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

Lucky Nafi' Agtiannisa [1)], Satrio Agung W., S.Kom., M.Kom.[2)], Niken Hendrakusuma W., S.Kom, M.Kom[2)]

[1)]*Mahasiswa,* [2)]*Dosen Pembimbing*
Program Studi Sistem Infromasi
Fakultas Ilmu Komputer
Universitas Brawijaya, Malang 65145, Indonesia
lucky.agtiannisa@gmail.com

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: Basis Data Terdistribusi, *Optimasi Query, Distributed Query Processing, Teacher Learned Based Optimization (TLBO).*

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: Distributed Database, Optimasi Query, Distributed Query Processing, Teacher Learned Based Optimization (TLBO).

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 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, 2011). Data yang diterima

dari *site* yang berbeda di *database* terdistribusi dikenal sebagai *distributed query processing (QDP).*

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 mengingat banyaknya situs yang diakses, respon *time query* akan menjadi sangat tinggi. 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). *Query processing* dan optimasi menjadi salah satu faktor utama untuk meningkatkan kinerja *query* pada basis data terdistribusi.

Teacher learner based optimization (TLBO) merupakan jenis baru dari metode metaheruristik. Sebuah metode optimasi yang diusulkan untuk mandapatkan solusi dengan sedikit usaha komputasi dan konsistensi tinggi (Kumar, 2015). Metode *Teacher-Learner Based Optimization (TLBO)* didasarkan pada filosofi dari guru dan murid. *Teacher-Learner Based Optimization (TLBO)* merupakan metode yang berbasis populasi dan menggunakan populasi dari solusi untuk dilajutkan ke solusi global, populasi disini merupakan kumpulan murid atau kelas dari murid (Rao, 2011). *Teacher-Learner Based Optimization (TLBO)* bekerja dengan melihat efek dari pengaruh guru terhadap *output* murid di kelas, *output* disini adalah hasil atau grade. Guru secara umum dianggap sebagai orang yang sangat terpelajar yang membagi pengetahuannya kepada murid, kualitas guru mepengaruhi hasil dari murid. Jelas bahwa guru yang bagus melatih muridnya sehingga mereka dapat memiliki hasil yang lebih baik pada *grade* mereka. Selain itu murid juga belajar dari interaksi antar murid sendiri, yang juga membantu hasil belajar mereka(Rao, 2011).

Optimasi *query* pada basis data terdistribusi memiliki dua bagian penting, yakni *query strategy optimizations* dan *local processing optimizations* (Seed, 2013). Salah satu yang penting adalah strategi optimasi *query*, sehingga pada penelitian ini penulis akan melakukan perancangan *query* yang optimal pada database terdistribusi dengan strategi penerapan algoritma *Teacher-Learner Based Optimization(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 keefektifitasan algoritma.

1.2 Rumusan Masalah

Adapun yang menjadi rumusan masalah dalam pembahasan makalah ini adalah :

1. Bagaimana mengimplementasikan metode *Teacher Learner Based Optimization (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 *Teacher Learner Based Optimization (TLBO)*?
3. Bagaimana pengaruh dari tiga parameter heuristik *Query Affinity Cost (QAC), Query Localization Cost (QLC)* dan *Localization Processing Cost (LPC)* dalam *query optimization* dengan metode *Teacher Learner Based Optimization (TLBO)* pada penentuan query yang optimal pada basis data terdistribusi?
4. Bagaimana metode *Teacher Learner Based Optimization (TLBO)* dapat diterapkan sebagai metode dalam *query optimization* pada basis data terdistribusi?

1.3 Batasan Masalah

Agar permasalahan yang dirumuskan lebih terfokus, maka pada penelitian ini dibatasi dalam hal :

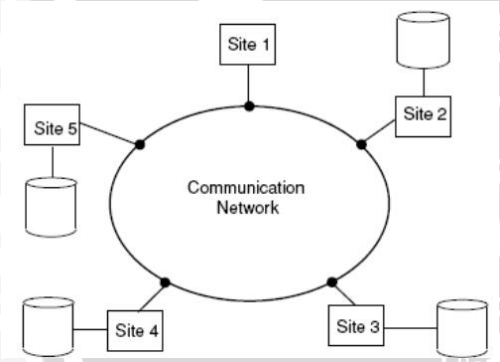
1. Penelitian ini berfokus pada implementasi metode *Teacher Learner Based Optimization (TLBO)* pada basis data terdistribusi untuk optimasi *query*.
2. Pengolahan data dan proses *query* dilakukan pada basis data terdistribusi dengan fragmentasi data secara vertikal.
3. Data yang digunakan pada penelitian 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 *Teacher Learner Based Optimization (TLBO)* didesain dengan mengac pada skema nasi data akademik mahasiswa Universitas Brawijaya.
5. Topologi yang digunakan adalah topologi mesh dan dengan *transfer rate* yang sama.

2. LANDASAN KEPUSTAKAAN

2.1 Sistem 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, *database* terdistribusi memungkinkan integrasi informasi dan pengolahannya dengan aplikasi yang mungkin terpusat atau didistribusikan (Elmasri et al, 2010). Lingkungan basis data terdistribusi ditunjukkan pada gambar 1.

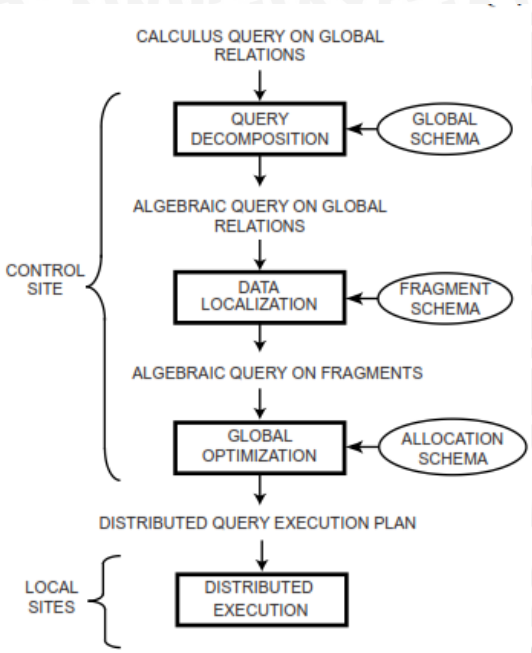


Gambar 1 Lingkungan Basis Data Terdistribusi

Sumber : Valduriez (2011)

2.2 Proses *Query* pada Sistem Basis Data Terdistribusi

Query terdistribusi adalah proses selection dari data pada *database* yang diletakkan pada multiple *site* di sebuah jaringan dan memproses pendistribusian operasi komputasi pada *multiple* CPU untuk menerima hasil tunggal. *Query processing* lebih susah pada lingkungan terdistribusi daripada lingkungan terpusat karena banyak parameter yang mempengaruhi kinerja distributed query. Tujuan *Distributed Query Processing (QDP)* adalah untuk mengeksekusi suatu *query* seefisien mungkin agar meminimalkan respon time dimana user harus menunggu hasil ataupun waktu *delay* dari program aplikasi (Ghaemi et al, 2008).



Gambar 2 Proses *Query* Terdistribusi

Sumber : Valduriez (2011)

Gambar 2 menunjukkan lapisan pada QDP. 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, 2011).

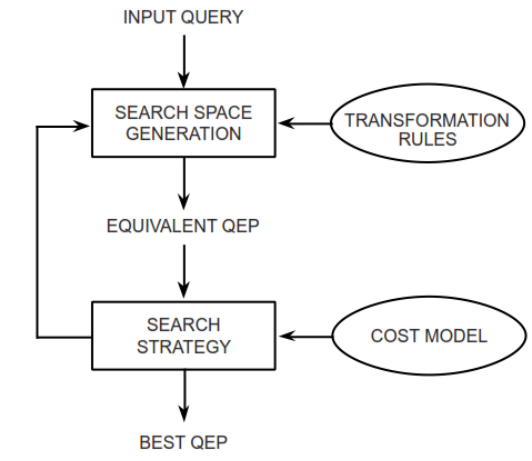
Pada layer ini, *query* keluaran dari layer *query decomposition* berbentuk aljabar dialokasikan menggunakan *data distribution schema* yang ada pada *fragment shema*. Fungsi dari layer ini adalah menentukan *fragment* mana yang digunakan pada sebuah *query* dan mentransformasikan *distributed query*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

Pada layer ini, masukan berupa *query* aljabar yang terdapat pada skema fragmen. Tujuan dari layer ini adalah mencari strategi untuk memproses sebuah *query* dengan maksud mencari biaya yang paling kecil untuk mengoptimalkan *query processing* (Valduriez, 2011).

Lapisan terakhir dilakukan oleh semua situs yang memiliki fragmen yang terlibat dalam *query*. Setiap *subquery* mengeksekusi pada satu situs, yang disebut *query* lokal, kemudian dioptimalkan menggunakan skema situs lokal dan dieksekusi (Valduriez, 2011).

2.3 Optimasi *Query* pada Sistem Basis Data Terdistribusi

Optimasi *query* mengacu pada proses menghasilkan rencana eksekusi *query* (QEP) yang merupakan strategi eksekusi untuk *query*. QEP ini meminimalkan fungsi biaya obyektif. Sebuah *query optimizer*, modul *software* yang melakukan optimasi *query*, biasanya terdiri dari tiga komponen: ruang pencarian, model biaya, dan strategi pencarian (Valduriez, 2011).



Gambar 3 Optimasi *Query*

Sumber : Valduriez (2011)

Gambar 3 adalah proses dari query optimization. Untuk *query* yang diberikan, *search space* sehingga dapat didefinisikan sebagai kumpulan *operator tree* setara yang dapat diproduksi menggunakan aturan transformasi (Valduriez, 2011).

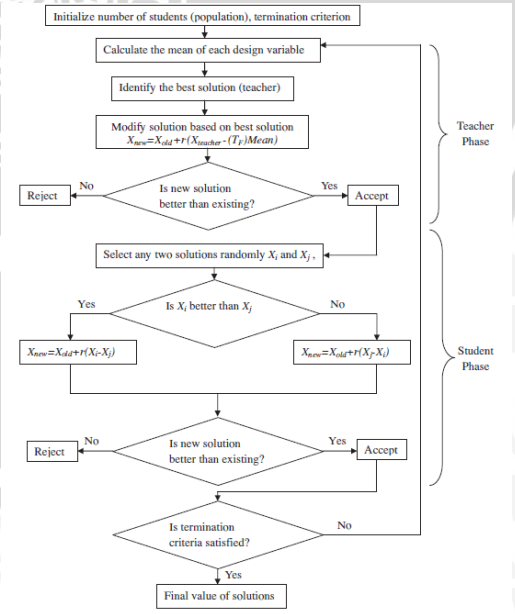
Untuk *search strategy* mengurangi biaya optimasi, rencana parsial yang tidak cenderung mengarah ke rencana optimal dipangkas (yaitu, dibuang) sesegera mungkin (Valduriez, 2011).

Sebuah *cost based ptimization* termasuk fungsi biaya untuk memprediksi biaya operator, statistik dan data dasar, dan formula untuk mengevaluasi ukuran antara hasil-hasil. Biaya adalah dalam hal waktu eksekusi, sehingga fungsi biaya merupakan waktu eksekusi *query*. Biaya strategi eksekusi terdistribusi dapat dinyatakan sehubungan baik dengan total waktu atau waktu respon (Valduriez, 2011).

2.4 *Teacher-Learner Based Optimization (TLBO)*

Algoritma Teacher-Learner Based Optimization adalah sebuah metode yang diinspirasi dari proses belajar-mengajar yang baru-baru ini diusulkan (Rao et al, 2011, 2012) dan berdasar pada efek dari pengaruh guru pada output dari learner atau siswa di kelas. Algoritma ini menganggap sebuah kelompok learner sebagai populasi dan subyek-subyek yang berbeda ditawarkan pada learner yang dianggap sebagai desain obyektif yang berbeda dan hasil dari learner dianalogikan sebagai nilai "fitness" dari permasalahan optimasi. Solusi terbaik dari keseluruhan populasi dianggap sebagai teacher. Algoritma Teacher-Learner Based Optimization terbagi menjadi 2 fase, "teacher phase" dan "learner phase" (Rao et al, 2011, 2012).

Algoritma ini mendeskripsikan dua model dasar pembelajaran (i) melalui guru (dikenal sebagai "teacher phase") dan (ii) interakasi antar murid (dikenal sebagi "learner phase"). Pada algoritma optimasi ini sekelompok learner dianggap sebagai populasi dan subyek berbeda yang ditawarkan pada learner dianggap sebagai desain variable yang berbeda pada permasalahan optimasi dan hasil dari learner dianalogikan sebagai nilai "fitness" dari permasalahan optimasi. Solusi terbaik dari keseluruhan optimasi dianggap sebagai teacher (Rao et al, 2011, 2012).



Gambar 4 Alur Metode TLBO

Sumber : Rao et al (2012)

2.5 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

Jumlah site yang terlibat pada setiap query plan.

$$QAC = \sum_{i=1}^{M} \frac{ki}{N} (1 - \frac{ki}{N}) \quad (2.1)$$

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

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.

$$QPcost = MIN_{i=1}^{Nr}\left[\sum_{j=1 \&\& j \neq i}^{Nr} \frac{Size\ (Rsj)}{\sum_{ki}^{Nr} Size\ (Rk)}\right] \quad (2.2)$$

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 tuples pada relasi di site Sj

Size (Rk) = Jumlah tuples pada relasi Rk

3) Local Processing Cost

Focus pada relasi yang terletak pada site lokal dan kemampuannya untuk memilih pada relasi lokal. Pengukuran LPC dibagi menjadi dua sub-perhitungan, pertama adalah LPC pada remote site dan kedua LPC pada control site. (Rumus asumsi)

Relation Processing Cost

$$RPC = Nt * Sr / \sum_{K=1}^{Nr} Nt\ (k) \quad (2.3)$$

a. LPC untuk remote site yang digunakan pada query plan

$$RLPC = Max_{i=1\ to\ Rs}[RPC(i)] \quad (2.4)$$

b. LPC untuk control site yang digunakan pada query plan

$$CLPC = Max_{i,j-1\ to\ Nr}[Nt(JOIN(Ri,Rj)) * Sj(JOIN(Ri,Rj))/\sum_{k=1}^{Nr} Nt(k)] \quad (2.5)$$

$$LPC\ dari\ query\ plan\ QPi = \sum_{k=1}^{sqp-1}(RLPC(k)) + (CLPC - Max_{i=1\ to\ Nr \&\&} R(i)CS\&JOIN[RPC(i)] \quad (2.6)$$

Nt = Jumlah tuples

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

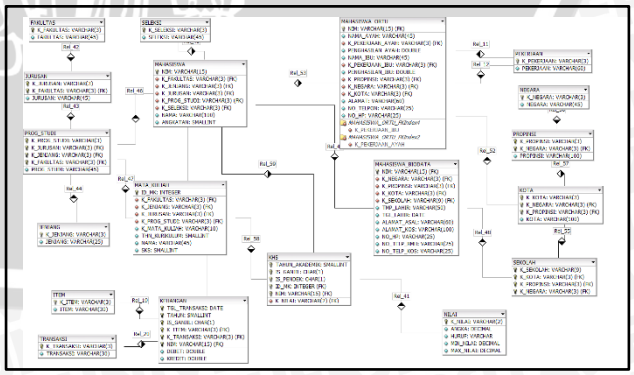
3. METODOLOGI

3.1 Studi Literatur

Pada tahap ini, penulis melakukan studi literatur mengenai basis data terdistribusi, optimasi *query*, fragmentasi data, aljabar realsional dan metode yang digunakan pada penelitian ini yakni *Teacher-Learner Based Optimization* sebagai metode untuk melakukan optimasi *query* pada basis data terdistribusi.

3.2 Studi Kasus

Dalam penelitian ini menggunakan skema basis data sistem informasi akademik mahasiswa dari universitas Brawijaya. Skema basis data ditunjukkan pada gambar 4.



Gambar 5 Skema Basis Data Universitas Brawijaya

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). Tabel 1 merupakan data mahasiswa S1 aktif tahun ajaran 2015/2016.

Tabel 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

Jumlah total mahasiswa dari ke 15 fakultas adalah 48181 mahasiswa.

3.3 Perancangan Basis Data Terdistribusi

Dari skema pada gambar 5 selanjutnya akan dipecah menjadi beberapa *site*. 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. Dihasilkan fragmen yang dapat dilihat pada table 2.

Tabel 2 Jumlah data mahasiswa aktif S1 hasil fragmentasi

No SItus	Akses Data	Relasi	Jumlah Data
1	Mahasiswa	Fakultas	17
		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	-
		Nilai	9
		Fakultas	17
		Jurusan	64
		Program Studi	64
		Jenjang	5
		Seleksi	27
5	Keuangan	Mahasiswa	48181
		Keuangan	96262
		Keterangan	2
		Item	3

3.4 Pemilihan *Query*

Daftar query yang akan diptimasi adalah 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.

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.

4. IMPELEMENTASI

4.1 Lingkungan Implementasi

Pada bab ini akan dijelaskan lingkungan impelementasi, lingkup dari lingku-ngan implementasi perangkat lunak yang digunakan dalam penelitian antara lain :

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

Sedangkan lingkungan implementasi setiap virtual server adalah sebagai berikut:

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

4.2 Query plan

4.2.1 *Query* 1

Query 1 merupakan sebuah query yang di desain untuk menghasilkan data keterangan hasil studi dari masing-masing mahasiswa pada universitas. Pada *query* 1 ini kita membuat 5 model *query* sebagai alternatif *query* untuk pengambilan data.

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
2	1. Tabel Mahasiswa	1. Tabel KHS 5. Tabel Nilai 2. Tabel Mata Kuliah 6. Tabel Fakultas 3. Tabel Jurusan 7. Tabel Jenjang 4. Tabel Prog_Studi
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
4	1. Tabel Mahasiswa 2. Tabel Prog_Studi	1. Tabel KHS 4. Tabel Nilai 2. Tabel Mata Kuliah 5. Tabel Fakultas 3. Tabel Jurusan 6. Tabel Jenjang
5	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan	1. Tabel KHS 4. Tabel Nilai 2. Tabel Mata Kuliah 5. Tabel Prog_Studi 3. Tabel Jenjang

Gambar 6 Daftar *Query Plan Query* 1

4.2.2 *Query* 2

Query 2 merupakan sebuah query yang di desain 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.

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	
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
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
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
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	

Gambar 7 Daftar *Query Plan Query* 2

4.2.3 *Query* 3

Query 3 merupakan sebuah *query* yang di desain 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.

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	
2	1. Tabel Mahasiswa 2. Tabel Fakultas 3. Tabel Jurusan 4. Tabel Jenjang 5. Tabel Prog_Studi	1. Tabel Mhs_Biodata 2. Tabel Negara 3. Tabel Kota 4. Tabel Propinsi	1.Tabel Mhs_ortu 2.Tabel Pekerjaan 3.Tabel Pekerjaan	
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
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
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	

Gambar 8 Daftar *Query Plan Query* 3

4.3 Desain Parameter

4.3.1 *Query* 1

Desain parameter yang kita gunakan berjumlah 3 (*Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC)) dan batas nilainya yaitu nilai minimal 0 dan maksimal 1.

Tabel 3 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 3 berisi nilai dari masing-masing parameter *Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC) bagi tiap *Query Plan*.

4.3.2 *Query* 2

Desain parameter yang kita gunakan berjumlah 3 (*Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC)) dan batas nilainya yaitu nilai minimal 0 dan maksimal 1.

Tabel 4 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	0.00044	0.3332

Pada tabel 4 berisi nilai dari masing-masing parameter *Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC) bagi tiap *Query Plan*.

4.3.3 *Query* 3

Desain parameter yang kita gunakan berjumlah 3 (*Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC)) dan batas nilainya yaitu nilai minimal 0 dan maksimal 1.

Tabel 5 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 5 berisi nilai dari masing-masing parameter *Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC) bagi tiap *Query Plan*.

5. PENGUJIAN DAN ANALISA

5.1 TLBO *Query* 1

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 penulis memilih parameter *fitness* sebagai parameter yang mampu mewakili 3 parameter yang ada. Nilai fitness disini didapatkan dari hasil penghitungan nilai rata-rata dari ketiga parameter. Berikut adalah tabel 6 yang berisi parameter dengan nilai *fitness* dan *mean*.

Tabel 6 Tabel Nilai *Fitness 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	0.500	0.00044	0.1202	0.206
4	0.375	0.00044	0.1202	0.165
5	0.468	0.00044	0.1202	0.196

Dari tabel 6 dia atas sudah terdapat nilai fitness. Pada fase *teacher* akan memilih *Teacher* untuk digunakan merubah nilai dari *Mean* yang ada agar menjadi lebih baik. *Teacher* dipilih dari *Query Plan (learner)* yang dianggap sebagai *learner* yang memiliki pengetahuan tinggi. Dari tabel 6 kita memilih QP yang diasumsikan sebagai *teacher* berdasar nilai *fitness* yang terkecil sehingga QP yang terpilih adalah QP 2 sebagai *Teacher*.

Tabel 7 Tabel Nilai *Mean*

Mean Value	
QAC	**0.405**
QLC	**0.00044**
LPC	**0.1202**

Tabel 8 Tabel *Teachng Factor* atau *Difference Factor*

Difference Factor (Tf)	
QAC	QAC
QLC	QLC
LPC	LPC

Dengan Tf pada tabel 8 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. *Update* Populasi terbaru ditunjukkan pada tabel 9.

Tabel 9 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

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 10.

Tabel 10 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. Hasil dari interaksi antar *learner* akan menghasilkan *Update* Populasi 02 yang ditampilkan pada tabel 11.

Tabel 11 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 11 merupakan hasil akhir dari iterasi metode *Teacher Learner Based Optimization (TLBO)*. Selanjutnya kita akan merangking learner (*Query Plan*) dengan berdasar nilai fitness yang terbaik. Hasil dari perengkingan ditunjukkan pada tabel 12.

Tabel 12 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 12 *QP* yang menempati posisi pertama merupakan alternatif *Query Plan* yang terbaik yang memiliki biaya pemrosesan query yang rendah berdasar *Teacher Learner Based Optimization (TLBO)*.

5.2 TLBO Query 2

Tahap yang sama yakni fase *teacher* dan fase *learner* juga dilakukan untuk *query* 2 sehingga menghasilkan tabel perangkingan seperti ditunjukkan pada tabel 13 .

Tabel 13 Tabel Perangkingan *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 13 *QP* yang menempati posisi pertama merupakan alternatif *Query Plan* yang terbaik yang memiliki biaya pemrosesan query yang rendah berdasar *Teacher Learner Based Optimization (TLBO).*

5.3 TLBO Query 3

Tahap yang sama yakni fase *teacher* dan fase *learner* juga dilakukan untuk query 3 sehingga menghasilkan tabel perangkingan seperti ditunjukkan pada gambar .

Tabel 14 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 14 *QP* yang menempati posisi pertama merupakan alternatif *Query Plan* yang terbaik yang memiliki biaya pemrosesan query yang rendah berdasar *Teacher Learner Based Optimization (TLBO).*

5.4 Analisa Hasil

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* dilakukan perangkingan pada tabel 12 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 *Query Affinity Cost* (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. Untuk parameter *Query Localization Cost* (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 *Query Localization Cost* (QLC) untuk kelima model sama. Sedangkan parameter *Localization Processing Cost* (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 *Localization Processing Cost* (LPC) dari kelima model sama.

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* dilakukan perangkingan pada tabel 13 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 *Query Affinity Cost* (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. Untuk parameter *Query Localization Cost* (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 *Query Localization Cost* (QLC) untuk kelima model sama. Sedangkan parameter *Localization Processing Cost* (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 *Localization Processing Cost* (LPC) dari kelima model sama.

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* dilakukan perangkingan pada tabel 14 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 *Query Affinity Cost* (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. Untuk parameter *Query Localization Cost* (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 *Query Localization Cost* (QLC) untuk kelima model sama. Sedangkan parameter *Localization Processing Cost* (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 *Localization Processing Cost* (LPC) dari kelima model sama.

6. PENUTUP

6.1 Kesimpulan

Berikut adalah kesimpulan dari penelitian yang telah dilakukan :

1. Metode *Teacher Learner Based Optimization* (TLBO) diimplemen-tasikan 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. Selanjutnya dilakukan perhitungan biaya pada query plan berdasar parameter heurisrik dan dilakukan perbandingan dengan metode *Teacher Learner Based Optimization* (TLBO) dan menghasilkan query plan yang paling optimal.
2. Dengan penyusunan 5 *query plan* yang *equivalent* tapi dengan susunan jumlah site dan konsentrasi relasi yang berbeda pada tiap site, menghasilkan biaya yang berbeda berdasar 3 parameter *Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC) berbeda dan hasil dari optimasi *query* dengan metode *Teacher Learner Based Optimization* (TLBO) untuk *query* 1 adalah *query* model 2 dengan jumlah *site* 2 dan jumlah relasi 1 relasi pada *site* Mahasiswa dan 7 relasi pada *site* KHS. *Query* 2 yang paling optimal adalah *query* model 1 dengan jumlah *site* 2 dan jumlah relasi 5 pada *site* Mahasiswa dan 3 relasi pada *site* Keuangan. *Query* 3 yang paling optimal adalah *query* model 1 dengan jumlah *site* 3 dan jumlah relasi 5 pada *site* Mahasiswa, 1 relasi pada *site* Mahasiswa Biodata dan 6 relasi pada *site* Mahasiswa Orang Tua.
3. Dari ketiga parameter heuristik yakni *Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan LPC untuk studi kasus dan *query plan* yang dirancang, parameter *Query Affinity Cost* (QAC) yang paling berpengaruh dari pada kedua parameter lain karena kedua parameter lain mengacu pada relasi serta tupel yang terlibat dimana pada kelima *query plan* relasi dan tupel hampir sama jumlahnya pada keseluruhan *query plan*, sedangkan untuk parameter *Query Affinity Cost* (QAC) jumlah *site* dan konsentrasi relasi pada setiap *site* yang berpengaruh sehingga hasilnya berbeda satu model *query* dengan yang lain pada *query plan*.
4. Penggunaan metode *Teacher Learner Based Optimization* (TLBO) merupakan *query optimization* dimana langkah penerapan-

nya sama seperti *query optimization* pada *query processing* basis data, inisialisasi *query plan* yang 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. Selanjutnya *search strategi* yakni pemilihan parameter pengukur biaya dari setiap *query* dimana pada penelitian ini memilih parameter heuristik yakni *Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC). Selanjutnya untuk 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.

6.2 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 *Teacher Learner Based Optimization* (TLBO) dapat menggunakan parameter lain tidak hanya terpaku pada *Query Affinity Cost* (QAC), *Query Localization Cost* (QLC) dan LPC yang memungkinkan hasil yang lebih maksimal dalam lingkungan basis data terdistribusi yang sangat besar ataupun kecil.

2. Pada penelitian ini parameter *Query Localization Cost* (QLC) dan *Localization Processing Cost* (LPC) kurang menunjukkan perbandingan yang baik pada keseluruhan *query plan* sehingga dibutuhkan penelitian lebih lanjut dengan memperbaiki populasi data.

3. Pengaruh parameter *Query Localization Cost* (QLC) dan *Localization Processing Cost* (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.

7. DAFTAR PUSTAKA

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, 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

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

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. 2012. *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

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/bitstrea

m/handle/123456789/1221/Query%20Optimization%20%20in%20Distributed%20Database%20System%20%20using%20Mobile%20Agent.pdf?sequence=1&isAllowed=y

Valduriez, P., & Ozsu, T. 2011. *Principle of Distributed Database System. 3rd Editions.* Pearson Educations, Inc. New York