswa pada universitas. Pada *query* 2 ini kita

membuat 5 model *query* sebagai alternatif *query* untuk pengambilan data. Pada pengujian *query* ini kelima model *query* yang telah terbentuk tidak dijalankan/dieksekusi pada data studio, karena parameter yang digunakan untuk pengujian *query* meggunakan data dari *query* bukan dari data hasil eksekusi *query*. Sehingga biaya *query* tidak dihitung dari waktu eksekusi *query* atau biaya-biaya yang muncul dari hasil eksekusi *query*, tetapi biaya *query* dihitung berdasar parameter yang digunakan pada peneletian ini yaitu pada pada bab 2 sub bab 2.9.2 desain parameter heuristik. Bedasarkan desain parameter tersebut terdapat 3 desain parameter yang digunakan.

6.2.1 *Query Affinity Cost*

Parameter pertama yang ada pada desain parameter heuristik yaitu QAC yang menunjukkan derajat dari keberagaman jumlah dari situs yang terlibat dalam *query plan* yang dibuat, *Query plan* yang melibatkan jumlah situs yang paling sedikit dianggap lebih baik dari alternatif lain, perhitungan nilai QAC menggunakan persamaan 2.13 dan perhitungan lengkap terdapat pada lampiran A.2. Pada *query* 2 terdapat 5 model *query* yang akan diuji untuk mendapatkan hasil dari data hasil studi mahasiswa. Nilai dari QAC setiap model *query* 1 terdapat pada tabel 6.4.

Tabel 6.4 Tabel Nilai *QAC Query* 2

Query Plan	*Query Affinity Cost*
1	0.468
2	0.593
3	0.656
4	0.593
5	0.494

Nilai dari parameter QAC berada antara 0 sampai 1, hal tersebut ditentukan agar membuat perhitungan lebih mudah dan juga mempermudah proses membandingkan antar nilai QAC *query plan* satu dengan *query plan* lain. Pada tabel 6.4 menampilkan nilai dari parameter QAC yang telah dihitung berdasarkan aspek-aspek yang ada pada 5 model *query*. Terdapat dua aspek penting yaitu isi dari *query plan* dan panjang dari *query plan*. Panjang dari *query plan* dihitung sebagai jumlah relasi yang ada pada klausa FROM dari *query* pengguna dan yang disebut isi adalah situs dari relasi tersebut. Perhitungan terdapat pada lampiran.

Untuk *query* 2 model 1 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QAC dari model *query* (*QP*) 1 adalah 0.468, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 1 adalah 0.468 dari 1.

Untuk *query* 2 model 2 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QAC dari model *query* (*QP*) 2 adalah 0.593, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 2 adalah 0.593 dari 1.

Untuk *query* 2 model 3 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan tabel Prodi, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QAC dari model *query* (*QP*) 3 adalah 0.656, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 3 adalah 0.656 dari 1.

Untuk *query* 2 model 4 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Fakultas dan tabel Jenjang, situs KHS dimana relasi yang terlibat adalah tabel Prodi dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QAC dari model *query* (*QP*) 4 adalah 0.593, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 4 adalah 0.593 dari 1.

Untuk *query* 2 model 5 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QAC dari model *query* (*QP*) 5 adalah 0.494, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 5 adalah 0.494 dari 1.

Dari hasil perhitungan nilai setiap QP dan membandingkan nilai *QP* 1 dengan yang lain maka nilai dari parameter QAC yang terbaik adalah *QP* 1 dengan nilai 0.468 dari 1. Nilai terbaik adalah nilai yang paling kecil diantara keseluruhan alternatif yang ada.

6.2.2 *Query Localization Cost*

Parameter kedua yang ada pada desain parameter heuristik yaitu QLC yang menunjukkan derajat dua bentuk/model tujuan, pertama jumlah derajat dari komunikasi antara dua situs yang berbeda pada *query* plan dan kedua, hal tersebut memainkan peran penting sekali untuk menentukan situs pengontrol untuk menjawab *query* dari masing-masing *QP*, perhitungan nilai QAC menggunakan persamaan 2.14 dan perhitungan lengkap terdapat pada lampiran B.2. Pada *query* 2 terdapat 5 model *query* yang akan diuji untuk mendapatkan hasil dari data hasil studi mahasiswa. Nilai dari *QPcost* minimal (biaya komunikasi/penempatan *query plan* paling minimum) setiap model *query* 1 terdapat pada tabel 6.2.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 6.5 Tabel Nilai *QLC Query* 2

Query Plan	*Query Localization Cost*
1	0.00044
2	0.00044
3	0.00044
4	0.00044
5	0.00033

Nilai dari parameter QLC berada antara 0 sampai 1, hal tersebut ditentukan agar membuat perhitungan lebih mudah dan juga mempermudah proses membandingkan antar nilai QLC *query plan* satu dengan *query plan* lain. Pada tabel 6.2 menampilkan nilai dari parameter QLC yang telah dihitung berdasarkan aspek-aspek yang ada pada 5 model *query*.

Untuk *query* 2 model 1 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QLC dari model *query* (*QP*) 1 adalah 0.00044, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 1 adalah 0.00044 dari 1.

Untuk *query* 2 model 2 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QLC dari model *query* (*QP*) 2 adalah 0.00044, dimana biaya komunikasi/penempatan *query plan* paling minimum pada QP 2 adalah 0.00044 dari 1.

Untuk *query* 2 model 3 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan tabel Prodi, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QLC dari model *query* (*QP*) 3 adalah 0.00044, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 3 adalah0.00044dari 1.

Untuk *query* 2 model 4 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Fakultas dan tabel Jenjang, situs KHS dimana relasi yang terlibat adalah tabel Prodi dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QLC dari model *query* (*QP*) 4 adalah 0.00033, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 4 adalah 0.00033 dari 1.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Untuk *query* 2 model 5 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Keuangan, tabel Item dan tabel Transaksi. Nilai QLC dari model *query* (*QP*) 5 adalah 0.00044, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 5 adalah 0.00044 dari 1.

Dari hasil perhitungan nilai setiap *QP* dan membandingkan nilai *QP* 5 dengan yang lain maka nilai dari parameter QLC yang terbaik adalah *QP* 5. Nilai terbaik adalah nilai yang paling kecil diantara keseluruhan alternatif yang ada yaitu 0.00033.

6.2.3 *Localization Processing Cost*

Parameter ketiga yang ada pada desain parameter heuristik yaitu *query* LPC yang menunjukkan pemrosesan lokal yang secara pokok berfokus pada relasi yang disimpan pada situs lokal. Secara umum LPC bergantung pada jumlah dari jumlah akses memori atau memori yang dimabil untuk mentransfer kumpulan tuple dari *second* memori ke *main* memori. Menghitung nilai dari LPC bergantung pada dua sub-perhitungan, pertama LPC pada situs remot yang ditunjukkan pada persamaan 2.16 dan kedua berdasar pada situs kontrol yang ditunjukkan pada persamaan 2.17 dan perhitungan lengkap terdapat pada lampiran C.2. Pada *query* 2 terdapat 5 model *query* yang akan diuji untuk mendapatkan hasil dari data hasil studi mahasiswa. Nilai dari LPC setiap model *query* 1 terdapat pada tabel 6.3.

Tabel 6.6 Tabel Nilai *LPC Query* 2

Query Plan	*Query Localization Cost*
1	0.3332
2	0.3336
3	0.3336
4	0.3336
5	0.4172

Nilai dari parameter LPC berada antara 0 sampai 1, hal tersebut ditentukan agar membuat perhitungan lebih mudah dan juga mempermudah proses membandingkan antar nilai LPC *query plan* satu dengan *query plan* lain. Pada tabel 6.3 menampilkan nilai dari parameter LPC yang telah dihitung berdasarkan aspek-aspek yang ada pada 5 model *query*.

Untuk *query* 2 model 1 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai LPC dari

model *query* (*QP*) 1 adalah 0.3332, dimana biaya pemrosesan lokal pada *QP* 1 adalah 0.3332 dari 1.

Untuk *query* 2 model 2 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai LPC dari model *query* (*QP*) 2 adalah 0.3336 , dimana biaya pemrosesan lokal pada *QP* 2 adalah 0.3336 dari 1.

Untuk *query* 2 model 3 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan tabel Prodi, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai LPC dari model *query* (*QP*) 3 adalah 0.3336, dimana biaya pemrosesan lokal pada *QP* 3 adalah 0.3336 dari 1.

Untuk *query* 2 model 4 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Fakultas dan tabel Jenjang, situs KHS dimana relasi yang terlibat adalah tabel Prodi dan situs Keuangan dimana relasi yang terlibat adalah tabel Keuangan, tabel Item dan tabel Transaksi. Nilai LPC dari model *query* (*QP*) 4 adalah 0.3336, dimana biaya pemrosesan lokal pada *QP* 4 adalah 0.3336 dari 1.

Untuk *query* 2 model 5 yaitu *query* yang melibatkan 2 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang dan situs Keuangan dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Keuangan, tabel Item dan tabel Transaksi. Nilai LPC dari model *query* (*QP*) 5 adalah 0.4172, dimana biaya pemrosesan lokal pada *QP* 5 adalah 0.4172 dari 1.

Dari hasil perhitungan nilai setiap *QP* dan membandingkan nilai QP 1 dengan yang lain maka nilai dari parameter LPC yang terbaik adalah QP no 1 dengan nilai 0.3332 dari 1. Nilai terbaik adalah nilai yang paling kecil diantara keseluruhan alternatif yang ada.

6.3 Pengujian Query 3

Query 3 merupakan sebuah *query* yang didesain untuk menghasilkan data mahasiswa dari masing-masing mahasiswa pada universitas yang meliputi data diri dan data orang tua mahasiswa. Pada *query* 3 ini kita membuat 5 model *query* sebagai alternatif *query* untuk pengambilan data. Pada pengujian *query* ini kelima model *query* yang telah terbentuk tidak dijalankan/dieksekusi pada data studio, karena parameter yang digunakan untuk pengujian *query* meggunakan data dari *query* bukan dari data hasil eksekusi *query*. Sehingga biaya *query* tidak dihitung dari waktu eksekusi *query* atau biaya-biaya yang muncul dari hasil eksekusi *query*, tetapi biaya *query* dihitung berdasar parameter yang digunakan pada peneletian ini yaitu pada pada bab 2 sub bab 2.9.2 desain parameter heuristik. Bedasarkan desain parameter tersebut terdapat 3 desain parameter yang digunakan.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6.3.1 *Query Affinity Cost*

Parameter pertama yang ada pada desain parameter heuristik yaitu QAC yang menunjukkan derajat dari keberagaman jumlah dari situs yang terlibat dalam *query plan* yang dibuat, *Query plan* yang melibatkan jumlah situs yang paling sedikit dianggap lebih baik dari alternatif lain, perhitungan nilai QAC menggunakan persamaan 2.13 dan perhitungan lengkap terdapat pada lampiran A.3. Pada *query* 3 terdapat 5 model *query* yang akan diuji untuk mendapatkan hasil dari data hasil studi mahasiswa. Nilai dari QAC setiap model *query* 1 terdapat pada tabel 6.1.

Tabel 6.7 Tabel Nilai *QAC Query* 3

Query Plan	*Query Affinity Cost*
1	0.594
2	0.653
3	0.708
4	0.736
5	0.653

Nilai dari parameter QAC berada antara 0 sampai 1, hal tersebut ditentukan agar membuat perhitungan lebih mudah dan juga mempermudah proses membandingkan antar nilai QAC *query plan* satu dengan *query plan* lain. Pada tabel 6.1 menampilkan nilai dari parameter QAC yang telah dihitung berdasarkan aspek-aspek yang ada pada 5 model *query*. Terdapat dua aspek penting yaitu isi dari *query plan* dan panjang dari *query plan*. Panjang dari *query plan* dihitung sebagai jumlah relasi yang ada pada klausa FROM dari *query* pengguna dan yang disebut isi adalah situs dari relasi tersebut. Perhitungan terdapat pada lampiran.

Untuk *query* 3 model 1 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu, tabel Kota, tabel Propinsi, tabel Negara dan tabel Pekerjaan. Nilai QAC dari model *query* (*QP*) 1 adalah 0.594, dimana derajat atau biaya dari keberagaman situs yang terlibat pada QP 1 adalah 0.594 dari 1.

Untuk *query* 3 model 2 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota, tabel Propinsi dan tabel Negara dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu dan tabel Pekerjaan. Nilai QAC dari model *query* (*QP*) 2 adalah 0.653, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 2 adalah 0.653 dari 1.

Untuk *query* 3 model 3 yaitu *query* yang melibatkan 4 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, situs KHS dimana

relasi yang terlibat adalah tabel Jurusan, tabel Fakultas, tabel Prodi dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota, tabel Propinsi dan tabel Negara dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu dan tabel Pekerjaan. Nilai QAC dari model *query* (*QP*) 3 adalah 0.708, dimana derajat atau biaya dari keberagaman situs yang terlibat pada *QP* 3 adalah 0.708 dari 1.

Untuk *query* 3 model 4 yaitu *query* yang melibatkan 4 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan tabel Prodi, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota dan tabel Propinsi dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu, tabel Negara dan tabel Pekerjaan. Nilai QAC dari model *query* (*QP*) 4 adalah 0.736, dimana derajat atau biaya dari keberagaman situs yang terlibat pada QP 4 adalah 0.736 dari 1.

Untuk *query* 3 model 5 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota dan tabel Propinsi dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu, tabel Negara dan tabel Pekerjaan. Nilai QAC dari model *query* (*QP*) 5 adalah 0.653, dimana derajat atau biaya dari keberagaman situs yang terlibat pada QP 5 adalah 0.653 dari 1.

Dari hasil perhitungan nilai setiap QP dan membandingkan nilai QP 1 dengan yang lain maka nilai dari parameter QAC yang terbaik adalah *QP* 1 dengan nilai 0.594 dari 1. Nilai terbaik adalah nilai yang paling kecil diantara keseluruhan alternatif yang ada.

6.3.2 *Query Localization Cost*

Parameter kedua yang ada pada desain parameter heuristik yaitu QLC yang menunjukkan derajat dua bentuk/model tujuan, pertama jumlah derajat dari komunikasi antara dua situs yang berbeda pada *query plan* dan kedua, hal tersebut memainkan peran penting sekali untuk menentukan situs pengontrol untuk menjawab *query* dari masing-masing *QP*, perhitungan nilai QAC menggunakan persamaan 2.14 dan perhitungan lengkap terdapat pada lampiran B.3. Pada *query* 3 terdapat 5 model *query* yang akan diuji untuk mendapatkan hasil dari data hasil studi mahasiswa. Nilai dari *QPcost* minimal (biaya komunikasi/penempatan *query plan* paling minimum) setiap model *query* 1 terdapat pada tabel 6.2.

Tabel 6.8 Tabel Nilai *QLC Query* 3

Query Plan	*Query Localization Cost*
1	0.00022
2	0.00022
3	0.00022
4	0.00022
5	0.00022

Nilai dari parameter QLC berada antara 0 sampai 1, hal tersebut ditentukan agar membuat perhitungan lebih mudah dan juga mempermudah proses membandingkan antar nilai QLC *query plan* satu dengan *query plan* lain. Pada tabel 6.2 menampilkan nilai dari parameter QLC yang telah dihitung berdasarkan aspek-aspek yang ada pada 5 model *query*.

Untuk *query* 3 model 1 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu, tabel Kota, tabel Propinsi, tabel Negara dan tabel Pekerjaan. Nilai QLC dari model *query* (*QP*) 1 adalah 0.00022, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 1 adalah 0.00022 dari 1.

Untuk *query* 3 model 2 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota, tabel Propinsi dan tabel Negara dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu dan tabel Pekerjaan. Nilai QLC dari model *query* (*QP*) 2 adalah 0.00022, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 2 adalah 0.00022 dari 1.

Untuk *query* 3 model 3 yaitu *query* yang melibatkan 4 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Fakultas, tabel Prodi dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota, tabel Propinsi dan tabel Negara dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu dan tabel Pekerjaan. Nilai QLC dari model *query* (*QP*) 3 adalah 0.00022, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 3 adalah 0.00022 dari 1.

Untuk *query* 3 model 4 yaitu *query* yang melibatkan 4 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan tabel Prodi, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Fakultas dan tabel

Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota dan tabel Propinsi dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu, tabel Negara dan tabel Pekerjaan. Nilai QLC dari model *query* (*QP*) 4 adalah 0.00022, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 4 adalah 0.00022 dari 1.

Untuk *query* 3 model 5 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota dan tabel Propinsi dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu, tabel Negara dan tabel Pekerjaan. Nilai QLC dari model *query* (*QP*) 5 adalah 0.00022, dimana biaya komunikasi/penempatan *query plan* paling minimum pada *QP* 5 adalah 0.00022 dari 1.

Dari hasil perhitungan nilai tiap *QP* dan membandingkan nilai *QP* 1 dengan yang lain maka nilai dari parameter QLC yang terbaik adalah keseluruhan *QP* karena keseluruhan bernilai 0.00022 dari 1. Nilai terbaik adalah nilai yang paling kecil diantara keseluruhan alternatif yang ada. Hal tersebut dapat terjadi karena QLC mengambil nilai minimal dari perhitungan yaitu 0.00022.

6.3.3 *Localization Processing Cost*

Parameter ketiga yang ada pada desain parameter heuristik yaitu LPC yang menunjukkan pemrosesan lokal yang secara pokok berfokus pada relasi yang disimpan pada situs lokal. Secara umum LPC bergantung pada jumlah dari jumlah akses memori atau memori yang dimabil untuk mentransfer kumpulan tupel dari *second* memori ke *main* memori. Menghitung nilai dari LPC bergantung pada dua sub-perhitungan, pertama LPC pada situs remot yang ditunjukkan pada persamaan 2.16 dan kedua berdasar pada situs kontrol yang ditunjukkan pada persamaan 2.17 dan perhitungan lengkap terdapat pada lampiran C.3. Pada *query* 3 terdapat 5 model *query* yang akan diuji untuk mendapatkan hasil dari data hasil studi mahasiswa. Nilai dari LPC tiap model *query* 1 terdapat pada tabel 6.3.

Tabel 6.9 Tabel Nilai *LPC Query* 3

Query Plan	*Query Localization Cost*
1	0.3322
2	0.3322
3	0.3326
4	0.3326
5	0.3322

Nilai dari parameter LPC berada antara 0 sampai 1, hal tersebut ditentukan agar membuat perhitungan lebih mudah dan juga mempermudah proses membandingkan antar nilai LPC *query plan* satu dengan *query plan* lain. Pada tabel

6.3 menampilkan nilai dari parameter LPC yang telah dihitung berdasarkan aspek-aspek yang ada pada 5 model *query*.

Untuk *query* 3 model 1 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu, tabel Kota, tabel Propinsi, tabel Negara dan tabel Pekerjaan. Nilai LPC dari model *query* (*QP*) 1 adalah 0.3322, dimana biaya pemrosesan lokal pada *QP* 1 adalah 0.3322 dari 1.

Untuk *query* 3 model 2 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota, tabel Propinsi dan tabel Negara dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu dan tabel Pekerjaan. Nilai LPC dari model *query* (*QP*) 2 adalah 0.3322, dimana biaya pemrosesan lokal pada *QP* 2 adalah 0.3322 dari 1.

Untuk *query* 3 model 3 yaitu *query* yang melibatkan 4 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Fakultas, tabel Prodi dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota, tabel Propinsi dan tabel Negara dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu dan tabel Pekerjaan. Nilai LPC dari model *query* (*QP*) 3 adalah 0.3326, dimana biaya pemrosesan lokal pada *QP* 3 adalah 0.3326 dari 1.

Untuk *query* 3 model 4 yaitu *query* yang melibatkan 4 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa dan tabel Prodi, situs KHS dimana relasi yang terlibat adalah tabel Jurusan, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota dan tabel Propinsi dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu, tabel Negara dan tabel Pekerjaan. Nilai LPC dari model *query* (*QP*) 4 0.3326, dimana biaya pemrosesan lokal pada *QP* 4 adalah 0.3326 dari 1.

Untuk *query* 3 model 5 yaitu *query* yang melibatkan 3 situs yaitu situs Mahasiswa dimana relasi yang terlibat adalah tabel Mahasiswa, tabel Jurusan, tabel Prodi, tabel Fakultas dan tabel Jenjang, situs Mhs_Bio dimana relasi yang terlibat adalah tabel Mhs_bio, tabel Kota dan tabel Propinsi dan situs Mhs_ortu dimana relasi yang terlibat adalah tabel Mhs_ortu, tabel Negara dan tabel Pekerjaan. Nilai LPC dari model *query* (*QP*) 5 adalah 0.3322, dimana biaya pemrosesan lokal pada *QP* 5 adalah 0.3322 dari 1.

Dari hasil perhitungan nilai setiap *QP* dan membandingkan nilai *QP* 1 dengan yang lain maka nilai dari parameter LPC yang terbaik adalah *QP* no 1,2 dan 5 dengan nilai 0.3322 dari 1. Nilai terbaik adalah nilai yang paling kecil diantara keseluruhan alternatif yang ada.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6.4 Pengujian Metode *TLBO Query* 1

Pada tahap pengujian metode atau implementasi dari metode TLBO untuk *Query* 1 dimana menghasilkan data hasil studi mahasiswa yang memiliki 5 model alternatif *query*. Pada metode TLBO terdapat dua tahap yakni fase *Teacher* dan fase *Learner*. Sebelumnya kita menentukan parameter umum yang ada yaitu jumlah populasinya (Pn) adalah 5 sesuai dengan *Query Plan* yang telah dibuat. Selanjutnya jumlah generasi (Gn) atau pembaruan nilai dari parameter sebanyak 2 yakni *Update* Populasi 01 dan *Update* Populasi 02. Jumlah desain variable (Dn) disini yaitu parameter yang kita gunakan berjumlah 3 (QAC, QLC dan LPC) dan batas nilainya yaitu nilai minimal 0 dan maksimal 1. Perhitungan secara rinci untuk penerapan metode TLBO query 1 terdapat pada lampiran D.1.

6.4.1 Fase *Teacher*

Pada fase *teacher* ini akan dilakukan pengolahan data parameter yang telah dihitung sebelumnya untuk mengetahui pada fase ini dari data set mana yang aakan berperan sebagai *teacher*. Pada tabel 6.10 berisi data parameter yang telah dihitung sebelumnya dan akan dilakukan beberapa tahap perhitungan pada fase *teacher*.

Tabel 6.10 Tabel Daftar Biaya Setiap *Query Plan*

QP	QAC	QLC	LPC
1	0.468	0.00044	0.1202
2	0.218	0.00044	0.1202
3	0.500	0.00044	0.1202
4	0.375	0.00044	0.1202
5	0.468	0.00044	0.1202

Pada tabel 6.10 berisi nilai dari masing-masing parameter QAC, QLC dan LPC bagi tiap *Query Plan*. Pada awal fase *teacher* dibutuhkan satu parameter yang berfungsi sebagai parameter yang dapat mewakili ketiga parameter yang ada sehingga memudahkan perbandingan QP yang optimal. Sehingga pada penelitian ini dipilih parameter *fitness* sebagai parameter yang mampu mewakili 3 parameter yang ada. Nilai fitness disini didapatkan dari hasil penghitungan nilai rata-rata dari ketiga parameter. Dan juga dibutuhkan nilai mean dari masing-masing parameter sehingga dihitung juga nilai rata-rata dari parameter QAC, QLC dan LPC. Berikut adalah tabel 6.11 yang berisi parameter dengan nilai *fitness* dan *mean*.

Tabel 6.11 Tabel Nilai Fitness dan Nilai Mean Query Plan

QP	QAC	QLC	LPC	FITNESS

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1	0.468	0.00044	0.1202	0.196
2	0.218	0.00044	0.1202	0.112
3	0.500	0.00044	0.1202	0.206
4	0.375	0.00044	0.1202	0.165
5	0.468	0.00044	0.1202	0.196
MEAN	**0.405**	**0.00044**	**0.1202**	

Dari tabel 6.11 dia atas sudah terdapat nilai *fitness* dan juga nilai *mean* yang akan kita gunakan untuk perhitungan selanjutnya. Kita akan memilih *Teacher* untuk digunakan merubah nilai dari *Mean* yang ada agar menjadi lebih baik. *Teacher* kita pilih dari *Query Plan (learner)* yang dianggap sebagai *learner* yang memiliki pengetahuan tinggi. Dengan cara mengidentifikasi keseluruhan *learner*, *learner* terbaik berdasar pada nilai dari *learner* tersebut yang diwakili parameter yang ada, *learner* dengan nilai *fitness* terkecil merupakan *learner* terbaik dan dianggap menjadi *teacher*. Dari tabel 6.11 kita memilih QP yang diasumsikan sebagai *teacher* berdasar nilai fitness yang terkecil sehingga QP yang terpilih adalah QP no 2 sebagai Ti. Nilai *Mean* baru akan dihitung sebagaimana pada persamaan 2.10 *Difference Mean* dari tiap parameter yang ada pada bab 2. Hasil dari perhitungan terdapat pada tabel 6.12.

Tabel 6.12 Tabel *Teachng Factor* atau *Difference Factor*

Difference Factor (Tf)	
QAC	-0.0865
QLC	0
LPC	0

Dengan *Tf* pada tabel 6.12 yang telah dihitung maka akan dilakukan *update Query Plan* yang ada menjeadi generasi baru. Nilai pada setiap parameter masing-masing *Query Plan* dimodifikasi berdasarkan nilai *Tf* yang telah dihitung sehingga menjadi *Update* Populasi 01. Persamaan untuk mengupdate populasi berada pada bab 2 persamaan 2.12. *Update* Populasi terbaru ditunjukkan pada tabel 6.13.

Tabel 6.13 Tabel *Update Population* 01 Berdasar *Tf*

QP	QAC	QLC	LPC
1	0.3815	0.00044	0.1202

2	0.1315	0.00044	0.1202
3	0.4135	0.00044	0.1202
4	0.2885	0.00044	0.1202
5	0.3815	0.00044	0.1202

6.4.2 Fase *Learner*

Fase selanjutnya adalah fase *learner*, dimana *learner (Query plan)* meningkatkan pengetahuan mereka dengan berinteraksi antar sesame *learner*. Setiap *learner* berinteraksi secara random dengan *learner* lainnya untuk memperluas pengetahuan mereka. *Update* Populasi 01 digunakan sebagai data masukan pada fase *learner* ini yang ditampilkan pada tabel 6.14.

Tabel 6.14 Tabel Nilai *Fitness Population* 01

QP	QAC	QLC	LPC	Fitness
1	0.3815	0.00044	0.1202	0.167
2	0.1315	0.00044	0.1202	0.084
3	0.4135	0.00044	0.1202	0.178
4	0.2885	0.00044	0.1202	0.136
5	0.3815	0.00044	0.1202	0.167

Secara random kita memilih *learner* dan membandingkan dengan *learner* lain, pemenang pada proses perbandingan (*QP 1* lebih baik dari *QP* 2 dan sebaliknya) akan mengupdate *learner* yang lebih lemah dan memperbaiki nilainya. Disini *learner* dengan nilai lebih rendah merupakan *QP* yang menang dan dengan *QP* tersebut kita akan meningkatkan nilai dari *QP* yang memiliki nilai tinggi. Meningkatkan disini berarti mengurangi nilai/biaya agar lebih rendah. Proses interaksi dan penurunan nilai digambarkan dalam gambar 2.22. Hasil dari interaksi antar *learner* akan menghasilkan *Update* Populasi 02 yang ditampilkan pada tabel 6.15.

Tabel 6.15 Tabel *Update Population* 02 Berdasar Interaksi Antar *Learner*

QP	QAC	QLC	LPC	Fitness
1	0.2196	0.00044	0.1202	0.1134
2	0.1315	0.00044	0.1202	0.084

3	0.2309	0.00044	0.1202	0.1171
4	0.1868	0.00044	0.1202	0.1024
5	0.2196	0.00044	0.1202	0.1134

Tabel 6.15 merupakan hasil akhir dari iterasi metode TLBO. Selanjutnya kita akan merangking *learner (Query Plan)* dengan berdasar nilai *fitness* yang terbaik. *QP* yang menempati posisi pertama merupakan alternatif *Query Plan* yang terbaik yang memiliki biaya pemrosesan *query* yang rendah berdasar TLBO. Hasil dari perengkingan ditunjukkan pada tabel 6.16.

Tabel 6.16 Tabel Perangkingan *Query Plan*

QP No	RANK	QC Cost
2	1	0.084
4	2	0.1024
1	3	0.1134
5	4	0.1134
3	5	0.1171

Pada tabel 6.16 adalah hasil akhir dari metode TLBO yang menampilkan urutan *query plan* dari yang paling optimal atau memiliki biaya paling rendah sampai biaya yang tertinggi.

6.5 Pengujian Metode *TLBO Query* 2

Pada tahap pengujian metode atau implementasi dari metode TLBO untuk *Query* 2 dimana menghasilkan data keuangan mahasiswa yang memiliki 5 model alternatif *query*. Pada metode TLBO terdapat dua tahap yakni fase *Teacher* dan fase *Learner*. Sebelumnya kita menentukan parameter umum yang ada yaitu jumlah populasinya (*Pn*) adalah 5 sesuai dengan *Query Plan* yang telah dibuat. Selanjutnya jumlah generasi (*Gn*) atau pembaruan nilai dari parameter sebanyak 2 yakni *Update* Populasi 01 dan *Update* Populasi 02. Jumlah desain variable (Dn) disini yaitu parameter yang kita gunakan berjumlah 3 (QAC, QLC dan LPC) dan batas nilainya yaitu nilai minimal 0 dan maksimal 1. Perhitungan secara rinci untuk penerapan metode TLBO query 1 terdapat pada lampiran D.2.

6.5.1 Fase *Teacher*

Pada fase *teacher* ini akan dilakukan pengolahan data parameter yang telah dihitung sebelumnya untuk mengetahui pada fase ini dari data set mana yang akan berperan sebagai *teacher*. Pada tabel 6.17 berisi data parameter yang telah dihitung sebelumnya dan akan dilakukan beberapa tahap perhitungan pada fase *teacher*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 6.17 Tabel Daftar Biaya Setiap *Query Plan*

QP	QAC	QLC	LPC
1	0.468	0.00044	0.3332
2	0.593	0.00044	0.3336
3	0.656	0.00044	0.3336
4	0.593	0.00044	0.3336
5	0.494	0.00033	0.4172

Pada tabel 6.17 berisi nilai dari masing-masing parameter QAC, QLC dan *LPC* bagi tiap *Query Plan*. Pada awal fase *teacher* dibutuhkan satu parameter yang berfungsi sebagai parameter yang dapat mewakili ketiga parameter yang ada sehingga memudahkan perbandingan QP yang optimal. Sehingga pada penelitian ini dipilih parameter *fitness* sebagai parameter yang mampu mewakili 3 parameter yang ada. Nilai *fitness* disini didapatkan dari hasil penghitungan nilai rata-rata dari ketiga parameter. Dan juga dibutuhkan nilai mean dari masing-masing parameter sehingga dihitung juga nilai rata-rata dari parameter QAC, QLC dan *LPC*. Berikut adalah tabel 6.17 yang berisi parameter dengan nilai *fitness* dan mean.

Tabel 6.18 Tabel Nilai Fitness dan Nilai Mean Query Plan

QP	QAC	QLC	LPC	FITNESS
1	0.468	0.00044	0.3332	0.267
2	0.593	0.00044	0.3336	0.309
3	0.656	0.00044	0.3336	0.330
4	0.593	0.00044	0.3336	0.309
5	0.494	0.00033	0.4172	0.303
MEAN	**0.5608**	**0.00041**	**0.7624**	

Dari tabel 6.18 dia atas sudah terdapat nilai *fitness* dan juga nilai *mean* yang akan kita gunakan untuk perhitungan selanjutnya. Kita akan memilih *Teacher* untuk digunakan merubah nilai dari *Mean* yang ada agar menjadi lebih baik. *Teacher* kita pilih dari *Query Plan (learner)* yang dianggap sebagai *learner* yang memiliki pengetahuan tinggi. Dengan cara mengidentifikasi keseluruhan *learner*, *learner* terbaik berdasar pada nilai dari *learner* tersebut yang diwakili parameter yang ada, *learner* dengan nilai *fitness* terkecil merupakan *learner* terbaik dan dianggap menjadi *teacher*. Dari tabel 6.18 kita memilih *QP* yang diasumsikan sebagai *teacher* berdasar nilai *fitness* yang terkecil sehingga *QP* yang terpilih adalah *QP* 5 sebagai *Ti*. Nilai *Mean* baru akan dihitung sebagaimana persamaan *Difference Mean* dari tiap parameter yang ada pada bab 2 pada persamaan 2.12. Hasil dari perhitungan terdapat pada tabel 6.19.

Tabel 6.19 Tabel *Teachng Factor* atau *Difference Factor*

Difference Factor (Tf)	
QAC	-0.068
QLC	0.00003
LPC	-0.00009

Dengan *Tf* pada tabel 6.19 yang telah dihitung maka akan dilakukan *update Query Plan* yang ada menjeadi generasi baru. Nilai pada setiap parameter masing-masing *Query Plan* dimodifikasi berdasarkan nilai *Tf* yang telah dihitung sehingga menjadi *Update* Populasi 01. Persamaan untuk mengupdate populasi berada pada bab 2 pada persamaan 2.12. *Update* Populasi terbaru ditunjukkan pada tabel 6.20.

Tabel 6.20 Tabel *Update Population* 01 Berdasar *Tf*

QP	QAC	QLC	LPC
1	0.400	0.00044	0.3331
2	0.525	0.00044	0.3335
3	0.588	0.00044	0.3335
4	0.525	0.00044	0.3335
5	0.426	0.00033	0.3331

6.5.2 Fase *Learner*

Fase selanjutnya adalah fase *learner*, dimana *learner (Query plan)* meningkatkan pengetahuan mereka dengan berinteraksi antar sesama *learner*. Setiap *learner* berinteraksi secara random dengan *learner* lainnya untuk memperluas pengetahuan mereka. *Update* Populasi 01 digunakan sebagai data masukan pada fase *learner* ini yang ditampilkan pada tabel 6.21.

Tabel 6.21 Tabel Nilai *Fitness Population* 01

QP	QAC	QLC	LPC	Fitness
1	0.400	0.00044	0.3331	0.244
2	0.525	0.00044	0.3335	0.286
3	0.588	0.00044	0.3335	0.307
4	0.525	0.00044	0.3335	0.286

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5	0.426	0.00033	0.3331	0.253

Secara random kita memilih *learner* dan membandingkan dengan *learner* lain, pemenang pada proses perbandingan (*QP* 1 lebih baik dari *QP* 2 dan sebaliknya) akan mengupdate *learner* yang lebih lemah dan memperbaiki nilainya. Disini learner dengan nilai lebih rendah merupakan *QP* yang menang dan dengan *QP* tersebut kita akan meningkatkan nilai dari *QP* yang memiliki nilai tinggi. Meningkatkan disini berarti mengurangi nilai/biaya agar lebih rendah. Proses interaksi dan penurunan nilai digambarkan dalam gambar 2.22. Hasil dari interaksi antar *learner* akan menghasilkan *Update* Populasi 02 yang ditampilkan pada tabel 6.22.

Tabel 6.22 Tabel *Update Population* 02 Berdasar Interaksi Antar *Learner*

QP	QAC	QLC	LPC	Fitness
1	0.400	0.00041	0.3331	0.244
2	0.453	0.00041	0.3332	0.262
3	0.480	0.00041	0.3332	0.271
4	0.453	0.00041	0.3332	0.262
5	0.411	0.00036	0.3331	0.248

Tabel 6.20 merupakan hasil akhir dari iterasi metode TLBO. Selanjutnya kita akan merangking *learner (Query Plan)* dengan berdasar nilai *fitness* yang terbaik. *QP* yang menempati posisi pertama merupakan alternatif *Query Plan* yang terbaik yang memiliki biaya pemrosesan *query* yang rendah berdasar TLBO. Hasil dari perengkingan ditunjukkan pada tabel 6.23.

Tabel 6.23 Tabel Perengkingan *Query Plan*

QP No	RANK	QC Cost
1	1	0.244
5	2	0.248
2	3	0.262
4	4	0.262
3	5	0.271

Pada tabel 6.23 adalah hasil akhir dari metode TLBO yang menampilkan urutan *query plan* dari yang paling optimal atau memiliki biaya paling rendah sampai biaya yang tertinggi.

6.6 Pengujian Metode *TLBO Query* 3

Pada tahap pengujian metode atau implementasi dari metode TLBO untuk *Query* 3 dimana menghasilkan data diri dan data orang tua mahasiswa yang memiliki 5 model alternatif *query*. Pada metode TLBO terdapat dua tahap yakni fase *Teacher* dan fase *Learner*. Sebelumnya kita menentukan parameter umum yang ada yaitu jumlah populasinya (*Pn*) adalah 5 sesuai dengan *Query Plan* yang telah dibuat. Selanjutnya jumlah generasi (*Gn*) atau pembaruan nilai dari parameter sebanyak 2 yakni *Update* Populasi 01 dan *Update* Populasi 02. Jumlah desain variable (Dn) disini yaitu parameter yang kita gunakan berjumlah 3 (QAC*,* QLC dan LPC) dan batas nilainya yaitu nilai minimal 0 dan maksimal 1. Perhitungan secara rinci untuk penerapan metode TLBO query 1 terdapat pada lampiran D.3.

6.6.1 Fase *Teacher*

Pada fase *teacher* ini akan dilakukan pengolahan data parameter yang telah dihitung sebelumnya untuk mengetahui pada fase ini dari data set mana yang akan berperan sebagai *teacher*. Pada tabel 6.24 berisi data parameter yang telah dihitung sebelumnya dan akan dilakukan beberapa tahap perhitungan pada fase *teacher*.

Tabel 6.24 Tabel Daftar Biaya Setiap *Query Plan*

QP	QAC	QLC	LPC
1	0.594	0.00022	0.3322
2	0.653	0.00022	0.3322
3	0.708	0.00022	0.3326
4	0.736	0.00022	0.3326
5	0.653	0.00022	0.3322

Pada tabel 6.24 berisi nilai dari masing-masing parameter QAC*,* QLC *dan* LPC bagi tiap *Query Plan*. Pada awal fase *teacher* dibutuhkan satu parameter yang berfungsi sebagai parameter yang dapat mewakili ketiga parameter yang ada sehingga memudahkan perbandingan QP yang optimal. Sehingga pada penelitian ini dipilih parameter *fitness* sebagai parameter yang mampu mewakili 3 parameter yang ada. Nilai *fitness* disini didapatkan dari hasil penghitungan nilai rata-rata dari ketiga parameter. Dan juga dibutuhkan nilai *mean* dari masing-masing parameter sehingga dihitung juga nilai rata-rata dari parameter QAC*,* QLC *dan LPC.* Berikut adalah tabel 6.25 yang berisi parameter dengan nilai *fitness* dan *mean*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 6.25 Tabel Nilai *Fitness* dan Nilai *Mean Query Plan*

QP	QAC	QLC	LPC	FITNESS
1	0.594	0.00022	0.3322	0.308
2	0.653	0.00022	0.3322	0.328
3	0.708	0.00022	0.3326	0.347
4	0.736	0.00022	0.3326	0.356
5	0.653	0.00022	0.3322	0.328
MEAN	**0.6688**	**0.00022**	**0.33236**	

Dari tabel 6.25 dia atas sudah terdapat nilai *fitness* dan juga nilai *mean* yang akan kita gunakan untuk perhitungan selanjutnya. Kita akan memilih *Teacher* untuk digunakan merubah nilai dari *Mean* yang ada agar menjadi lebih baik. *Teacher* kita pilih dari *Query Plan (learner)* yang dianggap sebagai *learner* yang memiliki pengetahuan tinggi. Dengan cara mengidentifikasi keseluruhan *learner*, *learner* terbaik berdasar pada nilai dari *learner* tersebut yang diwakili parameter yang ada, *learner* dengan nilai *fitness* terkecil merupakan *learner* terbaik dan dianggap menjadi *teacher*. Dari tabel 6.25 kita memilih *QP* yang diasumsikan sebagai *teacher* berdasar nilai *fitness* yang terkecil sehingga *QP* yang terpilih adalah *QP* 1 sebagai *Ti*. Nilai *Mean* baru akan dihitung sebagaimana pada persamaan 2.10 *Difference Mean* dari tiap parameter yang ada pada bab 2. Hasil dari perhitungan terdapat pada tabel 6.26.

Tabel 6.26 Tabel *Teachng Factor* atau *Difference Factor*

Difference Factor (Tf)	
QAC	-0.0435
QLC	0
LPC	-0.00005

Dengan Tf pada tabel 6.26 yang telah dihitung maka akan dilakukan *update Query Plan* yang ada menjeadi generasi baru. Nilai pada setiap parameter masing-masing *Query Plan* dimodifikasi berdasarkan nilai Tf yang telah dihitung sehingga menjadi *Update* Populasi 01. pada persamaan 2.10 untuk mengupdate populasi berada pada bab 2. *Update* Populasi terbaru ditunjukkan pada tabel 6.27.

Tabel 6.27 Tabel *Update Population* 01 Berdasar *Tf*

QP	QAC	QLC	LPC
1	0.5505	0.00022	0.33215
2	0.6095	0.00022	0.33215

3	0.6645	0.00022	0.33255
4	0.6925	0.00022	0.33255
5	0.6095	0.00022	0.33215

6.6.2 Fase *Learner*

Fase selanjutnya adalah fase *learner*, dimana *learner (Query plan)* meningkatkan pengetahuan mereka dengan berinteraksi antar sesama *learner*. Setiap *learner* berinteraksi secara random dengan *learner* lainnya untuk memperluas pengetahuan mereka. *Update* Populasi 01 digunakan sebagai data masukan pada fase *learner* ini yang ditampilkan pada tabel 6.28.

Tabel 6.28 Tabel Nilai *Fitness Population* 01

QP	QAC	QLC	LPC	Fitness
1	0.5505	0.00022	0.33215	0.294
2	0.6095	0.00022	0.33215	0.314
3	0.6645	0.00022	0.33255	0.332
4	0.6925	0.00022	0.33255	0.341
5	0.6095	0.00022	0.33215	0.314

Secara random kita memilih *learner* dan membandingkan dengan *learner* lain, pemenang pada proses perbandingan (*QP1* lebih baik dari *QP* 2 dan sebaliknya) akan mengupdate *learner* yang lebih lemah dan memperbaiki nilainya. Disini learner dengan nilai lebih rendah merupakan *QP* yang menang dan dengan *QP* tersebut kita akan meningkatkan nilai dari *QP* yang memiliki nilai tinggi. Meningkatkan disini berarti mengurangi nilai/biaya agar lebih rendah. Proses interaksi dan penurunan nilai digambarkan dalam gambar 2.22. Hasil dari interaksi antar *learner* akan menghasilkan *Update* Populasi 02 yang ditampilkan pada tabel 6.29.

Tabel 6.29 Tabel *Update Population* 02 Berdasar Interaksi Antar *Learner*

QP	QAC	QLC	LPC	Fitness
1	0.5505	0.00022	0.33215	0.294
2	0.5609	0.00022	0.33215	0.297
3	0.5706	0.00022	0.33255	0.301
4	0.5755	0.00022	0.33226	0.302
5	0.5609	0.00022	0.33226	0.297

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 6.29 merupakan hasil akhir dari iterasi metode TLBO. Selanjutnya kita akan merangking *learner (Query Plan)* dengan berdasar nilai *fitness* yang terbaik. *QP* yang menempati posisi pertama merupakan alternatif *Query Plan* yang terbaik yang memiliki biaya pemrosesan *query* yang rendah berdasar TLBO. Hasil dari perengkingan ditunjukkan pada tabel 6.30.

Tabel 6.30 Tabel Perangkingan *Query Plan*

QP No	RANK	QC Cost
1	1	0.294
2	2	0.314
5	3	0.314
3	4	0.332
4	5	0.341

Pada tabel 6.30 adalah hasil akhir dari metode TLBO yang menampilkan urutan *query plan* dari yang paling optimal atau memiliki biaya paling rendah sampai biaya yang tertinggi.

6.7 Analisis Hasil

Pada penggunaan *query* 1 yang digunakan untuk menampilkan data KHS Mahasiswa, pada *query* 1 terdapat 5 model *query* yang *equivalent* dimana 5 *query* tersebut berbeda tetapi bertujuan untuk menghasilkan data yang sama. Hal yang membedakan dari kelima *query* tersebut adalah perbedaan jumlah *site* dan relasi dari setiap *site* yang berbeda. Tetapi pada *query* 1 ini kelima model melibatkan jumlah *site* yang sama seperti yang telah dijelaskan pada saat pengujian *Query* 1 sehingga dalam proses pemilihan *query* yang optimal untuk *query* 1 faktor yang dipertimbangkan adalah jumlah dan/atau konsentrasi relasi pada setiap *site*, meskipun jumlah *site* sama tetapi jumlah relasi yang terlibat pada tiap *site* berbeda maka akan menghasilkan biaya pemrosesan *query* yang berbeda. Untuk model 1 jumlah relasi yang terlibat 5 relasi pada *site* Mahasiswa dan 3 relasi pada *site* KHS, untuk model 2 jumlah relasi yang terlibat 1 relasi pada *site* Mahasiswa dan 7 relasi pada *site* KHS, untuk model 3 jumlah relasi yang terlibat adalah 4 relasi pada *site* Mahasiswa dan 4 relasi pada *site* KHS, untuk model 4 jumlah relasi yang terlibat adalah 2 relasi pada *site* Mahasiswa dan 6 relasi pada *site* KHS dan model 5 jumlah relasi yang terlibat adalah 3 relasi pada *site* Mahasiswa dan 5 relasi pada *site* KHS. Hasil dari penggunaan metode TLBO untuk membandingkan antar model *query* tersebut mana yang paling optimal yakni melalui *update population* 01 dan 02 sehingga menghasilkan hasil akhir biaya *query* yang ditampilkan pada tabel 6.15 dan dilakukan perangkingan pada tabel 6.16 maka model *query* yang paling optimal untuk *query* 1 adalah model 2. Hasil tersebut dapat dilihat dari parameter yang digunakan untuk menghitung biaya dari masing-masing *query*, pada

parameter QAC model *query* 2 dikatakan paling optimal dari kelima model tersebut karena jumlah relasi yang terlibat lebih banyak terkonsentrasi pada 1 *site* bila dibandingkan dengan model lain, seperti yang dijabarkan pada parameter QAC untuk jumlah *site* yang terlibat sama maka konsentrasi relasi yang dilihat, untuk *site* yang memiliki konsentrasi dari relasi yang sama yang terlibat lebih tinggi maka lebih baik, pada perhitungan QAC juga menunjukkan bahwa model 2 biayanya paling kecil karena banyak relasi yang terkonsentrasi pada 1 *site*. Untuk parameter QLC adalah menghitung biaya jumlah relasi yang terlibat pada klausa FROM, sehingga untuk kelima model *query* dimana memiliki jumlah relasi yang sama pada klausa FROM sehingga nilai dari parameter QLC untuk kelima model sama. Sedangkan parameter LPC yang berfokus pada relasi basis data yang tersimpan pada *site* lokal dan operator *selectivity* yang digunakan, sehingga pada parameter ini menghitung biaya pemrosesan relasi pada setiap *site*, pada *query* 1 pada kelima model menggunakan relasi serta operator *selectivity* yang sama sehingga nilai LPC dari kelima model sama.

Begitu juga pada *query* 2 yang digunakan untuk menampilkan data Keuangan Mahasiswa, pada *query* 2 terdapat 5 model *query* yang *equivalent* dimana 5 *query* tersebut berbeda tetapi bertujuan untuk menghasilkan data yang sama. Hal yang membedakan dari kelima *query* tersebut adalah perbedaan jumlah *site* dan relasi dari setiap *site* yang berbeda. Untuk model 1 jumlah *site* yang terlibat adalah 2 dan relasi yang terlibat 5 relasi pada *site* Mahasiswa dan 3 relasi pada *site* Keuangan, untuk model 2 jumlah *site* yang terlibat adalah 3 dan relasi yang terlibat 1 relasi pada *site* Mahasiswa, 3 relasi pada *site* Keuangan dan 4 relasi pada *site* KHS, untuk model 3 jumlah *site* yang terlibat adalah 3 dan relasi yang terlibat adalah 2 relasi pada *site* Mahasiswa, 3 relasi pada *site* Keuangan dan 3 relasi pada *site* KHS, untuk model 4 jumlah *site* yang terlibat adalah 3 dan relasi yang terlibat adalah 4 relasi pada *site* Mahasiswa, 3 relasi pada *site* Keuangan dan 1 relasi pada *site* KHS dan model 5 jumlah *site* yang terlibat adalah 2 dan relasi yang terlibat adalah 5 relasi pada *site* Mahasiswa dan 4 relasi pada *site* Keuangan. Hasil dari penggunaan metode TLBO untuk membandingkan antar model *query* tersebut mana yang paling optimal yakni melalui *update population* 01 dan 02 sehingga menghasilkan hasil akhir biaya *query* yang ditampilkan pada tabel 6.22 dan dilakukan perangkingan pada tabel 6.23 maka model *query* yang paling optimal untuk *query* 1 adalah model 1. Hasil tersebut dapat dilihat dari parameter yang digunakan untuk menghitung biaya dari masing-masing *query*, pada parameter QAC model *query* 1 dikatakan paling optimal dari kelima model tersebut karena melibatkan jumlah *site* yang lebih sedikit dari 3 model yang ada (model 2, 3 dan 4) yakni 2 *site* sedangkan 3 model (model 2, 3 dan 4) menggunakan 3 *site*, sedangkan untuk 1 model yakni model 5 dimana menggunakan jumlah site yang sama yakni 2 *site* maka yang dibandingkan selanjutnya adalah jumlah relasi yang terlibat pada model 1 lebih banyak terkonsentrasi pada 1 *site* bila dibandingkan dengan model 5 yang relasinya pada setiap *site* sama jumlahnya, seperti yang dijabarkan pada parameter QAC untuk jumlah *site* yang terlibat sama maka konsentrasi relasi yang dilihat, untuk *site* yang memiliki konsentrasi dari relasi yang sama yang terlibat lebih tinggi maka lebih baik, pada perhitungan QAC juga menunjukkan bahwa

model 1 biayanya paling kecil karena banyak relasi yang terkonsentrasi pada 1 *site*. Untuk parameter QLC adalah menghitung biaya jumlah relasi yang terlibat pada klausa FROM, sehingga untuk kelima model *query* dimana memiliki jumlah relasi yang sama pada klausa FROM sehingga nilai dari parameter QLC untuk kelima model sama. Sedangkan parameter LPC yang berfokus pada relasi basis data yang tersimpan pada *site* lokal dan operator *selectivity* yang digunakan, sehingga pada parameter ini menghitung biaya pemrosesan relasi pada setiap *site*, pada *query* 2 pada kelima model menggunakan relasi serta operator *selectivity* yang sama sehingga nilai LPC dari kelima model sama.

Query terakhir yakni *query* 3 yang digunakan untuk menampilkan data Mahasiswa, pada *query* 2 juga terdapat 5 model *query* yang *equivalent* dimana 5 *query* tersebut berbeda tetapi bertujuan untuk menghasilkan data yang sama. Hal yang membedakan dari kelima *query* tersebut adalah perbedaan jumlah *site* dan relasi dari setiap *site* yang berbeda. Untuk model 1 jumlah *site* yang terlibat adalah 3 dan relasi yang terlibat 5 relasi pada *site* Mahasiswa, 1 relasi pada *site* Mahasiswa Biodata dan 6 relasi pada *site* Mahasiswa Orang Tua, untuk model 2 jumlah *site* yang terlibat adalah 3 dan relasi yang terlibat 5 relasi pada *site* Mahasiswa, 4 relasi pada *site* Mahasiswa Biodata dan 3 relasi pada *site* Mahasiswa Orang Tua, untuk model 3 jumlah *site* yang terlibat adalah 4 dan relasi yang terlibat adalah 1 relasi pada *site* Mahasiswa, 4 relasi pada *site* KHS, 4 relasi pada *site* Mahasiswa Biodata dan 3 relasi pada *site* Mahasiswa Orang Tua, untuk model 4 jumlah *site* yang terlibat adalah 4 dan relasi yang terlibat adalah 2 relasi pada *site* Mahasiswa, 3 relasi pada *site* KHS, 3 relasi pada *site* Mahasiswa Biodata dan 4 relasi pada *site* Mahasiswa Orang Tua dan model 5 jumlah *site* yang terlibat adalah 3 dan relasi yang terlibat adalah 5 relasi pada *site* Mahasiswa, 3 relasi pada *site* Mahasiswa Biodata dan 4 relasi pada *site* Mahasiswa Orang Tua. Hasil dari penggunaan metode TLBO untuk membandingkan antar model *query* tersebut mana yang paling optimal yakni melalui *update population* 01 dan 02 sehingga menghasilkan hasil akhir biaya *query* yang ditampilkan pada tabel 6.29 dan dilakukan perangkingan pada tabel 6.30 maka model *query* yang paling optimal untuk *query* 1 adalah model 1. Hasil tersebut dapat dilihat dari parameter yang digunakan untuk menghitung biaya dari masing-masing *query*, pada parameter QAC model *query* 1 dikatakan paling optimal dari kelima model tersebut karena melibatkan jumlah *site* yang lebih sedikit dari 2 model yang ada (model 3 dan 4) yakni 3 *site* sedangkan 2 model (model 3 dan 4) menggunakan 4 *site*, sedangkan untuk 2 model lain yakni model 2 dan 5 dimana menggunakan jumlah *site* yang sama yakni 3 *site* maka yang dibandingkan selanjutnya adalah jumlah relasi yang terlibat pada model 1 lebih banyak terkonsentrasi pada 1 *site* bila dibandingkan dengan model 3 dan 5 yang relasinya pada setiap *site* sama banyak jumlahnya, seperti yang dijabarkan pada parameter QAC untuk jumlah *site* yang terlibat sama maka konsentrasi relasi yang dilihat, untuk *site* yang memiliki konsentrasi dari relasi yang sama yang terlibat lebih tinggi maka lebih baik, pada perhitungan QAC juga menunjukkan bahwa model 2 biayanya paling kecil karena banyak relasi yang terkonsentrasi pada 1 *site*. Untuk parameter QLC adalah menghitung biaya jumlah relasi yang terlibat pada klausa FROM, sehingga untuk kelima model *query* dimana

memiliki jumlah relasi yang sama pada klausa FROM sehingga nilai dari parameter QLC untuk kelima model sama. Sedangkan parameter LPC yang berfokus pada relasi basis data yang tersimpan pada *site* lokal dan operator *selectivity* yang digunakan, sehingga pada parameter ini menghitung biaya pemrosesan relasi pada setiap *site*, pada *query* 1 pada kelima model menggunakan relasi serta operator *selectivity* yang sama sehingga nilai LPC dari kelima model sama.

BAB 7 PENUTUP

7.1 Kesimpulan

Sebagaimana tujuan dari penelitian ini dimaksudkan untuk mengimplementasikan metode TLBO pada basis data terdistribusi untuk membandingkan beberapa *query plan* dan menentukan *query* yang paling optimal, maka dari hasil penelitian yang diperoleh, berikut adalah kesimpulan dari penelitian yang telah dilakukan :

1. Metode *Teacher Learner Based Optimization (TLBO)* diimplementasikan untuk query optimization pada basis data terdistribusi dengan membuat simulasi lingkungan basis data terdistribusi dengan merancang skema basis data yakni skema data Mahasiswa Universitas yang difrgamentasi menjadi 5 site dengan menggunakan fragmentasi vertikal dan aljabar *relational* dari *query plan* sebagai populasi metode TLBO. Skema tersebut diimplementasikan ke 5 server basis data dan memilih 1 server sebagai *site* pusat selanjutnya dilakukan federasi basis data yang memungkinkan bagi *site* pusat dapat mengakses data pada 4 *site* lain. Aljabar *relational* dari *query plan* diubah menjadi *query* dan dibuktikan *query plan* tersebut *equivalent* dan selanjutnya melakukan perhitungan biaya dari setiap *query plan* berdasar parameter heuristik yang dipilih yakni QAC*,* QLC *dan LPC.* Hasil dari perhitungan biaya setiap *query plan* merupakan masukan dari metode TLBO sebagai metode perbandingan *query plan*, selanjutnya *query plan* melalui tahap *Teacher* dan tahap *Learner* sehingga menghasilkan *query* yang paling optimal.
2. Perbedaan jumlah site dan konsentrasi relasi berpengaruh pada penentuan query yang optimal. Untuk *query* 1, *query* model yang optimal adalah query plan 2 dengan jumlah *site* 2 dan jumlah relasi pada *site* Mahasiswa 1 dan pada *site* KHS 7. *Query* 2, query model yang paling optimal adalah *query* plan 1 dengan jumlah *site* 2 dan jumlah relasi pada *site* Mahasiswa 5 dan pada *site* Keuangan 3. *Query* 3 , query model yang paling optimal adalah *query* plan 1 dengan jumlah *site* 3 dan jumlah relasi pada *site* Mahasiswa 5, pada *site* Mahasiswa Biodata 1 dan pada *site* Mahasiswa Orang Tua 6. Keseluruhan hasil *query* model yang optimal merupakan *query* model yang melibatkan *site* yang paling sedikit dibandingkan dengan *query equivalent plan* lainnya. Sedikit *site* yang terlibat menyebabkan biaya transfer data juga sedikit jika dibandingkan dengan banyak *site* yang terlibat.
3. Untuk keseluruh hasil query model yang optimal merupakan query model Parameter QAC adalah parameter yang paling berpengaruh dibandingkan parameter QLC dan LPC. Parameter QLC mengacu pada nilai dari klausa FROM sehingga setiap query plan menghasilkan nilai yang sama. Parameter LPC mengacu pada nilai site pusat dan site lokal sehingga setiap query plan menghasilkan nilai yang sama. Sedangkan parameter QAC

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

mengacu pada jumlah *site* dan konsentrasi relasi pada setiap *site* sehingga hasil setiap *query plan* berbeda.

4. Penggunaan metode TLBO merupakan bagian dari *query optimization* pada *query processing* basis data terdistribusi, karena tahapan yang dilakukan pada metode TLBO hampir serupa dengan *query optimization* secara umum yang dilakukan oleh DBMS.
   i. Inisialisasi *query plan* pada metode TLBO merupakan perancangan *search space* yang lebih luas dimana lebih bervariasi keterlibatan *site* dan konsentrasi relasi pada *query equivalent plan* daripada *search space* pada basis data yang hanya menyusun satu *query* yang diproses kebeberapa alterntif *query* dengan mengacu urutan relasi yang diakses berbeda.
   ii. Pada tahap *search strategi* yakni pemilihan parameter pengukur biaya dari setiap *query* dimana pada penelitian ini memilih parameter heuristik yakni QAC*,* QLC *dan* LPC.
   iii. Tahap *cost based optimization* dari hasil perhitungan biaya berdasar parameter heuristik yang dipilih selanjutnya masuk ke tahap *teacher* dan *learner* yang mampu menurunkan biaya pemrosesan *query* dan menghasilkan *query* yang paling optimal dari *query plan* yang telah dirancang.

7.2 Saran

Dengan hasil implementasi dari penelitian ini, terdapat beberapa saran yang dapat digunakan untuk mengembangkan penelitian ini lebih lanjut dalam penerapan dan desain basis data terdistribusi antara lain :

1. Penelitian selanjutnya dalam penerapan metode TLBO dapat menggunakan parameter lain tidak hanya terpaku pada QAC*,* QLC dan LPC yang memungkinkan hasil yang lebih maksimal dalam lingkungan basis data terdistribusi yang sangat besar ataupun kecil.
2. Pada penelitian ini parameter QLC *dan* LPC kurang menunjukkan perbandingan yang baik pada keseluruhan *query plan* sehingga dibutuhkan penelitian lebih lanjut dengan memperbaiki populasi data.
3. Pengaruh parameter QLC *dan* LPC akan lebih maksimal dengan skema basis data yang lebih besar dan fragmentasi *site* dan relasi yang lebih beragam sehingga relasi dan jumlah tupel lebih beragam dan berbeda sehingga perhitungan biaya dengan parameter heuristik tersebut lebih maksimal.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Alom, B.M., Monjurul, Henskin, F., dan Hannaford, M. 2009. *Query Processing and Optimization in Distributed Database Systems*. International Journal of Computer Science and Network Security, VOL.9 No.9. Dapat diakses : http://paper.ijcsns.org/07_book/200909/20090918.pdf.

Crepinsek, M., Liu, S., dan Mernik, L. 2012. *A Note On Teaching–Learning-Based Optimization Algorithm*. Information Sciences 212. Hal: 79–93. Dapat diakses : http://www.sciencedirect.com/science/article/pii/S0020025512003532

Elmasri, R., dan Navathe, Shamkant B. 2010. *Fundamentals of Database System. 6th Editions.* Pearson Education, Inc., Addison-Wesley.

Ghaemi, R., Fard, Amin M., Tabatabaee, H., dan Sadeghizadeh, M. 2008. *Evolutionary Query Optimization for Heterogeneous Distributed Database Systems*. International Journal of Computer Electical, Automation, Control and Information Engineering Vol:2 No:7. Dapat diakses : http://waset.org/publications/12865/evolutionary-query-optimization-for-heterogeneous-distributed-database-systems.

Kumar, T.V. Vijay, Singh, V., dan Verma, Ajak K. 2011. *Distributed Query Processing Plans Generation using Genetic Algorithm.* International Journal of Computer Theory and Engineering, Vol.3, No.1. Dapat diakses : http://www.ijcte.org/papers/280-D064.pdf.

Kumar, K. Satheesh, dan Samuel, R. Harris. 2015. *Teaching Learning based Optimization.* International Journal for Innovative Research in Science & Technology Volume 1, Issue 11. Dapat diakses : http://www.ijirst.org/articles/IJIRSTV1I11177.pdf.

Mahmudy, W.F. 2013. Algoritma Evolusi. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya, Malang. Dapat diakses : http://wayanfm.lecture.ub.ac.id/files/2014/04/2013-Algoritma-Evolusi-Modul.pdf.

Mishra, Vikash, dan Singh, Vikram. 2015. *Generating Optimal Query Plans for Distributed Query Processing using Teacher-Learner Based Optimization.* Procedia Computer Science 54. Hal. 281 – 290. Dapat diakses : http://ac.els-cdn.com/S1877050915013575/1-s2.0-S1877050915013575-main.pdf?_tid=225f8ae6-1c45-11e6-b976-00000aacb35d&acdnat=1463499573_2252c08d4f2cff973bd1f9c1770288f1.

Nugroho, Nurcahyo, Pujo. 2015. Implementasi *Materialized Query Table* Untuk Pembangunan *Data Mart* (Studi Kasus: Universitas Brawijaya Bagian Akademik). Fakultas Ilmu Komputer Universitas Brawijaya.

Prawira, Yulius, Deni. 2014. *Query Optimasi* pada Basis Data Terdistribusi dengan *Materialized Query Table*. Fakultas Ilmu Komputer Universitas Brawijaya.

Rao, R.V., Savasani, V.J., dan Vakharia, D.P. 2011. *Teaching–Learning-Based Optimization: A Novel Method For Constrained Mechanical Design Optimization Problems.* Computer-Aided Design 43. Hal. 303 – 315. Dapat diakses : http://www.sciencedirect.com/science/article/pii/S0010448510002484.

Rao, R.V., Savasani, V.J., dan Vakharia, D.P. 2012a. *Teaching–Learning-Based Optimization: An Optimization Method For Continuous Non-Linear Large Scale Problems.* Information Sciences 183. Hal. 1–15. Dapat diakses : http://www.sciencedirect.com/science/article/pii/S0020025511004191.

Rao, R.V., dan Patel, V. 2012b. *An Elitist Teaching-Learning-Based Optimization Algorithm For Solving Complex Constrained Optimization Problems.* International Journal of Industrial Engineering Computations 3 (2012) 535–560. Dapat diakses : http://www.growingscience.com/ijiec/Vol3/IJIEC_2012_37.pdf.

Seed, Saba S.E. 2013. *Query Optimization In Distributed Datbase System Using Mobile Agent.* Msc Computer Science University of Khartoum. Dapat diakses: http://khartoumspace.uofk.edu:8080/bitstream/handle/123456789/1221/Query%20Optimization%20%20in%20Distributed%20Database%20System%20%20using%20Mobile%20Agent.pdf?sequence=1&isAllowed=y.

Taylor, Robert. 2010. *Query Optimization for Distributed Database Systems.* Master of Computer Science. University of Oxford. Dapat diakses : http://www.cs.ox.ac.uk/people/dan.olteanu/theses/Robert.Taylor.pdf

Tutorials Point. 2015. Database Management System (DBMS). Dapat diakses : www.tutorialspoint.com.

Valduriez, P., & Ozsu, T. 2011. *Principle of Distributed Database System. 3rd Editions.* Pearson Educations, Inc. New York.

Victor, Li O.K. 1981. Query Processing in Distributed Data Bases, MIT. Lab. For Information and Decision System Series/Report no: LIDS-P ; 1107,. Dapat diakses : http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA103826.

LAMPIRAN A PERHITUNGAN *QUERY AFFINITY COST (QAC)*

A.1 *QUERY* 1

QP1

M = 2 K1 = 5 N = 8

K2 = 3

$$QAC = \sum_{i=1}^{2} \frac{ki}{N} \left(1 - \frac{ki}{N}\right)$$

$$= \frac{K1}{N1} \left(1 - \frac{K1}{N1}\right) + \frac{K2}{N2} \left(1 - \frac{K2}{N2}\right)$$

$$= \frac{5}{8} \left(1 - \frac{5}{8}\right) + \frac{3}{8} \left(1 - \frac{3}{8}\right)$$

$$= \frac{5}{8}\left(\frac{3}{8}\right) + \frac{3}{8}\left(\frac{5}{8}\right)$$

$$= \frac{15}{64} + \frac{15}{64}$$

$$= \frac{30}{64} = 0{,}468$$

QP2

M = 2 K1 = 1 N = 8

K2 = 7

$$QAC = \sum_{i=1}^{2} \frac{ki}{N} \left(1 - \frac{ki}{N}\right)$$

$$= \frac{K1}{N1} \left(1 - \frac{K1}{N1}\right) + \frac{K2}{N2} \left(1 - \frac{K2}{N2}\right)$$

$$= \frac{1}{8} \left(1 - \frac{1}{8}\right) + \frac{7}{8} \left(1 - \frac{7}{8}\right)$$

$$= \frac{1}{8}\left(\frac{7}{8}\right) + \frac{7}{8}\left(\frac{1}{8}\right)$$

$$= \frac{7}{64} + \frac{7}{64}$$

$$= \frac{14}{64} = 0{,}218$$

QP3

M = 2 K1 = 4 N = 8

K2 = 4

$$QAC = \sum_{i=1}^{2} \frac{ki}{N} \left(1 - \frac{ki}{N}\right)$$

$$= \frac{K1}{N1} \left(1 - \frac{K1}{N1}\right) + \frac{K2}{N2} \left(1 - \frac{K2}{N2}\right)$$

$$= \frac{4}{8} \left(1 - \frac{4}{8}\right) + \frac{4}{8} \left(1 - \frac{4}{8}\right)$$

$$= \frac{4}{8}\left(\frac{4}{8}\right) + \frac{4}{8}\left(\frac{4}{8}\right)$$

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

$= \frac{16}{64} + \frac{16}{64}$

$= \frac{32}{64} = 0{,}5$

QP4

M= 2 K1 = 2 N = 8

K2 = 6

QAC $= \sum_{i=1}^{2} \frac{ki}{N} (1 - \frac{ki}{N})$

$= \frac{K1}{N1} (1 - \frac{K1}{N1}) + \frac{K2}{N2} (1 - \frac{K2}{N2})$

$= \frac{2}{8} (1 - \frac{2}{8}) + \frac{6}{8} (1 - \frac{6}{8})$

$= \frac{2}{8} (\frac{6}{8}) + \frac{6}{8} (\frac{2}{8})$

$= \frac{12}{64} + \frac{12}{64}$

$= \frac{24}{64} = 0{,}375$

QP5

M = 2 K1 = 3 N = 8

K2 = 5

QAC $= \sum_{i=1}^{2} \frac{ki}{N} (1 - \frac{ki}{N})$

$= \frac{K1}{N1} (1 - \frac{K1}{N1}) + \frac{K2}{N2} (1 - \frac{K2}{N2})$

$= \frac{3}{8} (1 - \frac{3}{8}) + \frac{5}{8} (1 - \frac{5}{8})$

$= \frac{3}{8} (\frac{5}{8}) + \frac{5}{8} (\frac{3}{8})$

$= \frac{15}{64} + \frac{15}{64}$

$= \frac{30}{64} = 0{,}468$

A.2 *QUERY* 2

QP1

M = 2 K1 = 5 N = 8

K2 = 3

QAC $= \sum_{i=1}^{2} \frac{ki}{N} (1 - \frac{ki}{N})$

$= \frac{K1}{N1} \left(1 - \frac{K1}{N1}\right) + \frac{K2}{N2} \left(1 - \frac{K2}{N2}\right)$

$= \frac{5}{8} (1 - \frac{5}{8}) + \frac{3}{8} (1 - \frac{3}{8})$

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

$= \frac{5}{8}(\frac{3}{8}) + \frac{3}{8}(\frac{5}{8})$

$= \frac{15}{64} + \frac{15}{64}$

$= \frac{30}{64} = 0{,}468$

QP2

M = 3 K1 = 1 N = 8

K2 = 4

K3 = 3

QAC $= \sum_{i=1}^{3} \frac{ki}{N}(1 - \frac{ki}{N})$

$= \frac{K1}{N1}\left(1 - \frac{K1}{N1}\right) + \frac{K2}{N2}\left(1 - \frac{K2}{N2}\right) + \frac{K3}{N3}\left(1 - \frac{K3}{N3}\right)$

$= \frac{1}{8}(1 - \frac{1}{8}) + \frac{4}{8}(1 - \frac{4}{8}) + \frac{3}{8}(1 - \frac{3}{8})$

$= \frac{1}{8}(\frac{7}{8}) + \frac{4}{8}(\frac{4}{8}) + \frac{3}{8}(\frac{5}{8})$

$= \frac{7}{64} + \frac{16}{64} + \frac{15}{64}$

$= \frac{38}{64} = 0{,}593$

QP3

M = 3 K1 = 2 N = 8

K2 = 3

K3 = 3

QAC $= \sum_{i=1}^{3} \frac{ki}{N}(1 - \frac{ki}{N})$

$= \frac{K1}{N1}\left(1 - \frac{K1}{N1}\right) + \frac{K2}{N2}\left(1 - \frac{K2}{N2}\right) + \frac{K3}{N3}\left(1 - \frac{K3}{N3}\right)$

$= \frac{2}{8}(1 - \frac{2}{8}) + \frac{3}{8}(1 - \frac{3}{8}) + \frac{3}{8}(1 - \frac{3}{8})$

$= \frac{2}{8}(\frac{6}{8}) + \frac{3}{8}(\frac{5}{8}) + \frac{3}{8}(\frac{5}{8})$

$= \frac{12}{64} + \frac{15}{64} + \frac{15}{64}$

$= \frac{42}{64} = 0{,}656$

QP4

M = 3 K1 = 4 N = 8

K2 = 1

K3 = 3

QAC $= \sum_{i=1}^{3} \frac{ki}{N}(1 - \frac{ki}{N})$

$= \frac{K1}{N1}\left(1 - \frac{K1}{N1}\right) + \frac{K2}{N2}\left(1 - \frac{K2}{N2}\right) + \frac{K3}{N3}\left(1 - \frac{K3}{N3}\right)$

$= \frac{4}{8}\ (1 - \frac{4}{8}) + \frac{1}{8}\ (1 - \frac{1}{8}) + \frac{3}{8}\ (1 - \frac{3}{8})$

$= \frac{4}{8}(\frac{4}{8}) + \frac{1}{8}(\frac{7}{8}) + \frac{3}{8}(\frac{5}{8})$

$= \frac{16}{64} + \frac{7}{64} + \frac{15}{64}$

$= \frac{38}{64}$ = 0,593

QP5

M = 2 K1 = 5 N = 9

K2 = 4

QAC $= \sum_{i=1}^{2} \frac{ki}{N}\ (1 - \frac{ki}{N})$

$= \frac{K1}{N1}\ (1 - \frac{K1}{N1}) + \frac{K2}{N2}\ (1 - \frac{K2}{N2})$

$= \frac{5}{9}\ (1 - \frac{5}{9}) + \frac{4}{9}\ (1 - \frac{4}{9})$

$= \frac{5}{9}(\frac{4}{9}) + \frac{4}{9}(\frac{5}{9})$

$= \frac{20}{81} + \frac{20}{81}$

$= \frac{40}{81}$ = 0,494

A.3 *QUERY* 3

QP1

M = 3 K1 = 5 N = 12

K2 = 1

K3 = 6

QAC $= \sum_{i=1}^{2} \frac{ki}{N}\ (1 - \frac{ki}{N})$

$= \frac{K1}{N1}\left(1 - \frac{K1}{N1}\right) + \frac{K2}{N2}\left(1 - \frac{K2}{N2}\right)$

$= \frac{5}{12}\ (1 - \frac{5}{12}) + \frac{1}{12}\ (1 - \frac{1}{12}) + \frac{6}{12}\ (1 - \frac{6}{12})$

$= \frac{5}{12}(\frac{7}{12}) + \frac{1}{12}(\frac{11}{12}) + \frac{6}{12}(\frac{6}{12})$

$= \frac{35}{144} + \frac{11}{144} + \frac{36}{144}$

$= \frac{82}{144}$ = 0,594

QP2

M = 3 K1 = 5 N = 12

K2 = 4

K3 = 3

$$QAC = \sum_{i=1}^{3} \frac{ki}{N}\left(1-\frac{ki}{N}\right)$$

$$= \frac{K1}{N1}\left(1-\frac{K1}{N1}\right) + \frac{K2}{N2}\left(1-\frac{K2}{N2}\right) + \frac{K3}{N3}\left(1-\frac{K3}{N3}\right)$$

$$= \frac{5}{12}\left(1-\frac{5}{12}\right) + \frac{4}{12}\left(1-\frac{4}{12}\right) + \frac{3}{12}\left(1-\frac{3}{12}\right)$$

$$= \frac{5}{12}\left(\frac{7}{12}\right) + \frac{4}{12}\left(\frac{8}{12}\right) + \frac{3}{12}\left(\frac{9}{12}\right)$$

$$= \frac{35}{144} + \frac{32}{144} + \frac{27}{144}$$

$$= \frac{94}{144} = 0{,}653$$

QP3

M = 4 K1 = 1 N = 12

K2 = 4

K3 = 3

K4 = 4

$$QAC = \sum_{i=1}^{3} \frac{ki}{N}\left(1-\frac{ki}{N}\right)$$

$$= \frac{K1}{N1}\left(1-\frac{K1}{N1}\right) + \frac{K2}{N2}\left(1-\frac{K2}{N2}\right) + \frac{K3}{N3}\left(1-\frac{K3}{N3}\right) + \frac{K3}{N3}\left(1-\frac{K3}{N3}\right)$$

$$= \frac{1}{12}\left(1-\frac{1}{12}\right) + \frac{4}{12}\left(1-\frac{4}{12}\right) + \frac{3}{12}\left(1-\frac{3}{12}\right) + \frac{4}{12}\left(1-\frac{4}{12}\right)$$

$$= \frac{1}{12}\left(\frac{11}{12}\right) + \frac{4}{12}\left(\frac{8}{12}\right) + \frac{3}{12}\left(\frac{9}{12}\right) + \frac{4}{12}\left(\frac{8}{12}\right)$$

$$= \frac{11}{144} + \frac{32}{144} + \frac{27}{144} + \frac{32}{144}$$

$$= \frac{102}{144} = 0{,}708$$

QP4

M = 4 K1 = 2 N = 12

K2 = 3

K3 = 4

K4 = 3

$$QAC = \sum_{i=1}^{3} \frac{ki}{N}\left(1-\frac{ki}{N}\right)$$

$$= \frac{K1}{N1}\left(1-\frac{K1}{N1}\right) + \frac{K2}{N2}\left(1-\frac{K2}{N2}\right) + \frac{K3}{N3}\left(1-\frac{K3}{N3}\right) + \frac{K3}{N3}\left(1-\frac{K3}{N3}\right)$$

$$= \frac{2}{12}\left(1-\frac{2}{12}\right) + \frac{3}{12}\left(1-\frac{3}{12}\right) + \frac{4}{12}\left(1-\frac{4}{12}\right) + \frac{3}{12}\left(1-\frac{3}{12}\right)$$

$$= \frac{2}{12}\left(\frac{10}{12}\right) + \frac{3}{12}\left(\frac{9}{12}\right) + \frac{4}{12}\left(\frac{8}{12}\right) + \frac{3}{12}\left(\frac{9}{12}\right)$$

$$= \frac{20}{144} + \frac{27}{144} + \frac{32}{144} + \frac{27}{144}$$

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

$= \frac{106}{144} = 0{,}736$

QP5

M = 3 K1 = 5 N = 12

K2 = 3

K3 = 4

$$QAC = \sum_{i=1}^{3} \frac{ki}{N} \left(1 - \frac{ki}{N}\right)$$

$$= \frac{K1}{N1}\left(1 - \frac{K1}{N1}\right) + \frac{K2}{N2}\left(1 - \frac{K2}{N2}\right) + \frac{K3}{N3}\left(1 - \frac{K3}{N3}\right)$$

$$= \frac{5}{12}\left(1 - \frac{5}{12}\right) + \frac{3}{12}\left(1 - \frac{3}{12}\right) + \frac{4}{12}\left(1 - \frac{4}{12}\right)$$

$$= \frac{5}{12}\left(\frac{7}{12}\right) + \frac{3}{12}\left(\frac{9}{12}\right) + \frac{4}{12}\left(\frac{8}{12}\right)$$

$$= \frac{35}{144} + \frac{27}{144} + \frac{32}{144}$$

$$= \frac{94}{144} = 0{,}653$$

LAMPIRAN B PERHITUNGAN *QUERY LOCALIZATION COST (QLC)*

B.1 *QUERY* 1

QP1

Nr = 8

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5
6. KHS	65535
7. MK	31549
8. NILAI	9
T0TAL	**145422**

Tabel 7.1 Klausa WHERE pada *Query* 1 Model 1

	MHS	FAK	JUR	PR	JENJANG	KHS	MK	NILAI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KHS	1	0	0	0	0	**0**	1	1
MK	0	0	0	0	0	1	**0**	0
NILAI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{8}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{8} \frac{Size\ (Rsj)}{\sum_{k=1}^{8} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{145423}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{145423}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{145423}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{145423}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145423}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145423}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145423}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.00044 + 0.000034 + 0.451]

= 0,45147

(i = 2) FAKULTAS

$$= \left[\frac{0}{145423}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{145423}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145423}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{0}{145423}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145423}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145423}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145423}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{15}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{\quad}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{64}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= [0.331 + 0.00044]

= 0.33144

(i = 5) JENJANG

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= 0.331

(i = 6) KHS

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{315419}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)} + \frac{9}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= [0.331 + 0.217 + 0.00006]

= 0.5486

(i = 7) MATKUL

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= 0.451

(i = 8) NILAI

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)}\right]$$

= 0.451

QPcost = $MIN_{i=1}^{8}[(0.45147), (0.00044), (0.00055), (0.33144), (0.331), (0.5486), (0.451), (0.451)]$ = 0.00044

QP2

Nr = 8

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PRODI	64
5. JENJANG	5

6. KHS 65535
7. MK 31549
8. NILAI 9

T0TAL 145422

Tabel 7.2 Klausa WHERE pada *Query* 1 Model 2

	MHS	FAK	JUR	PR	JENJANG	KHS	MK	NILAI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KHS	1	0	0	0	0	**0**	1	1
MK	0	0	0	0	0	1	**0**	0
NILAI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{8}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{8}\frac{Size\ (Rsj)}{\sum_{k=1}^{8} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{145422}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.00044 + 0.000034 + 0.451]

= 0,45147

(i = 2) FAKULTAS

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{64}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{15}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{64}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{\quad}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{64}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= [0.331 + 0.00044]

= 0.33144

(i = 5) JENJANG

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 6 \,\&\&\, j \neq i\,(KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\,(MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= 0.331

(i = 6) KHS

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{315419}{145422}_{j \equiv 7 \,\&\&\, j \neq i\,(MATKUL)} + \frac{9}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= [0.331 + 0.217 + 0.00006]

= 0.5486

(i = 7) MATKUL

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\,(KHS)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= 0.451

(i = 8) NILAI

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)}\right]$$

= 0.451

QPcost = $MIN_{i=1}^{8}[(0.45147), (0.00044), (0.00055), (0.33144), (0.331), (0.5486), (0.451), (0.451)]$ = 0.00044

QP3

Nr = 8

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5
6. KHS	65535
7. MK	31549
8. NILAI	9
T0TAL	**145422**

Tabel 7.3 Klausa WHERE pada *Query* 1 Model 3

	MHS	FAK	JUR	PR	JENJANG	KHS	MK	NILAI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KHS	1	0	0	0	0	**0**	1	1
MK	0	0	0	0	0	1	**0**	0
NILAI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{8}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{8} \frac{Size\ (Rsj)}{\sum_{k=1}^{8} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{145422}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.00044 + 0.000034 + 0.451]

= 0,45147

(i = 2) FAKULTAS

$$= \left[\frac{0}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{15}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{\quad}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.331 + 0.00044]

= 0.33144

(i = 5) JENJANG

$$= \left[\frac{48181}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= 0.331

(i = 6) KHS

$$= \left[\frac{48181}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{315419}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{9}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.331 + 0.217 + 0.00006]

= 0.5486

(i = 7) MATKUL

$$= \left[\frac{0}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= 0.451

(i = 8) NILAI

$$= \left[\frac{0}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)}\right]$$

= 0.451

$QPcost = MIN_{i=1}^{8}[(0.45147), (0.00044), (0.00055), (0.33144), (0.331), (0.5486), (0.451), (0.451)]$ = 0.00044

QP4

Nr = 8

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5
6. KHS	65535
7. MK	31549
8. NILAI	9
T0TAL	**145422**

Tabel 7.4 Klausa WHERE pada *Query* 1 Model 4

	MHS	FAK	JUR	PR	JENJANG	KHS	MK	NILAI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KHS	1	0	0	0	0	**0**	1	1
MK	0	0	0	0	0	1	**0**	0
NILAI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{8}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{8} \frac{Size\ (Rsj)}{\sum_{k=1}^{8} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{145422}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.00044 + 0.000034 + 0.451]

= 0,45147

(i = 2) FAKULTAS

$$= \left[\frac{0}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{145422}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{15}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{64}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\,(KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\,(MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= [0.331 + 0.00044]

= 0.33144

(i = 5) JENJANG

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 6 \,\&\&\, j \neq i\,(KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\,(MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= 0.331

(i = 6) KHS

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{315419}{145422}_{j \equiv 7 \,\&\&\, j \neq i\,(MATKUL)} + \frac{9}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= [0.331 + 0.217 + 0.00006]

= 0.5486

(i = 7) MATKUL

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\,(KHS)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= 0.451

(i = 8) NILAI

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\,(KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\,(MATKUL)}\right]$$

= 0.451

QPcost = $MIN_{i=1}^{8}[(0.45147), (0.00044), (0.00055), (0.33144), (0.331), (0.5486), (0.451), (0.451)]$ = 0.00044

QP5

Nr = 8

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PRODI	64
5. JENJANG	5
6. KHS	65535

7. MK 31549
8. NILAI 9
T0TAL 145422

Tabel 7.5 Klausa WHERE pada *Query* 1 Model 5

	MHS	FAK	JUR	PR	JENJANG	KHS	MK	NILAI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KHS	1	0	0	0	0	**0**	1	1
MK	0	0	0	0	0	1	**0**	0
NILAI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{8}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{8}\frac{Size\ (Rsj)}{\sum_{k=1}^{8} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{145422}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{145422}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{145422}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{145422}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j\equiv 6\ \&\&\ j\neq i\ (KHS)} + \frac{0}{145422}_{j\equiv 7\ \&\&\ j\neq i\ (MATKUL)} + \frac{0}{145422}_{j\equiv 8\ \&\&\ j\neq i\ (NILAI)}\right]$$

= [0.00044 + 0.000034 + 0.451]

= 0,45147

(i = 2) FAKULTAS

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{64}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{15}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{64}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{64}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\ (MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= [0.331 + 0.00044]

= 0.33144

(i = 5) JENJANG

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 6 \,\&\&\, j \neq i\,(KHS)} + \frac{0}{145422}_{j \equiv 7 \,\&\&\, j \neq i\,(MATKUL)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= 0.331

(i = 6) KHS

$$= \left[\frac{48181}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{315419}{145422}_{j \equiv 7 \,\&\&\, j \neq i\,(MATKUL)} + \frac{9}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= [0.331 + 0.217 + 0.00006]

= 0.5486

(i = 7) MATKUL

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\,(KHS)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\,(NILAI)}\right]$$

= 0.451

(i = 8) NILAI

$$= \left[\frac{0}{145422}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{145422}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{145422}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{145422}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{65535}{145422}_{j \equiv 6 \,\&\&\, j \neq i\ (KHS)} + \frac{0}{145422}_{j \equiv 8 \,\&\&\, j \neq i\ (NILAI)}\right]$$

= 0.451

QPcost = $MIN_{i=1}^{8}[(0.45147), (0.00044), (0.00055), (0.33144), (0.331), (0.5486), (0.451), (0.451)]$ = 0.00044

B.2 *QUERY* 2

QP1

Nr = 8

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5
6. KEUANGAN	96362
7. ITEM	3
8. TRANSAKSI	2
T0TAL	**144696**

Tabel 7.6 Klausa WHERE pada *Query* 2 Model 1

	MHS	FAK	JUR	PR	JENJANG	KEUANGAN	ITEM	TRANSAKSI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KEUANGAN	1	0	0	0	0	**0**	1	1
ITEM	0	0	0	0	0	1	**0**	0
TRANSAKSI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{8}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{8} \frac{Size\ (Rsj)}{\sum_{k=1}^{8} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{144696}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144696}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{144696}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{144696}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{96362}{144696}_{j\equiv 6\ \&\&\ j\neq i\ (KEUANGAN)} + \frac{0}{144696}_{j\equiv 7\ \&\&\ j\neq i\ (ITEM)} + \frac{0}{144696}_{j\equiv 8\ \&\&\ j\neq i\ (TRANSAKSI)}\right]$$

= [0.00044 + 0.000034 + 0.666]

= 0.666474

(i = 2) FAKULTAS

$$= \left[\frac{0}{144696}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{144696}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144696}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{0}{144696}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{15}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{64}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{64}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= [0.332 + 0.00044]

= 0.33244

(i = 5) JENJANG

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= 0.332

(i = 6) KEUANGAN

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \right.$$

$$\left.\frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{3}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{2}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= [0.332 + 0.00002 + 0.00001]

= 0.33203

(i = 7) ITEM

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \right.$$

$$\left.\frac{96362}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= 0.666

(i = 8) TRANSAKSI

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \right.$$

$$\left.\frac{96362}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{192878}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)}\right]$$

= 0.666

$QPcost = MIN_{i=1}^{8}[(0.666474), (0.00044), (0.00055), (0.33244), (0.332), (0.333203), (0.666), (0.666)]$ = 0.00044

QP2

Nr = 8

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5
6. KEUANGAN	96362
7. ITEM	3
8. TRANSAKSI	2
T0TAL	**144696**

Tabel 7.7 Klausa WHERE pada *Query* 2 Model 2

	MHS	FAK	JUR	PR	JENJANG	KEUANGAN	ITEM	TRANSAKSI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KEUANGAN	1	0	0	0	0	**0**	1	1
ITEM	0	0	0	0	0	1	**0**	0
TRANSAKSI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{8}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{8} \frac{Size\ (Rsj)}{\sum_{k=1}^{8} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{144696}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144696}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{144696}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{144696}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{96362}{144696}_{j\equiv 6\ \&\&\ j\neq i\ (KEUANGAN)} + \frac{0}{144696}_{j\equiv 7\ \&\&\ j\neq i\ (ITEM)} + \frac{0}{144696}_{j\equiv 8\ \&\&\ j\neq i\ (TRANSAKSI)}\right]$$

= [0.00044 + 0.000034 + 0.666]

= 0.666474

(i = 2) FAKULTAS

$$= \left[\frac{0}{144696}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{144696}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144696}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{0}{144696}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j\equiv 6\ \&\&\ j\neq i\ (KEUANGAN)} + \frac{0}{144696}_{j\equiv 7\ \&\&\ j\neq i\ (ITEM)} + \frac{0}{144696}_{j\equiv 8\ \&\&\ j\neq i\ (TRANSAKSI)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{144696}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{15}{144696}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{144696}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{0}{144696}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j\equiv 6\ \&\&\ j\neq i\ (KEUANGAN)} + \frac{0}{144696}_{j\equiv 7\ \&\&\ j\neq i\ (ITEM)} + \frac{0}{144696}_{j\equiv 8\ \&\&\ j\neq i\ (TRANSAKSI)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{64}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= [0.332 + 0.00044]

= 0.33244

(i = 5) JENJANG

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= 0.332

(i = 6) KEUANGAN

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{3}{144696}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{2}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= [0.332 + 0.00002 + 0.00001]

= 0.33203

(i = 7) ITEM

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{96362}{144696}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= 0.666

(i = 8) TRANSAKSI

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{96362}{144696}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{192878}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)}\right]$$

= 0.666

$QPcost = MIN_{i=1}^{8}[(0.666474), (0.00044), (0.00055), (0.33244), (0.332), (0.333203), (0.666), (0.666)]$

QP3

Nr = 8

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5

6. KEUANGAN	96362
7. ITEM	3
8. TRANSAKSI	2
T0TAL	**144696**

Tabel 7.8 Klausa WHERE pada *Query* 2 Model 3

	MHS	FAK	JUR	PR	JENJANG	KEUANGAN	ITEM	TRANSAKSI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KEUANGAN	1	0	0	0	0	**0**	1	1
ITEM	0	0	0	0	0	1	**0**	0
TRANSAKSI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{8}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{8} \frac{Size\ (Rsj)}{\sum_{k=1}^{8} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{144696}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144696}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{144696}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{144696}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{96362}{144696}_{j\equiv 6\ \&\&\ j\neq i\ (KEUANGAN)} + \frac{0}{144696}_{j\equiv 7\ \&\&\ j\neq i\ (ITEM)} + \frac{0}{144696}_{j\equiv 8\ \&\&\ j\neq i\ (TRANSAKSI)}\right]$$

= [0.00044 + 0.000034 + 0.666]

= 0.666474

(i = 2) FAKULTAS

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{64}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{15}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{64}{144696}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{64}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= [0.332 + 0.00044]

= 0.33244

(i = 5) JENJANG

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= 0.332

(i = 6) KEUANGAN

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{3}{144696}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{2}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= [0.332 + 0.00002 + 0.00001]

= 0.33203

(i = 7) ITEM

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{96362}{144696}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)}\right]$$

= 0.666

(i = 8) TRANSAKSI

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \right]$$

$$\left[\frac{96362}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{192878}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)}\right]$$

= 0.666

QPcost = $MIN_{i=1}^{8}[(0.666474), (0.00044), (0.00055), (0.33244), (0.332), (0.333203), (0.666), (0.666)]$

QP4

Nr = 8

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5
6. KEUANGAN	96362
7. ITEM	3
8. TRANSAKSI	2
T0TAL	**144696**

Tabel 7.9 Klausa WHERE pada *Query* 2 Model 4

	MHS	FAK	JUR	PR	JENJANG	KEUANGAN	ITEM	TRANSAKSI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KEUANGAN	1	0	0	0	0	**0**	1	1
ITEM	0	0	0	0	0	1	**0**	0
TRANSAKSI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{8}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{8} \frac{Size\ (Rsj)}{\sum_{k=1}^{8} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{144696}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144696}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{144696}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{144696}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{96362}{144696}_{j\equiv 6\ \&\&\ j\neq i\ (KEUANGAN)} + \frac{0}{144696}_{j\equiv 7\ \&\&\ j\neq i\ (ITEM)} + \frac{0}{144696}_{j\equiv 8\ \&\&\ j\neq i\ (TRANSAKSI)}\right]$$

= [0.00044 + 0.000034 + 0.666]

= 0.666474

(i = 2) FAKULTAS

$$= \left[\frac{0}{144696}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{144696}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144696}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{0}{144696}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{15}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{64}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{64}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= [0.332 + 0.00044]

= 0.33244

(i = 5) JENJANG

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \right]$$

$$\left[\frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= 0.332

(i = 6) KEUANGAN

$$= \left[\frac{48181}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \right.$$

$$\left.\frac{0}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{3}{144696}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)} + \frac{2}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= [0.332 + 0.00002 + 0.00001]

= 0.33203

(i = 7) ITEM

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \right.$$

$$\left.\frac{96362}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{144696}_{j \equiv 8 \,\&\&\, j \neq i\ (TRANSAKSI)}\right]$$

= 0.666

(i = 8) TRANSAKSI

$$= \left[\frac{0}{144696}_{j \equiv 1 \,\&\&\, j \neq i\ (MAHASISWA)} + \frac{0}{144696}_{j \equiv 2 \,\&\&\, j \neq i\ (FAKULTAS)} + \frac{0}{144696}_{j \equiv 3 \,\&\&\, j \neq i\ (JURUSAN)} + \frac{0}{144696}_{j \equiv 4 \,\&\&\, j \neq i\ (PRoDI)} + \right.$$

$$\left.\frac{96362}{144696}_{j \equiv 5 \,\&\&\, j \neq i\ (JENJANG)} + \frac{0}{144696}_{j \equiv 6 \,\&\&\, j \neq i\ (KEUANGAN)} + \frac{0}{192878}_{j \equiv 7 \,\&\&\, j \neq i\ (ITEM)}\right]$$

= 0.666

$QPcost = MIN_{i=1}^{8}[(0.666474), (0.00044), (0.00055), (0.33244), (0.332), (0.333203), (0.666), (0.666)]$

QP5

Nr = 9

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5
6. KEUANGAN	96362
7. ITEM	3
8. TRANSAKSI	2
9. MAHASISWA	48181
T0TAL	192877

Tabel 7.10 Klausa WHERE pada *Query* 2 Model 5

	MHS	FAK	JUR	PR	JENJANG	KEUANGAN	ITEM	TRANSAKSI
MHS	**0**	0	0	1	1	1	0	0
FAK	0	**0**	1	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0
PR	1	0	1	**0**	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0
KEUANGAN	1	0	0	0	0	**0**	1	1
ITEM	0	0	0	0	0	1	**0**	0
TRANSAKSI	0	0	0	0	0	1	0	**0**

$$QPcost = MIN_{i=1}^{9}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{9} \frac{Size\ (Rsj)}{\sum_{k=1}^{9} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{192877}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{192877}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{192877}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{5}{192877}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{192877}_{j\equiv 6\ \&\&\ j\neq i\ (KEUANGAN)} + \frac{0}{192877}_{j\equiv 7\ \&\&\ j\neq i\ (ITEM)} + \frac{0}{192877}_{j\equiv 8\ \&\&\ j\neq i\ (TRANSAKSI)} + \frac{48181}{192877}_{j\equiv 9\ \&\&\ j\neq i\ (MAHASISWA)}\right]$$

= [0.00033 + 0.00002 + 0.249]

= 0.24935

(i = 2) FAKULTAS

$$= \left[\frac{0}{192877}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{192877}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{192877}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{0}{192877}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{192877}_{j\equiv 6\ \&\&\ j\neq i\ (KEUANGAN)} + \frac{0}{192877}_{j\equiv 7\ \&\&\ j\neq i\ (ITEM)} + \frac{0}{192877}_{j\equiv 8\ \&\&\ j\neq i\ (TRANSAKSI)} + \frac{0}{192877}_{j\equiv 9\ \&\&\ j\neq i\ (MAHASISWA)}\right]$$

= 0.00033

(i = 3) JURUSAN

$$= \left[\frac{0}{192877}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{15}{192877}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{192877}_{j\equiv 4\ \&\&\ j\neq i\ (PRoDI)} + \frac{0}{192877}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \right]$$

$$\left[\frac{0}{192877}_{j\equiv 6\ \&\&\ j\neq i\ (KEUANGAN)} + \frac{0}{192877}_{j\equiv 7\ \&\&\ j\neq i\ (ITEM)} + \frac{0}{192877}_{j\equiv 8\ \&\&\ j\neq i\ (TRANSAKSI)} + \frac{0}{192877}_{j\equiv 9\ \&\&\ j\neq i\ (MAHASISWA)}\right]$$

= [0.00008 + 0.00033]

= 0.00041

(i = 4) PRODI

$$= \left[\frac{48181}{192877}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{192877}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{64}{192877}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{192877}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \right]$$

$$\left[\frac{0}{192877}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{192877}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{192877}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)} + \frac{0}{192877}_{j \equiv 9 \,\&\&\, j \neq i\,(MAHASISWA)} \right]$$

= [0.249 + 0.00033]

= 0.24933

(i = 5) JENJANG

$$= \left[\frac{48181}{192877}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{192877}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{192877}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{192877}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{0}{192877}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{192877}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{192877}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)} + \frac{0}{192877}_{j \equiv 9 \,\&\&\, j \neq i\,(MAHASISWA)} \right]$$

= 0.249

(i = 6) KEUANGAN

$$= \left[\frac{0}{192877}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{192877}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{192877}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{192877}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{0}{192877}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{3}{192877}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{2}{192877}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)} + \frac{48181}{192877}_{j \equiv 9 \,\&\&\, j \neq i\,(MAHASISWA)} \right]$$

= [0.00001 + 0.00001 + 0.249]

= 0.24902

(i = 7) ITEM

$$= \left[\frac{0}{192877}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{192877}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{192877}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{192877}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{96362}{192877}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{192877}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{192877}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)} + \frac{0}{192877}_{j \equiv 9 \,\&\&\, j \neq i\,(MAHASISWA)} \right]$$

= 0.499

(i = 8) TRANSAKSI

$$= \left[\frac{0}{192877}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{192877}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{192877}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{192877}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{96362}{192877}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{192877}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{192877}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{192877}_{j \equiv 9 \,\&\&\, j \neq i\,(MAHASISWA)} \right]$$

= 0.499

(i = 9) MAHASISWA

$$= \left[\frac{0}{192877}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{192877}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{192877}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{192877}_{j \equiv 4 \,\&\&\, j \neq i\,(PRoDI)} + \right]$$

$$\left[\frac{96362}{192877}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{192877}_{j \equiv 6 \,\&\&\, j \neq i\,(KEUANGAN)} + \frac{0}{192877}_{j \equiv 7 \,\&\&\, j \neq i\,(ITEM)} + \frac{0}{192877}_{j \equiv 8 \,\&\&\, j \neq i\,(TRANSAKSI)} \right]$$

= 0.499

QPcost = $MIN_{i=1}^{8}[(0.24935), (0.00033), (0.00041), (0.24933), (0.249), (0.24902), (0.499), (0.499), (0.499)]$ = 0.00033

B.3 *QUERY* 3

QP1

Nr = 12

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PRODI	64
5. JENJANG	5
6. MHS_BIODATA	48181
7. MHS_ORTU	48181
8. KOTA	131
9. NEGARA	2
10. PROPINSI	33
11. PEKERJAAN	51
12. PEKERJAAN	51
TOTAL	**144959**

Tabel 7.11 Klausa WHERE pada *Query* 3 Model 1

	MHS	FAK	JUR	PR	JENJANG	MHS_BIO	MHS_ORTU	KOTA	NEGARA	PROPINSI	PEKERJAAN	PEKERJAAN
MHS	**0**	0	0	1	1	1	1	0	0	0	0	0
FAK	0	**0**	1	0	0	0	0	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0	0	0	0	0
PR	1	0	1	**0**	0	0	0	0	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0	0	0	0	0
MHS_BIO	1	0	0	0	0	**0**	0	0	0	0	0	0

MHS_0RTU	1	0	0	0	0	0	**0**	1	0	0	1	1
K0TA	0	0	0	0	0	0	1	**0**	0	1	0	0
NEGARA	0	0	0	0	0	0	0	0	**0**	1	0	0
PR0PINSI	0	0	0	0	0	0	0	1	1	**0**	0	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	**0**	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	0	**0**

$$QPcost = MIN_{i=1}^{12}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{12} \frac{Size\ (Rsj)}{\sum_{k=1}^{12} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{5}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{48181}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.00044 + 0.000034 + 0.332 + 0.332]

= 0.664474

(i = 2) FAKULTAS

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

$= 0.00044$

(i = 3) JURUSAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{15}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.332 + 0.00044]

= 0.33344

(i = 5) JENJANG

$$= \left[\frac{48181}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right] =$$

= 0.332

(i = 6) MHS_BIO

$$= \left[\frac{48181}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right] =$$

== 0 .332

(i = 7) MHS_ORTU

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{131}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{51}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{51}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)}\right]$$

= [0.332 + 0.0009 + 0.00035 + 0.00035]

= 0.3336

(i = 8) KOTA

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{48181}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \right]$$

$$\left[\frac{33}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)}\right]$$

= [0.332 + 0.00022]

= 0.33222

(i = 9) PROPINSI

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{131}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{2}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)}\right]$$

= [0.0009 + 0.00001]

= 0.00091

(i = 10) NEGARA

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{33}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)}\right]$$

= 0.00022

(i = 11) PEKERJAAN

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right] =$$

= 0.332

(i = 12) PEKERJAAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.332

$$QPcost = MIN_{i=1}^{12}\left[\begin{matrix}(0.664), (0.00044), (0.00055), (0.33244), (0.332), (0.332),\\ (0.3326), (0.3322), (0.00022), (0.00091), (0.332), (0.332)\end{matrix}\right] = 0.00022$$

QP2

Nr = 12

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64

4. PRODI	64
5. JENJANG	5
6. MHS_BIODATA	48181
7. MHS_ORTU	48181
8. KOTA	131
9. NEGARA	2
10. PROPINSI	33
11. PEKERJAAN	51
12. PEKERJAAN	51
TOTAL	**144959**

Tabel 7.12 Klausa WHERE pada *Query* 3 Model 2

	MHS	FAK	JUR	PR	JENJANG	MHS_BIO	MHS_ORTU	KOTA	NEGARA	PROPINSI	PEKERJAAN	PEKERJAAN
MHS	**0**	0	0	1	1	1	1	0	0	0	0	0
FAK	0	**0**	1	0	0	0	0	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0	0	0	0	0
PR	1	0	1	**0**	0	0	0	0	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0	0	0	0	0
MHS_BIO	1	0	0	0	0	**0**	0	1	0	0	0	0
MHS_ORTU	1	0	0	0	0	0	**0**	0	0	0	1	1
KOTA	0	0	0	0	0	1	0	**0**	0	1	0	0
NEGARA	0	0	0	0	0	0	0	0	**0**	1	0	0
PROPINSI	0	0	0	0	0	0	0	1	1	**0**	0	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	**0**	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	0	**0**

$$QPcost = MIN_{i=1}^{12}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{12} \frac{Size\ (Rsj)}{\sum_{k=1}^{12} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{5}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{48181}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.00044 + 0.000034 + 0.332 + 0.332]

= 0.664474

(i = 2) FAKULTAS

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{15}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{64}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{64}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right]$$

= [0.332 + 0.00044]

= 0.33344

(i = 5) JENJANG

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right] =$$

= 0.332

(i = 6) MHS_BIO

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{131}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right] =$$

= [0.332 + 0.0009] =

= 0 .3329

(i = 7) MHS_ORTU

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{51}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{51}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.332 + 0.00035 + 0.00035]

= 0.3327

(i = 8) KOTA

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{48181}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \right]$$

$$\left[\frac{33}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.332 + 0.00022]

= 0.33222

(i = 9) PROPINSI

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{131}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{2}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.0009 + 0.00001]

= 0.00091

(i = 10) NEGARA

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{33}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.00022

(i = 11) PEKERJAAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right] =$$

= 0.332

(i = 12) PEKERJAAN

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{48181}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} \right]$$

= 0.332

$$QPcost = MIN_{i=1}^{12}\left[\begin{matrix}(0.664), (0.00044), (0.00055), (0.33244), (0.332), (0.3329), \\ (0.3327), (0.3322), (0.00022), (0.00091), (0.332), (0.332)\end{matrix}\right] = 0.00022$$

QP3

Nr = 12

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5
6. MHS_BI0DATA	48181
7. MHS_0RTU	48181
8. K0TA	131
9. NEGARA	2
10. PR0PINSI	33

11. PEKERJAAN 51
12. PEKERJAAN 51
T0TAL 144959

Tabel 7.13 Klausa WHERE pada *Query* 3 Model 3

	MHS	FAK	JUR	PR	JENJANG	MHS_BI0	MHS_0RTU	K0TA	NEGARA	PR0PINSI	PEKERJAAN	PEKERJAAN
MHS	**0**	0	0	1	1	1	1	0	0	0	0	0
FAK	0	**0**	1	0	0	0	0	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0	0	0	0	0
PR	1	0	1	**0**	0	0	0	0	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0	0	0	0	0
MHS_BI0	1	0	0	0	0	**0**	0	1	0	0	0	0
MHS_0RTU	1	0	0	0	0	0	**0**	0	0	0	1	1
K0TA	0	0	0	0	0	1	0	**0**	0	1	0	0
NEGARA	0	0	0	0	0	0	0	0	**0**	1	0	0
PR0PINSI	0	0	0	0	0	0	0	1	1	**0**	0	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	**0**	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	0	**0**

$$QPcost = MIN_{i=1}^{12}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{12} \frac{Size\ (Rsj)}{\sum_{k=1}^{12} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{5}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{48181}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.00044 + 0.000034 + 0.332 + 0.332]

= 0.664474

(i = 2) FAKULTAS

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{15}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.332 + 0.00044]

= 0.33344

(i = 5) JENJANG

$$= \left[\frac{48181}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right] =$$

= 0.332

(i = 6) MHS_BIO

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{131}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)}\right] =$$

= [0.332 + 0.0009] =

= 0 .3329

(i = 7) MHS_ORTU

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{51}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{51}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)}\right]$$

= [0.332 + 0.00035 + 0.00035]

= 0.3327

(i = 8) KOTA

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{48181}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \right]$$

$$\left[\frac{33}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)}\right]$$

= [0.332 + 0.00022]

= 0.33222

(i = 9) PROPINSI

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{131}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{2}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)}\right]$$

= [0.0009 + 0.00001]

= 0.00091

(i = 10) NEGARA

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{33}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right]$$

= 0.00022

(i = 11) PEKERJAAN

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{48181}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right] =$$

= 0.332

(i = 12) PEKERJAAN

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.332

$$QPcost = MIN_{i=1}^{12}\left[\begin{matrix}(0.664), (0.00044), (0.00055), (0.33244), (0.332), (0.3329), \\ (0.3327), (0.3322), (0.00022), (0.00091), (0.332), (0.332)\end{matrix}\right] = 0.00022$$

QP4

Nr = 12

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PR0DI	64
5. JENJANG	5
6. MHS_BI0DATA	48181
7. MHS_0RTU	48181
8. K0TA	131
9. NEGARA	2
10. PR0PINSI	33
11. PEKERJAAN	51
12. PEKERJAAN	51
T0TAL	**144959**

Tabel 7.14 Klausa WHERE pada *Query* 3 Model 4

	MHS	FAK	JUR	PR	JENJANG	MHS_BIO	MHS_ORTU	KOTA	NEGARA	PROPINSI	PEKERJAAN	PEKERJAAN
MHS	**0**	0	0	1	1	1	1	0	0	0	0	0
FAK	0	**0**	1	0	0	0	0	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0	0	0	0	0
PR	1	0	1	**0**	0	0	0	0	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0	0	0	0	0
MHS_BIO	1	0	0	0	0	**0**	0	1	0	0	0	0
MHS_ORTU	1	0	0	0	0	0	**0**	0	0	0	1	1
KOTA	0	0	0	0	0	1	0	**0**	0	1	0	0
NEGARA	0	0	0	0	0	0	0	0	**0**	1	0	0
PROPINSI	0	0	0	0	0	0	0	1	1	**0**	0	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	**0**	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	0	**0**

$$QPcost = MIN_{i=1}^{12}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{12} \frac{Size\ (Rsj)}{\sum_{k=1}^{12} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{5}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{48181}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.00044 + 0.000034 + 0.332 + 0.332]

= 0.664474

(i = 2) FAKULTAS

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.00044

(i = 3) JURUSAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{15}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{64}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right]$$

= [0.332 + 0.00044]

= 0.33344

(i = 5) JENJANG

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right] =$$

= 0.332

(i = 6) MHS_BIO

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{131}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right] =$$

= [0.332 + 0.0009] =

= 0 .3329

(i = 7) MHS_ORTU

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{51}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{51}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right]$$

= [0.332 + 0.00035 + 0.00035]

= 0.3327

(i = 8) KOTA

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{48181}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \right]$$

$$\left[\frac{33}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.332 + 0.00022]

= 0.33222

(i = 9) PROPINSI

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{131}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{2}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.0009 + 0.00001]

= 0.00091

(i = 10) NEGARA

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{33}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.00022

(i = 11) PEKERJAAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right] =$$

= 0.332

(i = 12) PEKERJAAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.332

$$QPcost = MIN_{i=1}^{12}\begin{bmatrix}(0.664),(0.00044),(0.00055),(0.33244),(0.332),(0.3329),\\(0.3327),(0.3322),(0.00022),(0.00091),(0.332),(0.332)\end{bmatrix} = 0.00022$$

QP5

Nr = 12

1. MAHASISWA	48181
2. FAKULTAS	15
3. JURUSAN	64
4. PRODI	64
5. JENJANG	5
6. MHS_BIODATA	48181
7. MHS_ORTU	48181
8. KOTA	131
9. NEGARA	2
10. PROPINSI	33
11. PEKERJAAN	51
12. PEKERJAAN	51
TOTAL	**144959**

Tabel 7.15 Klausa WHERE pada *Query* 3 Model 5

	MHS	FAK	JUR	PR	JENJANG	MHS_BIO	MHS_ORTU	KOTA	NEGARA	PROPINSI	PEKERJAAN	PEKERJAAN
MHS	**0**	0	0	1	1	1	1	0	0	0	0	0
FAK	0	**0**	1	0	0	0	0	0	0	0	0	0
JUR	0	1	**0**	1	0	0	0	0	0	0	0	0
PR	1	0	1	**0**	0	0	0	0	0	0	0	0
JENJANG	1	0	0	0	**0**	0	0	0	0	0	0	0
MHS_BIO	1	0	0	0	0	**0**	0	1	0	0	0	0
MHS_ORTU	1	0	0	0	0	0	**0**	0	0	0	1	1
KOTA	0	0	0	0	0	1	0	**0**	0	1	0	0
NEGARA	0	0	0	0	0	0	0	0	**0**	1	0	0
PROPINSI	0	0	0	0	0	0	0	1	1	**0**	0	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	**0**	0
PEKERJAAN	0	0	0	0	0	0	1	0	0	0	0	**0**

$$QPcost = MIN_{i=1}^{12}\left[\sum_{j\equiv 1\ \&\&\ j\neq i}^{12} \frac{Size\ (Rsj)}{\sum_{k=1}^{12} Size\ (Rk)}\right]$$

(i = 1) MAHASISWA

$$= \left[\frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{64}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{5}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{48181}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.00044 + 0.000034 + 0.332 + 0.332]

= 0.664474

(i = 2) FAKULTAS

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{64}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

$= 0.00044$

(i = 3) JURUSAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{15}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{64}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.00011 + 0.00044]

= 0.00055

(i = 4) PRODI

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{64}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right]$$

= [0.332 + 0.00044]

= 0.33344

(i = 5) JENJANG

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right] =$$

= 0.332

(i = 6) MHS_BIO

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 7 \,\&\&\, j \neq i\,(MHS_ORTU)} + \frac{131}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{0}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{0}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right] =$$

= [0.332 + 0.0009] =

= 0 .3329

(i = 7) MHS_ORTU

$$= \left[\frac{48181}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 5 \,\&\&\, j \neq i\,(JENJANG)} + \frac{0}{144959}_{j \equiv 6 \,\&\&\, j \neq i\,(MHS_BIO)} + \frac{0}{144959}_{j \equiv 8 \,\&\&\, j \neq i\,(KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j \equiv 9 \,\&\&\, j \neq i\,(PROPINSI)} + \frac{0}{144959}_{j \equiv 10 \,\&\&\, j \neq i\,(NEGARA)} + \frac{51}{144959}_{j \equiv 11 \,\&\&\, j \neq i\,(PEKERJAAN)} + \frac{51}{144959}_{j \equiv 12 \,\&\&\, j \neq i\,(PEKERJAAN)} \right]$$

= [0.332 + 0.00035 + 0.00035]

= 0.3327

(i = 8) KOTA

$$= \left[\frac{0}{144959}_{j \equiv 1 \,\&\&\, j \neq i\,(MAHASISWA)} + \frac{0}{144959}_{j \equiv 2 \,\&\&\, j \neq i\,(FAKULTAS)} + \frac{0}{144959}_{j \equiv 3 \,\&\&\, j \neq i\,(JURUSAN)} + \frac{0}{144959}_{j \equiv 4 \,\&\&\, j \neq i\,(PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{48181}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \right]$$

$$\left[\frac{33}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.332 + 0.00022]

= 0.33222

(i = 9) PROPINSI

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{131}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{2}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= [0.0009 + 0.00001]

= 0.00091

(i = 10) NEGARA

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{0}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{33}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.00022

(i = 11) PEKERJAAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 12\ \&\&\ j\neq i\ (PEKERJAAN)}\right] =$$

= 0.332

(i = 12) PEKERJAAN

$$= \left[\frac{0}{144959}_{j\equiv 1\ \&\&\ j\neq i\ (MAHASISWA)} + \frac{0}{144959}_{j\equiv 2\ \&\&\ j\neq i\ (FAKULTAS)} + \frac{0}{144959}_{j\equiv 3\ \&\&\ j\neq i\ (JURUSAN)} + \frac{0}{144959}_{j\equiv 4\ \&\&\ j\neq i\ (PRODI)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 5\ \&\&\ j\neq i\ (JENJANG)} + \frac{0}{144959}_{j\equiv 6\ \&\&\ j\neq i\ (MHS_BIO)} + \frac{48181}{144959}_{j\equiv 7\ \&\&\ j\neq i\ (MHS_ORTU)} + \frac{0}{144959}_{j\equiv 8\ \&\&\ j\neq i\ (KOTA)} + \right]$$

$$\left[\frac{0}{144959}_{j\equiv 9\ \&\&\ j\neq i\ (PROPINSI)} + \frac{0}{144959}_{j\equiv 10\ \&\&\ j\neq i\ (NEGARA)} + \frac{0}{144959}_{j\equiv 11\ \&\&\ j\neq i\ (PEKERJAAN)}\right]$$

= 0.332

$$QPcost = MIN_{i=1}^{12}\begin{bmatrix}(0.664),(0.00044),(0.00055),(0.33244),(0.332),(0.3329),\\(0.3327),(0.3322),(0.00022),(0.00091),(0.332),(0.332)\end{bmatrix} = 0.00022$$

LAMPIRAN C PERHITUNGAN *LOCALIZATION PROCESSING COST (LPC)*

C.1 *QUERY* 1

C.1.1 *Relation Processing Cost*

QP 1

Tabel 7.16 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 1

QP	Site Mahasiswa	Site KHS
1	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan4 4. Tabel Jenjang 5. Tabel Prog_Studi	1. Tabel KHS 2. Tabel Nilai 3. Tabel Mata Kuliah

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{145422}$

= 0.331

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{145422}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{145422}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{145422}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5 * \frac{1}{5}}{145422}$

$= 0.0000068$

KHS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{65535 * \frac{65535}{65535}}{145422}$

= 0.451

MATKUL

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{31549 * \frac{31549}{31549}}{145422}$

= 0.217

NILAI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{9*\frac{5}{9}}{145422}$ = 0.00005

QP 2

Tabel 7.17 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 2

QP	Site Mahasiswa	Site KHS
2	1. Tabel Mahasiswa	1. Tabel KHS 2. Tabel Mata Kuliah 3. Tabel Jurusan 4. Tabel Prog_Studi 5. Tabel Nilai 6. Tabel Fakultas 7. Tabel Jenjang

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181*\frac{48181}{48181}}{145422}$

= 0.331

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15*\frac{15}{15}}{145422}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{145422}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{145422}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5*\frac{1}{5}}{145422}$

$= 0.0000068$

KHS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{65535*\frac{65535}{65535}}{145422}$

= 0.451

MATKUL

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{31549*\frac{31549}{31549}}{145422}$

= 0.217

NILAI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{9*\frac{5}{9}}{145422}$

= 0.00005

QP 3

Tabel 7.18 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 3

QP	Site Mahasiswa	Site KHS
3	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan 4. Tabel Jenjang	1. Tabel KHS 2. Tabel Nilai 3. Tabel Mata Kuliah 4. Tabel Prog_Studi

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{145422}$

= 0.331

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{145422}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{145422}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{145422}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5 * \frac{1}{5}}{145422}$

$= 0.0000068$

KHS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{65535 * \frac{65535}{65535}}{145422}$

= 0.451

MATKUL

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{31549 * \frac{31549}{31549}}{145422}$

= 0.217

NILAI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{9 * \frac{5}{9}}{145422}$

= 0.00005

QP 4

Tabel 7.19 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 4

QP	Site Mahasiswa	Site KHS
4	1. Tabel Mahasiswa 2. Tabel Prog_Studi	1. Tabel KHS 2. Tabel Mata Kuliah 3. Tabel Jurusan 4. Tabel Nilai 5. Tabel Fakultas 6. Tabel Jenjang

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{145422}$

= 0.331

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{145422}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{145422}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{145422}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5 * \frac{1}{5}}{145422}$

$= 0.0000068$

KHS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{65535 * \frac{65535}{65535}}{145422}$

= 0.451

MATKUL

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{31549 * \frac{31549}{31549}}{145422}$

= 0.217

NILAI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{9 * \frac{5}{9}}{145422}$

= 0.00005

QP 5

Tabel 7.20 *Site* dan Relasi yang Terlibat pada *Query* 1 Model 5

QP	Site Mahasiswa	Site KHS
5	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan	1. Tabel KHS 2. Tabel Mata Kuliah 3. Tabel Jenjang 4. Tabel Nilai 5. Tabel Prog_Studi

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{145422}$

= 0.331

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{145422}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{145422}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{145422}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5 * \frac{1}{5}}{145422}$

$= 0.0000068$

KHS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{65535 * \frac{65535}{65535}}{145422}$

= 0.451

MATKUL

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{31549 * \frac{31549}{31549}}{145422}$

= 0.217

NILAI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{9 * \frac{5}{9}}{145422}$

= 0.00005

C.1.2 *LPC* untuk *remote site* yang digunakan pada *query plan*

QP1

QP1	**SITE KHS**
	1. KHS
Rs = 3	2. NILAI
	3. MK

RLPC = $Max_{i=1}\ to\ _3[RPC(i)]$

= $Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

= $Max_{i=1}\ to\ _3[0.451;\ 0.217\ ; 0.00005]$

= 0.451

QP2

QP2	**SITE KHS**
Rs = 7	1.KHS 2.NILAI 3.MK 4. JURUSAN 5.PR0DI 6. JENJANG 7. FAKULTAS

$RLPC = Max_{i=1}\ to\ _7[RPC(i)]$

$= Max_{i=1}\ to\ _7[RPC(1), RPC(2), RPC(3), RPC(4), RPC(5), RPC(6), RPC(7)]$

$= Max_{i=1}\ to\ _7[0.451;\ 0.217; 0.00006; 0.0004; 0.0004; 0.00003;\ 0.0001]$

= 0.451

QP3

QP3	**SITE KHS**
Rs = 4	1. KHS 2. NILAI 3. MK 4. PR0DI

$RLPC = Max_{i=1}\ to\ _4[RPC(i)]$

$= Max_{i=1}\ to\ _4[RPC(1), RPC(2), RPC(3), RPC(4)]$

$= Max_{i=1}\ to\ _4[0.451;\ 0.217;\ 0.00006;\ 0.0004]$

= 0.451

QP4

QP4	**SITE KHS**
Rs = 6	1. KHS 2. NILAI 3. MK 4. FAKULTAS 5. JURUSAN 6. JENJANG

$RLPC = Max_{i=1}\ to\ _6[RPC(i)]$

$= Max_{i=1}\ to\ _6[RPC(1), RPC(2), RPC(3), RPC(4), RPC(5), RPC(6), RPC(7)]$

$= Max_{i=1}\ to\ _6[0.451;\ 0.217;\ 0.00006;\ 0.0004;\ 0.00003;\ 0.0001]$

= 0.451

QP5

QP5	**SITE KHS**
	1. KHS
	2. NILAI
Rs = 5	3. MK
	4. PR0DI
	5. JENJANG

RLPC = $Max_{i=1}\ to\ _5[RPC(i)]$

$= Max_{i=1}\ to\ _5[RPC(1), RPC(2), RPC(3), RPC(4), RPC(5)]$

$= Max_{i=1}\ to\ _5[0.451;\ 0.217;\ 0.00006;\ 0.0004;\ 0.00003]$

= 0.451

C.1.3 *LPC* untuk *control site* yang digunakan pada *query plan*

QP1

QP1	**SITE MAHASISWA**	
	1. MAHASISWA	48181
	2. FAKULTAS	15
Nr = 5	3. JENJANG	5
	4. JURUSAN	64
	5. PR0DI	64
	T0TAL	**48329**

CLPC=$Max_{i,j-1\ to\ 5}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1))/\sum_{k=1}^{5} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181*48181}}{48329}$

= 0.00002

i,j = 2 [FAKULTAS]

$= \frac{15 * \frac{15}{15*15}}{48329}$

= 0.00002

i,j = 3 [JENJANG]

$= \frac{5 * \frac{5}{5*5}}{48329}$

= 0.00002

i,j = 4 [JURUSAN]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

i,j = 5 [PR0DI]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

Max = 0.00002

QP2

QP2	**SITE MAHASISWA**	
Nr = 1	1. MAHASISWA	48181
	T0TAL	**48181**

$CLPC = Max_{i,j-1\ to\ 1}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{1} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181 * 48181}}{48181}$

= 0.00002

Max = 0.00002

QP3

QP3	**SITE MAHASISWA**	
Nr = 4	1. MAHASISWA	48181
	2. JURUSAN	64
	3. FAKULTAS	15
	4. JENJANG	5
	T0TAL	**48265**

$CLPC = Max_{i,j-1\ to\ 4}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{4} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181 * 48181}}{48265}$

= 0.00002

i,j = 2 [FAKULTAS]

$= \frac{15 * \frac{15}{15 * 15}}{48265}$

= 0.00002

i,j = 3 [JENJANG]

$= \frac{5 * \frac{5}{5 * 5}}{48266}$

= 0.00002

i,j = 4 [JURUSAN]

$= \frac{64 * \frac{64}{64 * 64}}{48266}$

= 0.00002

Max = 0.00002

QP4

QP4	**SITE MAHASISWA**	
Nr = 2	1. MAHASISWA	48181
	2. PR0DI	64
	T0TAL	**48245**

$CLPC = Max_{i,j-1\ to\ 2}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{2} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181 * 48181}}{48245}$

= 0.00002

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

i,j = 2 [PR0DI]

$= \frac{64*\frac{64}{64*64}}{48245}$

= 0.00002

Max = 0.00002

QP5

QP5	**SITE MAHASISWA**	
	1. MAHASISWA	48181
Nr = 3	2. FAKULTAS	15
	3. JURUSAN	64
	T0TAL	**48260**

$CLPC = Max_{i,j-1\ to\ 3}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1))/\sum_{k=1}^{3} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181*\frac{48181}{48181*48181}}{48260}$

= 0.00002

i,j = 2 [FAKULTAS]

$= \frac{15*\frac{15}{15*15}}{48260}$

= 0.00002

i,j = 3 [JURUSAN]

$= \frac{64*\frac{64}{64*64}}{48260}$

= 0.00002

Max = 0.00002

C.1.4 *Local Processing Cost (LPC)*

QP1

$LPC = \sum_{k=1}^{2-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 5\&\&} R(i)CS\&JOIN[RPC(i)]$

= 0.451 + (0.0002 – Max(0.331 ; 0.0001 ; 0.0004 ; 0.00003 ; 0.0004))

= 0.451 + (0.0002 – 0.331)

= 0.451 + (-0.3308)

= 0.1202

QP2

$LPC = \sum_{k=1}^{2-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 1\&\&} R(i)CS\&JOIN[RPC(i)]$

= 0.451 + (0.0002– Max(0.331))

= 0.451 + (0.0002 – 0.331)

= 0.451 + (-0.3308)

= 0.1202

QP3

LPC $= \sum_{k=1}^{2-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 4\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.451 + (0.249 – Max(0.331 ; 0.0001 ; 0.0004 ; 0.00003))

= 0.451 + (0.0002 – 0.331)

= 0.451 + (-0.3308)

= 0.1202

QP4

LPC $= \sum_{k=1}^{2-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 5\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.451 + (0.499 – Max(0.331 ; 0.0004))

= 0.451 + (0.0002 – 0.331)

= 0.451 + (-0.3308)

= 0.1202

QP5

LPC $= \sum_{k=1}^{2-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 5\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.451 + (0.332 – Max(0.331 ; 0.0001 ; 0.0004))

= 0.451 + (0.0002 – 0.331)

= 0.451 + (-0.3308)

= 0.1202

C.2 *QUERY* 2

C.2.1 *Relation Processing Cost*

QP1

Tabel 7.21 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 1

QP	Site Mahasiswa	Site Keuangan	Site KHS
1	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan 4. Tabel Jenjang 5. Tabel Prog_Studi	1. Tabel Keuangan 2. Tabel Item 3. Tabel Transaksi	

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144696}$

= 0.333

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{144696}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144696}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144696}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5 * \frac{1}{5}}{144696}$

= 0.0000069

KEUANGAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{96362 * \frac{96362}{96362}}{144696}$

= 0.666

ITEM

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{3 * \frac{3}{3}}{144696}$

= 0.00002

TRANSAKSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2 * \frac{2}{2}}{144696}$

= 0.00001

QP 2

Tabel 7.22 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 2

QP	Site Mahasiswa	Site Keuangan	Site KHS
2	1. Tabel Mahasiswa	1. Tabel Keuangan 2. Tabel Item 3. Tabel Transaksi	1. Tabel Fakultas 2. Tabel Jurusan 3. Tabel Jenjang 4. Tabel Prog_Studi

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144696}$

= 0.333

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{144696}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{144696}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{144696}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5*\frac{1}{5}}{144696}$

= 0.0000069

KEUANGAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{96362*\frac{96362}{96362}}{144696}$

= 0.666

ITEM

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{3*\frac{3}{3}}{144696}$

= 0.00002

TRANSAKSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2*\frac{2}{2}}{144696}$

= 0.00001

QP 3

Tabel 7.23 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 3

QP	Site Mahasiswa	Site Keuangan	Site KHS
3	1. Tabel Mahasiswa 2. Tabel Prog_Studi	1. Tabel Keuangan 2. Tabel Item 3. Tabel Transaksi	1. Tabel Fakultas 2. Tabel Jurusan 3. Tabel Jenjang

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181*\frac{48181}{48181}}{144696}$

= 0.333

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15*\frac{15}{15}}{144696}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{144696}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{144696}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5*\frac{1}{5}}{144696}$

= 0.0000069

KEUANGAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{96362 * \frac{96362}{96362}}{144696}$

= 0.666

ITEM

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{3 * \frac{3}{3}}{144696}$

= 0.00002

TRANSAKSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2 * \frac{2}{2}}{144696}$

= 0.00001

QP 4

Tabel 7.24 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 4

QP	Site Mahasiswa	Site Keuangan	Site KHS
4	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan 4. Tabel Jenjang	1. Tabel Keuangan 2. Tabel Item 3. Tabel Transaksi	1. Tabel Prog_Studi

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144696}$

= 0.333

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{144696}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144696}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144696}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5 * \frac{1}{5}}{144696}$

= 0.0000069

KEUANGAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{96362 * \frac{96362}{96362}}{144696}$

= 0.666

ITEM

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{3 * \frac{3}{3}}{144696}$

= 0.00002

TRANSAKSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2*\frac{2}{2}}{144696}$

= 0.00001

QP 5

Tabel 7.25 *Site* dan Relasi yang Terlibat pada *Query* 2 Model 5

QP	Site Mahasiswa	Site Keuangan	Site KHS
5	1. Tabel Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan 4. Tabel Jenjang 5. Tabel Prog_Studi	1. Tabel Keuangan 2. Tabel Item 3. Tabel Transaksi 4. Tabel Mahasiswa	

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181*\frac{48181}{48181}}{192877}$

= 0.249

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15*\frac{15}{15}}{192877}$

= 0.00008

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{192877}$

= 0.00033

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{192877}$

= 0.00033

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5*\frac{1}{5}}{192877}$

= 0.000005

KEUANGAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{96362*\frac{96362}{96362}}{192877}$

= 0.499

ITEM

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{3*\frac{3}{3}}{192877}$

= 0.00001

TRANSAKSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2 * \frac{2}{2}}{192877}$

= 0.00001

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{192877}$

= 0.249

C.2.2 *LPC* untuk *remote site* yang digunakan pada *query plan*

QP1

QP1	**SITE KEUANGAN**
Rs = 3	1. KEUANGAN 2. ITEM 3. TRANSAKSI

RLPC = $Max_{i=1}\ to\ _3[RPC(i)]$

= $Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

= $Max_{i=1}\ to\ _3[0.666;\ 0.00002;\ 0.00001]$

= 0.666

QP2

QP2	**SITE KEUANGAN**	**SITE KHS**
Rs = 3, Rs = 4	1.KHS 2.NILAI 3.MK	1. JURUSAN 2.PR0DI 3. JENJANG 4. FAKULTAS

RLPC = $Max_{i=1}\ to\ _3[RPC(i)]$

= $Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

= $Max_{i=1}\ to\ _3[0, .666; 0.00002;\ 0.00001]$

= 0.666

RLPC = $Max_{i=1}\ to\ _4[RPC(i)]$

= $Max_{i=1}\ to\ _4[RPC(1), RPC(2), RPC(3), RPC(4)]$

= $Max_{i=1}\ to\ _4[\ 0.0001;\ 0.0002;\ 0.0004;\ 0.00003]$

= 0.0004

QP3

QP3	**SITE KEUANGAN**	**SITE KHS**
Rs = 3, Rs = 3	1. KEUANGAN 2. ITEM	1. FAKULTAS 2. JURUSAN

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

3. TRANSAKSI 3. JENJANG

$RLPC = Max_{i=1}\ to\ _3[RPC(i)]$

$= Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

$= Max_{i=1}\ to\ _3[0.666;\ 0.00002;\ 0.00001]$

= 0.666

$RLPC = Max_{i=1}\ to\ _3[RPC(i)]$

$= Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

$= Max_{i=1}\ to\ _3[0.0001;\ 0.0002; 0.00003]$

= 0.0002

QP4

QP4	**SITE KEUANGAN**	**SITE KHS**
	1. KEUANGAN	1. PR0DI
Rs = 3 , Rs = 1	2. ITEM	
	3. TRANSAKSI	

$RLPC = Max_{i=1}\ to\ _3[RPC(i)]$

$= Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

$= Max_{i=1}\ to\ _3[0.666;\ 0.00002;\ 0.00001]$

= 0.666

$RLPC = Max_{i=1}\ to\ _1[RPC(i)]$

$= Max_{i=1}\ to\ _1[RPC(1)]$

$= Max_{i=1}\ to\ _1[0.0004]$

= 0.0004

QP5

QP5	**SITE KEUANGAN**
	1. KEUANGAN
	2. ITEM
Rs = 4	3. TRANSAKSI
	4. MAHASISWA

$RLPC = Max_{i=1}\ to\ _4[RPC(i)]$

$= Max_{i=1}\ to\ _4[RPC(1), RPC(2), RPC(3), RPC(4)]$

$= Max_{i=1}\ to\ _4[\ 0.499;\ 0.00001;\ 0.00001;\ 0.249]$

= 0.499

C.2.3 *LPC* untuk *control site* yang digunakan pada *query plan*

QP1

QP1	**SITE MAHASISWA**	
	1. MAHASISWA	48181
	2. FAKULTAS	15
Nr = 5	3. JENJANG	5
	4. JURUSAN	64
	5. PR0DI	64
	T0TAL	**48329**

$CLPC = Max_{i,j-1\ to\ 5}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{5} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181*48181}}{48329}$

= 0.00002

i,j = 2 [FAKULTAS]

$= \frac{15 * \frac{15}{15*15}}{48329}$

= 0.00002

i,j = 3 [JENJANG]

$= \frac{5 * \frac{5}{5*5}}{48329}$

= 0.00002

i,j = 4 [JURUSAN]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

i,j = 5 [PR0DI]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

Max = 0.00002

QP2

QP2	**SITE MAHASISWA**	
Nr = 1	1. MAHASISWA	48181
	T0TAL	**48181**

$CLPC = Max_{i,j-1\ to\ 1}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{1} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181*48181}}{48181}$

= 0.00002

Max = 0.00002

QP3

QP3	**SITE MAHASISWA**	
Nr = 2	1. MAHASISWA	48181
	2. PR0DI	64
	T0TAL	**48245**

$CLPC = Max_{i,j-1\ to\ 2}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{2} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181 * 48181}}{48245}$

= 0.00002

i,j = 2 [PR0DI]

$= \frac{64 * \frac{64}{64 * 64}}{48245}$

= 0.00002

Max = 0.00002

QP4

QP4	**SITE MAHASISWA**	
Nr = 4	1. MAHASISWA	48181
	2. JURUSAN	64
	3. FAKULTAS	15
	4. JENJANG	5
	T0TAL	**48265**

$CLPC = Max_{i,j-1\ to\ 4}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{4} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181 * 48181}}{48265}$

= 0.00002

i,j = 2 [FAKULTAS]

$= \frac{15 * \frac{15}{15 * 15}}{48265}$

= 0.00002

Max = 0.00002

QP5

QP5	**SITE MAHASISWA**	
Nr = 5	1. MAHASISWA	48181
	2. FAKULTAS	15

3. JENJANG	5
4. JURUSAN	64
5. PR0DI	64
T0TAL	**48329**

$CLPC = Max_{i,j-1\ to\ 5}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1))/\sum_{k=1}^{5} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181*48181}}{48329}$

= 0.00002

i,j = 2 [FAKULTAS]

$= \frac{15 * \frac{15}{15*15}}{48329}$

= 0.00002

i,j = 3 [JENJANG]

$= \frac{5 * \frac{5}{5*5}}{48329}$

= 0.00002

i,j = 4 [JURUSAN]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

i,j = 5 [PR0DI]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

Max = 0.00002

C.2.4 *Local Processing Cost (LPC)*

QP1

$LPC = \sum_{k=1}^{2-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 5\&\&} R(i)CS\&JOIN[RPC(i)]$

= 0.666 + (0.0002 – Max(0.333 ; 0.0001 ; 0.0004 ; 0.00003 ; 0.0004))

= 0.666 + (0.0002 – 0.333)

= 0.666 + (-0.3328)

= 0.3332

QP2

$LPC = \sum_{k=1}^{3-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 1\&\&} R(i)CS\&JOIN[RPC(i)]$

= 0.0004 + 0.0002 + (0.0002– Max(0.333))

= 0.6664 + (0.0002– 0.333)

= 0.6664 + (-0.3328)

= 0.3336

QP3

$LPC \quad = \sum_{k=1}^{3-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 2\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.0002 + 0.666 + (0.0002– Max(0.333; 0.0004))

= 0.6662 + (0.0002– 0.333)

= 0.6662 + (-0.3328)

= 0.3336

QP4

$LPC \quad = \sum_{k=1}^{3-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 4\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.0004 + 0.666 + (0.0002– Max(0.333; 0.0001 ; 0.0004 ; 0.00003))

= 0.6664 + (0.0002– 0.333)

= 0.6664+ (-0.3328)

= 0.3336

QP5

$LPC \quad = \sum_{k=1}^{2-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 5\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.499 + (0.0002 – Max(0.249 ; 0.00008 ; 0.00033 ; 0.000005 ; 0.00033))

= 0.499 + (0.0002 – 0.249)

= 0.499 + (-0.0818)

= 0.4172

C.3 *QUERY* 3

C.3.1 *Relation Processing Cost*

QP 1

Tabel 7.26 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 1

QP	Site Mahasiswa	Site Mahasiswa Biodata	Site Mahasiswa Ortu	Site KHS
1	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan 4. Tabel Jenjang 5. Tabel Prog_Studi	1. Tabel Mhs_Biodata	1.Tabel Mhs_ortu 4. Tabel Kota 2.Tabel Negara 5. Tabel Propinsi 3.Tabel Pekerjaan 6. Tabel Pekerjaan	

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15*\frac{15}{15}}{144859}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{144859}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{144859}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5*\frac{1}{5}}{144859}$

= 0.0000069

MHS_BIODATA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181*\frac{48181}{48181}}{144859}$

= 0.332

MHS_ORTU

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181*\frac{48181}{48181}}{144859}$

= 0.332

KOTA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{151*\frac{151}{151}}{144859}$

= 0.0009

PROPINSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{33*\frac{33}{33}}{144859}$

= 0.0002

NEGARA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2*\frac{2}{2}}{144859}$

= 0.0000069

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51*\frac{51}{51}}{144859}$

= 0.00035

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51*\frac{51}{51}}{144859}$

= 0.00035

QP 2

Tabel 7.27 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 2

QP	Site Mahasiswa	Site Mahasiswa Biodata	Site Mahasiswa Ortu	Site KHS
2	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan 4. Tabel Jenjang	1. Tabel Mhs_Biodata 2. Tabel Negara 3. Tabel Kota 4. Tabel Propinsi	1.Tabel Mhs_ortu 2.Tabel Pekerjaan 3.Tabel Pekerjaan	

	5. Tabel Prog_Studi			

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

MHS_ORTU

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{144859}$

= 0.0001

KOTA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{151 * \frac{151}{151}}{144859}$

= 0.0009

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144859}$

= 0.0004

PROPINSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{33 * \frac{33}{33}}{144859}$

= 0.0002

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144859}$

= 0.0004

NEGARA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2 * \frac{2}{2}}{144859}$

= 0.0000069

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5 * \frac{1}{5}}{144859}$

= 0.0000069

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51 * \frac{51}{51}}{144859}$

= 0.00035

MHS_BIODATA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51 * \frac{51}{51}}{144859}$

= 0.00035

QP 3

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 7.28 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 3

QP	Site Mahasiswa	Site Mahasiswa Biodata	Site Mahasiswa Ortu	Site KHS
3	1. Tabel Mahasiswa	1 Tabel Mhs_Biodata 2 Tabel Negara 3 Tabel Kota 4 Tabel Propinsi	1. Tabel Mhs_ortu 2. Tabel Pekerjaan 3. Tabel Pekerjaan	1. Tabel Fakultas 2. Tabel Jurusan 3. Tabel Jenjang 4. Tabel Jenjang 5. Tabel Prog_Studi

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{144859}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144859}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144859}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5 * \frac{1}{5}}{144859}$

= 0.0000069

MHS_BIODATA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

MHS_ORTU

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

KOTA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{151 * \frac{151}{151}}{144859}$

= 0.0009

PROPINSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{33 * \frac{33}{33}}{144859}$

= 0.0002

NEGARA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2 * \frac{2}{2}}{144859}$

= 0.0000069

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51 * \frac{51}{51}}{144859}$

= 0.00035

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51 * \frac{51}{51}}{144859}$

= 0.00035

QP 4

Tabel 7.29 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 4

QP	Site Mahasiswa	Site Mahasiswa Biodata	Site Mahasiswa Ortu	Site KHS
4	1.Tabel Mahasiswa 2.Tabel Prog_Studi	1 Tabel Mhs_Biodata 2 Tabel Kota 3 Tabel Propinsi	1. Tabel Mhs_ortu 2. Tabel Pekerjaan 3. Tabel Pekerjaan 4. Tabel Negara	1 Tabel Fakultas 2 Tabel Jurusan 3 Tabel Jenjang

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15 * \frac{15}{15}}{144859}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144859}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64 * \frac{64}{64}}{144859}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5 * \frac{1}{5}}{144859}$

= 0.0000069

MHS_BIODATA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

MHS_ORTU

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181 * \frac{48181}{48181}}{144859}$

= 0.332

KOTA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{151 * \frac{151}{151}}{144859}$

= 0.0009

PROPINSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{33 * \frac{33}{33}}{144859}$

= 0.0002

NEGARA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2*\frac{2}{2}}{144859}$

= 0.0000069

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51*\frac{51}{51}}{144859}$

= 0.00035

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51*\frac{51}{51}}{144859}$

= 0.00035

QP 5

Tabel 7.30 *Site* dan Relasi yang Terlibat pada *Query* 3 Model 5

QP	Site Mahasiswa	Site Mahasiswa Biodata	Site Mahasiswa Ortu	Site KHS
5	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan 4. Tabel Jenjang 5. Tabel Prog_Studi	1 Tabel Mhs_Biodata 2 Tabel Kota 3 Tabel Propinsi	1. Tabel Mhs_ortu 2. Tabel Pekerjaan 3. Tabel Pekerjaan 4. Tabel Negara	

MAHASISWA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181*\frac{48181}{48181}}{144859}$

= 0.332

FAKULTAS

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{15*\frac{15}{15}}{144859}$

= 0.0001

JURUSAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{144859}$

= 0.0004

PRODI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{64*\frac{64}{64}}{144859}$

= 0.0004

JENJANG

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{5*\frac{1}{5}}{144859}$

= 0.0000069

MHS_BIODATA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181*\frac{48181}{48181}}{144859}$

= 0.332

MHS_ORTU

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{48181*\frac{48181}{48181}}{144859}$

= 0.332

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

KOTA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{151 * \frac{151}{151}}{144859}$

= 0.0009

PROPINSI

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{33 * \frac{33}{33}}{144859}$

= 0.0002

NEGARA

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{2 * \frac{2}{2}}{144859}$

= 0.0000069

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51 * \frac{51}{51}}{144859}$

= 0.00035

PEKERJAAN

RPC = $Nt * Sr / \sum_{K=1}^{8} Nt\ (k)$

$= \frac{51 * \frac{51}{51}}{144859}$

= 0.00035

C.3.2 *LPC* untuk *remote site* yang digunakan pada *query plan*

QP1

QP1	**SITE MHS_BI0**	**SITE MHS_0RTU**
	1. MHS_BI0DATA	1. MHS_0RTU
		2. K0TA
Rs = 1, Rs = 6		3. NEGARA
		4. PR0PINSI
		5. PEKERJAAN
		6. PEKERJAAN

RLPC = $Max_{i=1}\ to\ _{1}[RPC(i)]$

= $Max_{i=1}\ to\ _{1}[RPC(1), RPC(2), RPC(3)]$

= $Max_{i=1}\ to\ _{1}[0.332]$

= 0.332

RLPC = $Max_{i=1}\ to\ _{6}[RPC(i)]$

= $Max_{i=1}\ 6[RPC(1), RPC(2), RPC(3)]$

= $Max_{i=1}\ to\ _{6}[\ 0.332;\ 0.0009;\ 0.0002;\ 0.00001;\ 0.00035;\ 0.00035]$

= 0.332

QP2

QP2	**SITE MHS_BI0**	**SITE MHS_0RTU**
	1. MHS_BI0DATA	1. MHS_0RTU
	2. K0TA	2. PEKERJAAN
Rs = 4, Rs = 3	3. NEGARA	3. PEKERJAAN
	4. PR0PINSI	

$RLPC = Max_{i=1}\ to\ _4[RPC(i)]$

$= Max_{i=1}\ to\ _4[RPC(1), RPC(2), RPC(3), RPC(4)]]$

$= Max_{i=1}\ to\ _4[0.332;\ 0.0009;\ 0.0002;\ 0.00001]$

= 0.332

$RLPC = Max_{i=1}\ to\ _3[RPC(i)]$

$= Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

$= Max_{i=1}\ to\ _3[0.332; 0.00035;\ 0.00035]$

= 0.332

QP3

QP3	SITE MHS_BI0	SITE MHS_0RTU	SITE KHS
Rs = 3, Rs = 3, Rs = 4	1. MHS_BI0DATA 2. K0TA 3. PR0PINSI 4. NEGARA	1. MHS_0RTU 2. PEKERJAAN 3. PEKERJAAN	1. FAKULTAS 2. JURUSAN 3. JENJANG 4. PR0DI

$RLPC = Max_{i=1}\ to\ _4[RPC(i)]$

$= Max_{i=1}\ to\ _4[RPC(1), RPC(2), RPC(3), RPC(4)]$

$= Max_{i=1}\ to\ _4\ [0.332; 0.0009;\ 0.0002;\ 0.00001]$

= 0.332

$RLPC = Max_{i=1}\ to\ _3[RPC(i)]$

$= Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

$= Max_{i=1}\ to\ _3\ [0.332;\ 0.00035;\ 0.00035]$

= 0.332

$RLPC = Max_{i=1}\ to\ _4[RPC(i)]$

$= Max_{i=1}\ to\ _4[RPC(1), RPC(2), RPC(3), RPC(4)]$

$= Max_{i=1}\ to\ _4[0.0001;\ 0.0004;\ 0.00003;\ 0.0004]$

= 0.0004

QP4

QP4	SITE MHS_BI0	SITE MHS_0RTU	SITE KHS
Rs = 3 , Rs = 4, Rs = 3	1. MHS_BI0DATA 2. K0TA 3. PR0PINSI	1. MHS_0RTU 2. PEKERJAAN 3. PEKERJAAN 4. NEGARA	1. JURUSAN 2. FAKULTAS 3. JENJANG

$RLPC = Max_{i=1}\ to\ _4[RPC(i)]$

$= Max_{i=1}\ to\ _4[RPC(1), RPC(2), RPC(3)]$

$= Max_{i=1}\ to\ _4\ [0.332; 0.0009;\ 0.0002]$

= 0.332

RLPC = $Max_{i=1}\ to\ _4[RPC(i)]$

$= Max_{i=1}\ to\ _4[RPC(1), RPC(2), RPC(3), RPC(4)]$

$= Max_{i=1}\ to\ _4\ [0.332;\ 0.00035;\ 0.00035;\ 0.00001]$

= 0.332

RLPC = $Max_{i=1}\ to\ _3[RPC(i)]$

$= Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

$= Max_{i=1}\ to\ _3[0.0001;\ 0.0004;\ 0.00003]$

= 0.0004

QP5

QP5	**SITE KEUANGAN**	**SITE MHS_0RTU**
	1. MHS_BI0DATA	1. MHS_0RTU
	2. K0TA	2. PEKERJAAN
Rs = 3, Rs = 4	3. PR0PINSI	3. PEKERJAAN
		4. NEGARA

RLPC = $Max_{i=1}\ to\ _3[RPC(i)]$

$= Max_{i=1}\ to\ _3[RPC(1), RPC(2), RPC(3)]$

$= Max_{i=1}\ to\ _3[0.332;\ 0.0009;\ 0.0002]$

= 0.332

RLPC = $Max_{i=1}\ to\ _4[RPC(i)]$

$= Max_{i=1}\ to\ _4[RPC(1), RPC(2), RPC(3),, RPC(4)]$

$= Max_{i=1}\ to\ _4[0.332;\ 0.00035;\ 0.00035;\ 0.00001]$

= 0.332

C.3.3 *LPC* untuk *control site* yang digunakan pada *query plan*

QP1

QP1	**SITE MAHASISWA**	
	1. MAHASISWA	48181
	2. FAKULTAS	15
Nr = 5	3. JENJANG	5
	4. JURUSAN	64
	5. PR0DI	64
	T0TAL	**48329**

CLPC=$Max_{i,j-1\ to\ 5}[Nt(JOIN(R1, R1)) * Sj(JOIN(R1, R1)) / \sum_{k=1}^{5} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181*48181}}{48329}$

= 0.00002

i,j = 2 [FAKULTAS]

$= \frac{15 * \frac{15}{15*15}}{48329}$

= 0.00002

i,j = 3 [JENJANG]

$= \frac{5 * \frac{5}{5*5}}{48329}$

= 0.00002

i,j = 4 [JURUSAN]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

i,j = 5 [PR0DI]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

Max = 0.00002

QP2

QP2	**SITE MAHASISWA**	
	1. MAHASISWA	48181
	2. FAKULTAS	15
Nr = 5	3. JENJANG	5
	4. JURUSAN	64
	5. PR0DI	64
	T0TAL	**48329**

$CLPC = Max_{i,j-1\ to\ 5}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{5} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181*48181}}{48329}$

= 0.00002

i,j = 2 [FAKULTAS]

$= \frac{15 * \frac{15}{15*15}}{48329}$

= 0.00002

i,j = 3 [JENJANG]

$= \frac{5 * \frac{5}{5*5}}{48329}$

= 0.00002

i,j = 4 [JURUSAN]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

i,j = 5 [PR0DI]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

Max = 0.00002

QP3

QP3	**SITE MAHASISWA**	
Nr = 1	1. MAHASISWA	48181
	T0TAL	**48181**

$CLPC = Max_{i,j-1\ to\ 1}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{1} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181 * 48181}}{48181}$

= 0.00002

Max = 0.00002

QP4

QP4	**SITE MAHASISWA**	
Nr = 2	1. MAHASISWA	48181
	2. PR0DI	64
	T0TAL	**48245**

$CLPC = Max_{i,j-1\ to\ 2}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{2} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181 * 48181}}{48245}$

= 0.00002

i,j = 2 [PR0DI]

$= \frac{64 * \frac{64}{64 * 64}}{48245}$

= 0.00002

Max = 0.00002

QP5

QP5	**SITE MAHASISWA**	
	1. MAHASISWA	48181
	2. FAKULTAS	15
Nr = 5	3. JENJANG	5
	4. JURUSAN	64
	5. PR0DI	64
	T0TAL	**48329**

$CLPC = Max_{i,j-1\ to\ 5}[Nt(JOIN(R1,R1)) * Sj(JOIN(R1,R1)) / \sum_{k=1}^{5} Nt(k)]$

i,j = 1 [MAHASISWA]

$= \frac{48181 * \frac{48181}{48181*48181}}{48329}$

= 0.00002

i,j = 2 [FAKULTAS]

$= \frac{15 * \frac{15}{15*15}}{48329}$

= 0.00002

i,j = 3 [JENJANG]

$= \frac{5 * \frac{5}{5*5}}{48329}$

= 0.00002

i,j = 4 [JURUSAN]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

i,j = 5 [PR0DI]

$= \frac{64 * \frac{64}{64*64}}{48329}$

= 0.00002

Max = 0.00002

C.3.4 *Local Processing Cost (LPC)*

QP1

LPC $= \sum_{k=1}^{3-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 5\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.332 + 0.332 + (0.0002 – Max(0.332 ; 0.0001 ; 0.0004 ; 0.00003 ; 0.0004))

= 0.332 + 0.332 + (0.0002– 0.333)

= 0.664 + (-0.3328)

= 0.3322

QP2

LPC $= \sum_{k=1}^{3-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 5\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.332 + 0.332 + (0.0002 – Max(0.332 ; 0.0001 ; 0.0004 ; 0.00003 ; 0.0004))

= 0.332 + 0.332 + (0.0002 – 0.333)

= 0.664 + (-0.3318)

= 0.3322

QP3

LPC $= \sum_{k=1}^{4-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ 1\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.332 + 0.332 + 0.0004 + (0.0002– Max(0.332))

= 0.6644 + (0.0002 – 0.332)

= 0.6644 + (-0.3318)

= 0.3326

QP4

$LPC \quad = \sum_{k=1}^{3-1}\left(RLPC(k)\right) + (CLPC - \ Max_{i=1\ to\ 2\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.332 + 0.332 + 0.0004 + (0.0002– Max(0.332; 0.0001 ; 0.0004 ; 0.00003))

= 0.6644 + (0.0002 – 0.332)

= 0.6644+ (0.3318)

= 0.3326

QP5

$LPC \quad = \sum_{k=1}^{3-1}\left(RLPC(k)\right) + (CLPC - \ Max_{i=1\ to\ 5\&\&}R(i)CS\&JOIN[RPC(i)]$

= 0.332 + 0.332 + (0.0002 – Max(0.332 ; 0.0001 ; 0.0004 ; 0.00003 ; 0.0004))

= 0.332 + 0.332 + (0.0002 – 0.333)

= 0.664 + (-0.3318)

= 0.3322

LAMPIRAN D PERHITUNGAN TLBO

D.1 *QUERY* 1

Tabel 7.31 Tabel Nilai *Fitness* dan Nilai *Mean Query Plan*

QP	QAC	QLC	LPC	FITNESS
1	0.468	0.00044	0.1202	0.196
2	0.218	0.00044	0.1202	0.112
3	0.5	0.00044	0.1202	0.206
4	0.375	0.00044	0.1202	0.165
5	0.468	0.00044	0.1202	0.196
MEAN	**0.405**	**0.00044**	**0.1202**	

Tf = rand [1 + rand(0,1){2 - 1}]

Tf = 1

r1 = 0.462433

r2 = 0.656533

r3 = 0.326031

diff mean QAC = r1 (XQAC – Tf. MEANQAC)

= 0.462433 (0.218 – 1 (0.405))

= 0.462433 (-0.187)

= - 0.0865

diff mean QLC = r1 (XQLC – Tf. MEANQLC)

= 0.656533 (0.00044 – 1 (0.00044))

= 0.656533 (0)

= 0

diff mean LPC = r1 (XLPC – Tf. MEANLPC)

= 0.326031 (0.1202 – 1 (0.1202))

= 0.326031 (0)

= 0

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

D.1.1 TEACHER PHASE

X = Query Plan (QP)

QACnew

X1 = X1old + diff(QAC)

= 0.468 + (- 0.0865)

= 0.3815

X2 = X2old + diff(QAC)

= 0.218 + (- 0.0865)

= 0.1315

X3 = X3old + diff(QAC)

= 0.5 + (- 0.0865)

= 0.4135

X4 = X4old + diff(QAC)

= 0.375 + (- 0.0865)

= 0.2885

X5 = X5old + diff(QAC)

= 0.468 + (- 0.0865)

= 0.3815

QLCnew

Difference QLC = 0

LPCnew

Difference LPC = 0

Generation 01

Tabel 7.32 Tabel *Update Population* 01 berdasar *Tf*

QP	QAC	QLC	LPC
1	0.3815	0.00044	0.1202
2	0.1315	0.00044	0.1202
3	0.4135	0.00044	0.1202
4	0.2885	0.00044	0.1202
5	0.3815	0.00044	0.1202

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

D.1.2 LEARNER PHASE

Generation 01

Tabel 7.33 Tabel Nilai *Fitness Population* 01

QP	QAC	QLC	LPC	FITNESS
1	0.3815	0.00044	0.1202	0.167
2	0.1315	0.00044	0.1202	0.084
3	0.4135	0.00044	0.1202	0.178
4	0.2885	0.00044	0.1202	0.136
5	0.3815	0.00044	0.1202	0.167

Interaksi antar *learner*, jadi diasumsikan interaksi antara QP1 dengan QP2, QP3, QP4, QP5 dst. Dari hasil interaksi diambil yang terendah.

r1 = 0.647798

r2 = 0.43764

r3 = 0.232127

QAC

1 dan 2

X1 > X2

X1new = X1old + r1(X2 – X1)

= 0.3815 + 0.647798 (0.1315 – 0.3815)

= 0.3815 + (-0.1619)

= 0.2196

1 dan 3

X1 < X3

X3new = X3old + r1(X1 – X3)

= 0.4135 + 0.647798 (0.3815 – 0.4135)

= 0.4135 + (-0.0207)

= 0.3928

1 dan 4

X1 > X4

X1new = X1old + r1(X4 – X1)

= 0.3815 + 0.647798 (0.2885 – 0.3815)

= 0.3815 + (-0.0602)

= 0.3213

1 dan 5

X1 < X5

X5new = X5old + r1(X1 – X5)

= 0.3815 + 0.647798 (0.3815 – 0.3815)

= 0.3815 + (0)

= 0.3815

2 dan 3

X2 < X3

X3new = X3old + r1(X2 – X3)

= 0.4135 + 0.647798 (0.1315 – 0.4135)

= 0.4135 + (-0.1826)

= 0.2309

2 dan 4

X2 < X4

X4new = X4old + r1(X2 – X4)

= 0.2885 + 0.647798 (0.1315 – 0.2885)

= 0.2885 + (-0.1017)

= 0.1868

2 dan 5

X2 < X5

X5new = X5old + r1(X2 – X5)

= 0.3815 + 0.647798 (0.1315 – 0.3815)

= 0.3815 + (-0.1619)

= 0.2196

3 dan 4

X3 > X4

X3new = X3old + r1(X4 – X3)

= 0.4135+ 0.647798 (0.2885 – 0.4135)

= 0.4135 + (-0.0809)

= 0.3326

3 dan 5

X3 > X5

X3new = X3old + r1(X5 – X3)

= 0.4135 + 0.647798 (0.3815 – 0.4135)

= 0.4135 + (-0.0207)

= 0.3928

4 dan 5

X4 < X5

X5new = X5old + r1(X4 – X5)

= 0.3815 + 0.647798 (0.2885 – 0.3815)

= 0.3815 + (-0.0602)

= 0.3215

QLC

1 dan 2

X1 > X2

X1new = X1old + r2(X2 – X1)

= 0.00044+ 0.43764 (0.00044 – 0.00044)

= 0.00044 + (0)

= 0.00044

LPC

1 dan 2

X1 > X2

X1new = X1old + r2(X2 – X1)

= 0.1202+ 0. 232127 (0.1202 – 0.1202)

= 0.1202 + (0)

= 0.1202

Tabel 7.34 Tabel *Update Population* 02 berdasar Interaksi *Learner*

QP	QAC	QLC	LPC
1	0.2196	0.00044	0.1202
2	0.1315	0.00044	0.1202
3	0.2309	0.00044	0.1202

4	0.1868	0.00044	0.1202
5	0.2196	0.00044	0.1202

D.2 *QUERY* 2

Tabel 7.35 Tabel Nilai *Fitness* dan Nilai *Mean Query Plan*

QP	QAC	QLC	LPC	FITNESS
1	0.468	0.00044	0.3332	0.267
2	0.593	0.00044	0.3336	0.309
3	0.656	0.00044	0.3336	0.330
4	0.593	0.00044	0.3336	0.309
5	0.494	0.00033	0.4172	0.303
MEAN	**0.5608**	**0.00041**	**0.7624**	

Tf = rand [1 + rand(0,1){2 - 1}]

Tf = 1

r1 = 0.742069

r2 = 0.606498

r3 = 0.451934

diff mean QAC = r1 (XQAC – Tf. MEANQAC)

= 0.742069 (0.348 – 1 (0.439))

= 0. 742069 (- 0.091)

= -0.067

diff mean QLC = r1 (XQLC – Tf. MEANQLC)

= 0.606498 (0.00044 – 1 (0.00041))

= 0.00003

diff mean LPC = r1 (XLPC – Tf. MEANLPC)

= 0.451934 (0.532 – 1 (0.7624))

= 0. 451934 (-0.3304)

= -0.00009

D.2.1 TEACHER PHASE

X = Query Plan (QP)

QACnew

X1 = X1old + diff(QAC)
= 0.468 + (-0.067)
= 0.400

X2 = X2old + diff(QAC)
= 0.593 + (-0.067)
= 0.525

X3 = X3old + diff(QAC)
= 0.656 + (-0.067)
= 0.588

X4 = X4old + diff(QAC)
= 0.593 + (-0.067)
= 0.525

X5 = X5old + diff(QAC)
= 0.494 + (-0.067)
= 0.426

QLCnew

X1 = X1old + diff(QLC)
= 0.00044 + 0.00003
= 0.00047

X2 = X2old + diff(QLC)
= 0.00044 + 0.00003
= 0.00047

X3 = X3old + diff(QLC)
= 0.00044 + 0.00003
= 0.00047

X4 = X4old + diff(QLC)
= 0.00044 + 0.00003
= 0.00047

X5 = X5old + diff(QLC)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

= 0.00033 + 0.00003

= 0.00036

LPCnew

X1 = X1old + diff(LPC)

= 0.3332 + (-0.00009)

= 0.3331

X2 = X2old + diff(LPC)

= 0.3336 + (-0.00009)

= 0.3335

X3 = X3old + diff(LPC)

= 0.3336 + (-0.00009)

= 0.3335

X4 = X4old + diff(LPC)

= 0.3336 + (-0.00009)

= 0.3335

X5 = X5old + diff(LPC)

= 0.3332 + (-0.00009)

= 0.3331

Generation 01

Tabel 7.36 Tabel *Update Population* 01 berdasar *Tf*

QP	QAC	QLC	LPC
1	0.4	0.00047	0.3331
2	0.525	0.00047	0.3335
3	0.588	0.00047	0.3335
4	0.525	0.00047	0.3335
5	0.426	0.00036	0.3331

D.2.2 LEARNER PHASE

Generation 01

Tabel 7.37 Tabel Nilai *Fitness Population* 01

QP	QAC	QLC	LPC	FITNESS
1	0.4	0.00047	0.3331	0.244

2	0.525	0.00047	0.3335	0.286
3	0.588	0.00047	0.3335	0.307
4	0.525	0.00047	0.3335	0.286
5	0.426	0.00036	0.3331	0.253

Interaksi antar *learner*, jadi diasumsikan interaksi antara QP1 dengan QP2, QP3, QP4, QP5 dst. Dari hasil interaksi diambil yang terendah.

r1 = 0.576714

r2 = 0.286857

r3 = 0.594324

QAC

1 dan 2

X1 < X2

X2new = X2old + r1(X1 – X2)

= 0.525 + 0.576714(0.4 – 0.525)

= 0.525 + (-0.0072)

= 0.453

1 dan 3

X1 < X3

X3new = X3old + r1(X1 – X3)

= 0.588 + 0.576714(0.4 – 0.588)

= 0.588 + (-0.108)

= 0.480

1 dan 4

X1 < X4

X4new = X4old + r1(X1 – X4)

= 0.525 + 0.576714(0.4 – 0.525)

= 0.525 + (-0.0072)

= 0.453

1 dan 5

X1 < X5

X5new = X5old + r1(X1 – X5)

= 0.426 + 0.576714(0.4 – 0.426)

= 0.426 + (-0.015)

= 0.411

2 dan 3

X2 < X3

X3new = X3old + r1(X2 – X3)

= 0.588 + 0.576714(0.525 – 0.588)

= 0.588 + (-0.036)

= 0.552

2 dan 4

X2 > X4

X2new = X2old + r1(X4 – X2)

= 0.525 + 0.576714(0.525 – 0.525)

= 0.525 + (0)

= 0.525

2 dan 5

X2 > X5

X2new = X2old + r1(X5 – X2)

= 0.525 + 0.576714(0.426 – 0.525)

= 0.525 + (-0.057)

= 0.468

3 dan 4

X3 > X4

X3new = X3old + r1(X4 – X3)

= 0.588+ 0.576714(0.525 – 0.588)

= 0.588 + (-0.036)

= 0.552

3 dan 5

X3 > X5

X5new = X5old + r1(X3 – X5)

= 0.588 + 0.576714(0.281 – 0.627)

= 0.588 + (-0.0.093)

= 0.495

4 dan 5

X4 > X5

X4new = X4old + r1(X5 – X4)

= 0.525 + 0.576714(0.426 – 0.525)

= 0.525 + (-0.057)

= 0.468

CLP

1 dan 2

X1 > X2

X1new = X1old + r3(X2 – X1)

= 0.00047 + 0.594324 (0.00047 – 0.00047)

= 0.00047 + (0)

= 0.00047

1 dan 3

X1 > X3

X1new = X1old + r3(X3 – X1)

= 0.00047 + 0.594324 (0.00047 – 0.00047)

= 0.00047 + (0)

= 0.00047

1 dan 4

X1 > X4

X1new = X1old + r3(X4 – X1)

= 0.00047 + 0.594324 (0.00047 – 0.00047)

= 0.00047 + (0)

= 0.00047

1 dan 5

X1 > X5

X1new = X1old + r3(X5 – X1)

= 0.00047 + 0.594324(0.00036 – 0.00047)

= 0.00047 + (-0.00006)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

= 0.00041

2 dan 3

X2 > X3

X2new = X2old + r3(X3 – X2)

= 0.00047 + 0.594324 (0.00047 – 0.00047)

= 0.00047 + (0)

= 0.00047

2 dan 4

X2 > X4

X2new = X2old + r3(X4 – X2)

= 0.00047 + 0.594324 (0.00047 – 0.00047)

= 0.00047 + (0)

= 0.00047

2 dan 5

X2 > X5

X2new = X2old + r3(X5 – X2)

= 0.00047 + 0.594324(0.00036 – 0.00047)

= 0.00047 + (-0.00006)

= 0.00041

3 dan 4

X3 > X4

X3new = X3old + r3(X4 – X3)

= 0.00047 + 0.594324 (0.00047 – 0.00047)

= 0.00047 + (0)

= 0.00047

3 dan 5

X3 > X5

X5new = X5old + r3(X3 – X5)

= 0.00047 + 0.594324(0.00036 – 0.00047)

= 0.00047 + (-0.00006)

= 0.00041

4 dan 5

X4 > X5

X5new = X5old + r3(X4 – X5)

= 0.00047 + 0.594324(0.00036 – 0.00047)

= 0.00047 + (-0.00006)

= 0.00041

LPC

1 dan 2

X1 < X2

X2new = X2old + r3(X1 – X2)

= 0.3335 + 0.594324 (0.3331 – 0.3335)

= 0.3335 + (-0.00024)

= 0.33326

1 dan 3

X1 < X3

X3new = X3old + r3(X1 – X3)

= 0.3335 + 0.594324 (0.3331 – 0.3335)

= 0.3335 + (-0.00024)

= 0.33326

1 dan 4

X1 < X4

X4new = X4old + r3(X1 – X4)

= 0.3335 + 0.594324 (0.3331 – 0.3335)

= 0.3335 + (-0.00024)

= 0.33326

1 dan 5

X1 > X5

X1new = X1old + r3(X5 – X1)

= 0.3331 + 0.594324(0.3331 – 0.3331)

= 0.3331 + (0)

= 0.3331

2 dan 3

X2 > X3

X2new = X2old + r3(X3 – X2)

= 0.3335 + 0.594324(0.3335 – 0.3335)

= 0.3335 + (0)

= 0.3335

2 dan 4

X2 > X4

X2new = X2old + r3(X4 – X2)

= 0.3335 + 0.594324(0.3335 – 0.3335)

= 0.3335 + (0)

= 0.3335

2 dan 5

X2 > X5

X2new = X2old + r3(X5 – X2)

= 0.3335 + 0.594324 (0.3331 – 0.3335)

= 0.3335 + (-0.00024)

= 0.33326

3 dan 4

X3 > X4

X3new = X3old + r3(X4 – X3)

= 0.3335 + 0.594324(0.3335 – 0.3335)

= 0.3335 + (0)

= 0.3335

3 dan 5

X3 > X5

X5new = X5old + r3(X3 – X5)

= 0.3335 + 0.594324 (0.3331 – 0.3335)

= 0.3335 + (-0.00024)

= 0.33326

4 dan 5

X4 > X5

X5new = X5old + r3(X4 – X5)

= 0.3335 + 0.594324 (0.3331 – 0.3335)

= 0.3335 + (-0.00024)

= 0.33326

Tabel 7.38 Tabel *Update Population* 02 berdasar Interaksi *Learner*

QP	QAC	QLC	LPC
1	0.400	0.00041	0.3331
2	0.453	0.00041	0.3332
3	0.480	0.00041	0.3332
4	0.453	0.00041	0.3332
5	0.411	0.00036	0.3331

D.3 *QUERY* 3

Tabel 7.39 Tabel Nilai *Fitness* dan Nilai *Mean Query Plan*

QP	QAC	QLC	LPC	FITNESS
1	0.594	0.00022	0.3322	0.308
2	0.653	0.00022	0.3322	0.328
3	0.708	0.00022	1.3326	0.347
4	0.736	0.00022	0.3326	0.356
5	0.653	0.00022	0.3322	0.328
MEAN	**0.5798**	**0**	**0.751**	

Tf = rand [1 + rand(0,1){2 - 1}]

Tf = 1

r1 = 0.581999

r2 = 0.87074

r3 = 0.89361

diff mean QAC = r1 (XQAC – Tf. MEANQAC)

= 0.581999 (0.594 – 1 (0.6688))

= 0.581999 (- 0.0748)

= -0.0435

diff mean QLC = r1 (XQLC – Tf. MEANQLC)

= 0. 87074 (0.00022 – 1 (0.00022))

= 0

diff mean LPC = r1 (XLPC – Tf. MEANLPC)

= 0. 89361 (0.3322 – 1 (0.33236))

= 0. 89361 (-0.00006)

= -0.00005

D.3.1 TEACHER PHASE

X = Query Plan (QP)

QACnew

X1 = X1old + diff(QAC)

= 0.594 + (-0.0435)

= 0.5505

X2 = X2old + diff(QAC)

= 0.653 + (-0.0435)

= 0.6095

X3 = X3old + diff(QAC)

= 0.708 + (-0.0435)

= 0.6645

X4 = X4old + diff(QAC)

= 0.736 + (-0.0435)

= 0.6925

X5 = X5old + diff(QAC)

= 0.653 + (-0.0435)

= 0.6095

QLCnew

Difference mean QLC = 0

LPCnew

X1 = X1old + diff(QAC)

= 0.3322 + (-0.00005)

= 0.33215

X2 = X2old + diff(QAC)

= 0.3322 + (-0. 00005)

= 0.33215

X3 = X3old + diff(QAC)

= 1.3326 + (-0. 00005)

= 0.33255

X4 = X4old + diff(QAC)

= 0.3326 + (-0. 00005)

= 0.33255

X5 = X5old + diff(QAC)

= 0.3322 + (-0. 00005)

= 0.33215

Generation 01

Tabel 7.40 Tabel *Update Population* 01 berdasar *Tf*

QP	QAC	QLC	LPC
1	0.5505	0.00022	0.33215
2	0.6095	0.00022	0.33215
3	0.6645	0.00022	0.33255
4	0.6925	0.00022	0.33255
5	0.6095	0.00022	0.33215

D.3.2 LEARNER PHASE

Generation 01

Tabel 7.41 Tabel Nilai *Fitness Population* 01

QP	QAC	QLC	LPC	FITNESS
1	0.5505	0.00022	0.33215	0.294
2	0.6095	0.00022	0.33215	0.314
3	0.6645	0.00022	0.33255	0.332
4	0.6925	0.00022	0.33255	0.341
5	0.6095	0.00022	0.33215	0.314

Interaksi antar *learner*, jadi diasumsikan interaksi antara QP1 dengan QP2, QP3, QP4, QP5 dst. Dari hasil interaksi diambil yang terendah.

r1 = 0.82432

r2 = 0.755522

r3 = 0.732592

QAC

1 dan 2

X1 < X2

X2new = X2old + r1(X1 – X2)

= 0.6095 + 0.82432 (0.5505 – 0.6095)

= 0.6095 + (-0.0486)

= 0.5609

1 dan 3

X1 < X3

X3new = X3old + r1(X1 – X3)

= 0.6645 + 0.82432 (0.5505 – 0.6645)

= 0.423 + (-0.0939)

= 0.5706

1 dan 4

X1 < X4

X4new = X4old + r1(X1 – X4)

= 0.6925 + 0.82432 (0.5505 – 0.6925)

= 0.6925 + (-0.117)

= 0.5755

1 dan 5

X1 < X5

X5new = X5old + r1(X1 – X5)

= 0.6095 + 0.82432 (0.5505 – 0.6095)

= 0.6095 + (-0.0486)

= 0.5609

2 dan 3

X2 < X3

X3new = X3old + r1(X2 – X3)

= 0.6645 + 0.82432 (0.6095 – 0.6645)

= 0.6645 + (-0.045)

= 0.6195

2 dan 4

X2 < X4

X4new = X4old + r1(X2 – X4)

= 0.6925 + 0.82432 (0.6095 – 0. 6925)

= 0.6925 + (-0.068)

= 0.6245

2 dan 5

X2 > X5

X2new = X2old + r1(X5 – X2)

= 0.6095 + 0.82432 (0.6095 – 0.6095)

= 0.6095 + (0)

= 0.6095

3 dan 4

X3 < X4

X4new = X4old + r1(X3 – X4)

= 0.6925+ 0.82432 (0.6645 – 0.6925)

= 0.6925 + (-0.023)

= 0.6695

3 dan 5

X3 > X5

X5new = X5old + r1(X3 – X5)

= 0.6645 + 0.82432 (0.6095 – 0.6645)

= 0.6645 + (-0.045)

= 0.6195

4 dan 5

X4 > X5

X4new = X4old + r1(X5 – X4)

= 0.6925 + 0.82432 (0.6095 – 0.6645)

= 0.6925 + (-0.0684)

= 0.6241

CLP

1 dan 2

X1 > X2

X1new = X1old + r2(X2 – X1)

= 0.00022 + 0.755522 (0.00022 – 0.00022)

= 0.00022 + (0)

= 0.00022

LPC

1 dan 2

X1 > X2

X1new = X1old + r3(X1 – X2)

= 0.33215 + 0.732592 (0.33215 – 0.33215)

= 0.33215 + (0)

= 0.33215

1 dan 3

X1 < X3

X3new = X3old + r3(X1 – X3)

= 0.33255 + 0.732592 (0.33215 – 0.33255)

= 0.33255 + (-0.00029)

= 0.33226

1 dan 4

X1 < X4

X4new = X4old + r3(X1 – X4)

= 0.33255 + 0.732592 (0.33215 – 0.33255)

= 0.33255 + (-0.00029)

= 0.33226

1 dan 5

X1 < X5

X5new = X5old + r3(X1 – X5)

= 0.33215 + 0.732592 (0.33215 – 0.33215)

= 0.33215 + (0)

= 0.33215

2 dan 3

X2 < X3

X3new = X3old + r3(X2 – X3)

= 0.33255 + 0.732592 (0.33215 – 0.33255)

= 0.33255 + (-0.00029)

= 0.33226

2 dan 4

X2 < X4

X4new = X4old + r3(X2 – X4)

= 0.33255 + 0.732592 (0.33215 – 0.33255)

= 0.33255 + (-0.00029)

= 0.33226

2 dan 5

X2 > X5

X2new = X2old + r3(X5 – X2)

= 0.33215 + 0.732592 (0.33215 – 0.33215)

= 0.33215 + (0)

= 0.33215

3 dan 4

X3 < X4

X4new = X4old + r3(X3 – X4)

= 0.33255 + 0.732592 (0.33255 – 0.33255)

= 0.33255 + (0)

= 0.33255

3 dan 5

X3 > X5

X3new = X3old + r3(X5 – X3)

= 0.33255 + 0.732592 (0.33215 – 0.33255)

= 0.33255 + (-0.00029)

= 0.33226

4 dan 5

X4 > X5

X4new = X4old + r3(X5 – X4)

= 0.33255 + 0.732592 (0.33215 – 0.33255)

= 0.33255 + (-0.00029)

= 0.33226

Tabel 7.42 Tabel *Update Population* 02 berdasar Interaksi *Learner*

QP	QAC	QLC	LPC
1	0.5505	0.00022	0.33315
2	0.5609	0.00022	0.33315
3	0.5706	0.00022	0.33226
4	0.5755	0.00022	0.33226
5	0.5609	0.00022	0.33226