

## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	<b>i</b>
<b>PERNYATAAN ORISINALITAS SKRIPSI .....</b>	<b>iii</b>
<b>KATA PENGANTAR .....</b>	<b>iv</b>
<b>ABSTRAK .....</b>	<b>vi</b>
<b>ABSTRACT .....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xvi</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xviii</b>
 <b>BAB I</b>	
<b>PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan Penelitian .....	5
1.5 Manfaat Penelitian .....	5
1.6 Metode Penelitian .....	6
 <b>BAB II</b>	
<b>TINJAUAN PUSTAKA .....</b>	<b>7</b>
2.1 <i>Datawarehouse</i> .....	7
2.2 Karakteristik <i>Datawarehouse</i> .....	8
2.3 Struktur <i>Datawarehouse</i> .....	10
2.4 OLAP ( <i>Online Analytical Processing</i> ) .....	11
2.5 ETL ( <i>Extraction, Transformation, dan Loading</i> ).....	13
2.6 Model Data Multidimensi.....	16
2.6.1 Tabel Fakta dan Tabel Dimensi.....	18
2.6.2 Skema Model Data Multidimensi.....	18

2.7	Arsitektur Komponen OLAP Pentaho .....	20
2.8	Sistem Pendukung Keputusan .....	22
2.9	Logika <i>Fuzzy</i> .....	22
2.10	<i>Fuzzy Database</i> .....	25
2.11	Akurasi dan Evaluasi .....	29
2.12	Pengujian <i>Black Box</i> .....	30

### **BAB III**

#### **METODOLOGI DAN PERANCANGAN SISTEM ..... 31**

3.1	Analisa Sistem .....	32
3.1.1	Deskripsi Sistem .....	32
3.1.2	Analisa Kebutuhan Fungsional Sistem.....	33
3.1.3	Analisa Kebutuhan Non-Fungsional Sistem .....	34
3.2	Perancangan <i>Datawarehouse</i> .....	35
3.2.1	Analisa Sumber Data .....	35
3.2.2	Arsitektur <i>Datawarehouse</i> .....	38
3.2.3	Perancangan Skema <i>Datawarehouse</i> .....	39
3.2.4	Proses ETL ( <i>Extraction, Transformation, Loading</i> ) .....	44
3.2.5	Struktur Kubus Data .....	47
3.3	Perancangan Sistem .....	49
3.3.1	Alur Kerja Sistem .....	49
3.3.2	Diagram Alir.....	50
3.3.2.1	Menentukan Fungsi Keanggotaan .....	52
3.3.2.2	Menghitung <i>Firestrength</i> .....	56
3.3.2.3	Menghitung Nilai Minimum.....	57
3.4	Perancangan Antarmuka .....	58
3.5	Perhitungan Manual.....	61
3.6	Rancangan Pengujian Penelitian.....	71
3.6.1	Pengujian Perbandingan Hasil Sistem dengan Data Aktual.....	71
3.6.2	<i>Integration Testing</i> .....	74
3.6.3	Pengujian <i>Black Box</i> .....	74
3.6.4	<i>Acceptance Testing</i> .....	75



**BAB IV**

**IMPLEMENTASI ..... 77**

4.1 Lingkungan Implementasi .....77

    4.1.1 Lingkungan Perangkat Keras.....77

    4.1.2 Lingkungan Perangkat Lunak.....77

4.2 Implementasi Program.....78

    4.2.1 Implementasi ETL (*Extraction, Transformation, Loading*).....78

        4.2.1.1 Proses ETL dim\_siswa .....78

        4.2.1.2 Proses ETL dim\_kelas .....80

        4.2.1.3 Proses ETL dim\_mapel.....82

        4.2.1.4 Proses ETL dim\_periode .....84

        4.2.1.5 Proses ETL dim\_tatib .....86

        4.2.1.6 Proses ETL fact\_nilai .....88

        4.2.1.7 Proses ETL fact\_tatib .....91

        4.2.1.8 Proses ETL fact\_penghasilan .....94

        4.2.1.9 Proses ETL job\_basiswa .....96

    4.2.2 Implementasi Struktur Kubus Data (*Cube*) OLAP.....98

    4.2.3 Implementasi Program Rekomendasi Beasiswa.....102

        4.2.3.1 Menentukan Batasan Parameter .....102

        4.2.3.2 Mengambil Nilai pada Semua Parameter  
                untuk Setiap Siswa.....103

        4.2.3.3 Menentukan Derajat Keanggotaan .....105

        4.2.3.4 Menghitung *Firestrength*.....107

        4.2.3.5 Mendapatkan Calon Penerima Beasiswa.....108

        4.2.3.6 Menghitung Presentase Perbandingan Hasil .....109

4.3 Implementasi Antarmuka.....110

    4.3.1 Implementasi Antarmuka Rekap Laporan .....110

    4.3.2 Implementasi Antarmuka Rekomendasi Beasiswa.....114



4.3.2.1	Halaman Login .....	114
4.3.2.2	Halaman Beranda.....	114
4.3.2.3	Halaman Rekomendasi .....	115
4.3.2.4	Halaman Pengujian .....	117
4.3.2.5	Halaman <i>User</i> .....	118

**BAB V**

**PENGUJIAN DAN ANALISIS ..... 120**

5.1	Implementasi Pengujian.....	120
5.1.1	Pengujian Perbandingan Sistem terhadap Data Aktual .....	120
5.1.2	<i>Integration Testing</i> .....	123
5.1.3	Pengujian <i>Black Box</i> .....	124
5.1.4	Pengujian <i>User Acceptance</i> .....	124
5.2	Analisa Hasil Pengujian.....	124
5.2.1	Analisa Hasil Perbandingan Sistem terhadap Data Aktual .....	124
5.2.2	Analisa Hasil <i>Integration Testing</i> .....	126
5.2.3	Analisa Hasil Pengujian <i>Black Box</i> .....	129
5.2.4	Analisa Hasil Pengujian <i>User Acceptance</i> .....	129

**BAB VI**

**PENUTUP ..... 130**

6.1	Kesimpulan.....	130
6.2	Saran .....	132

**DAFTAR PUSTAKA ..... 133**



## DAFTAR GAMBAR

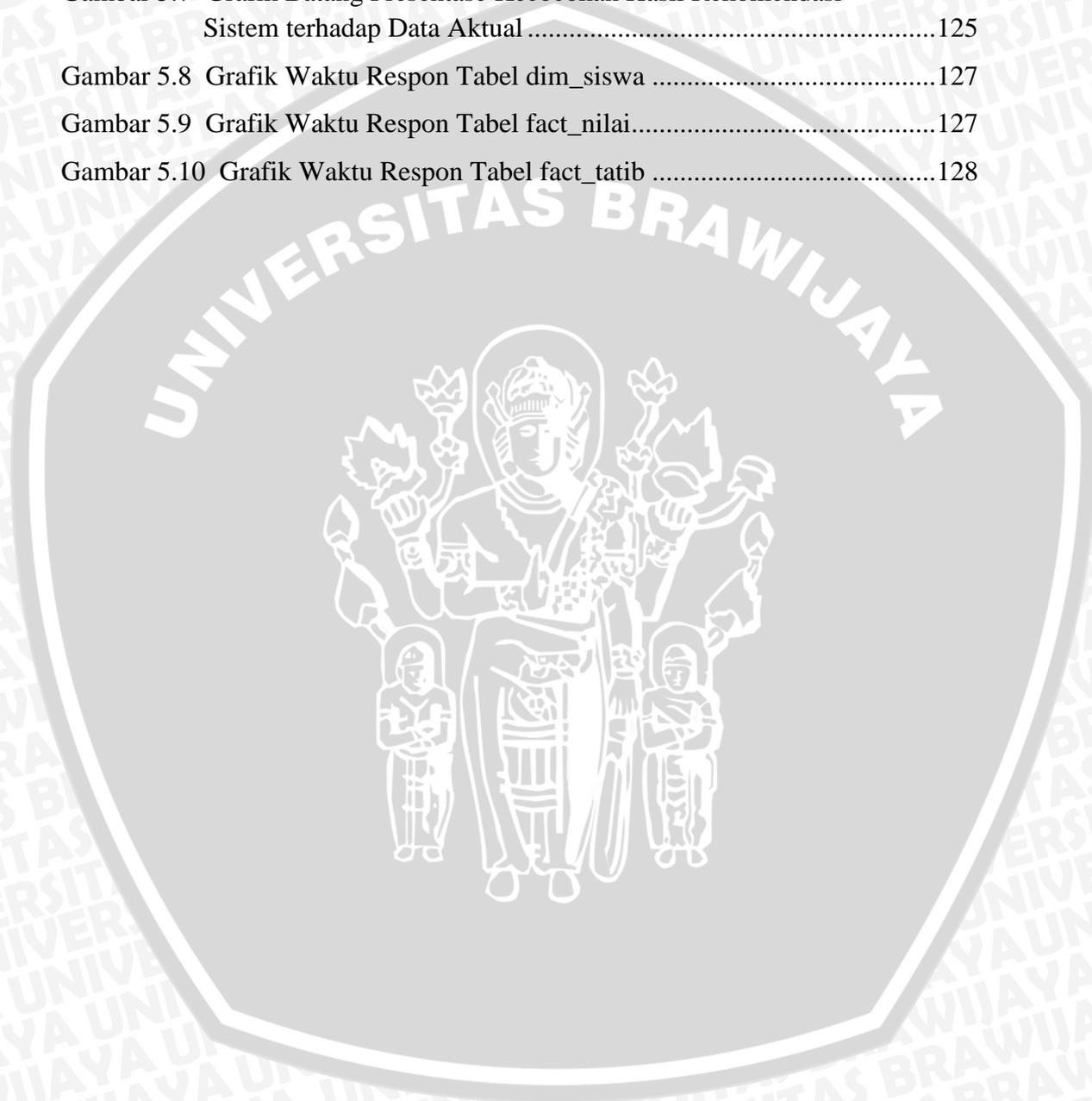
Gambar 2.1 Struktur Datawarehouse .....	11
Gambar 2.2 <i>Star Scheme</i> .....	19
Gambar 2.3 <i>Snowflake Scheme</i> .....	19
Gambar 2.4 <i>Galaxy Scheme</i> .....	20
Gambar 2.5 Gambaran Komponen OLAP Pentaho .....	21
Gambar 2.6 Fungsi Keanggotaan Kurva Segitiga.....	23
Gambar 2.8 Himpunan Fuzzy Untuk Suatu Variabel .....	24
Gambar 2.7 Fungsi Keanggotaan Kurva Bahu .....	24
Gambar 2.9 Fungsi Keanggotaan untuk Variabel Usia.....	26
Gambar 2.10 Fungsi Keanggotaan untuk Variabel Gaji .....	27
Gambar 3.1 Langkah Penelitian.....	31
Gambar 3.2 Diagram Relasional Data Akademik SMAN 7 Malang .....	35
Gambar 3.3 Arsitektur Tiga Tingkat <i>Datawarehouse</i> .....	38
Gambar 3.4 <i>Galaxy Scheme</i> .....	40
Gambar 3.5 UML Proses <i>Loading</i> Tabel <i>dim_kelas</i> .....	44
Gambar 3.6 UML Proses <i>Loading</i> Tabel <i>dim_siswa</i> .....	44
Gambar 3.7 UML Proses <i>Loading</i> Tabel <i>dim_mapel</i> .....	45
Gambar 3.8 UML Proses <i>Loading</i> Tabel <i>dim_tatib</i> .....	45
Gambar 3.9 UML Proses Agregasi Nilai Siswa .....	46
Gambar 3.10 UML Proses <i>Loading</i> Tabel <i>fakta_nilai</i> .....	46
Gambar 3.11 UML Proses Agregasi Nilai Penghasilan Orang Tua Siswa.....	46
Gambar 3.12 UML Proses <i>Loading</i> Tabel <i>fakta_penghasilan_ortu</i> .....	47
Gambar 3.13 Alur Kerja Sistem.....	49
Gambar 3.14 Diagram Alir Utama Perancangan Sistem .....	51
Gambar 3.15 Diagram Alir Menentukan Fungsi Keanggotaan "Parameter Kurang" .....	53
Gambar 3.16 Diagram Alir Menentukan Fungsi Keanggotaan "Parameter Cukup" .....	54
Gambar 3.17 Diagram Alir Menentukan Fungsi Keanggotaan "Parameter Baik" .....	55

Gambar 3.18	Diagram Alir Menghitung <i>Firestrength</i> .....	56
Gambar 3.19	Diagram Alir Menghitung Nilai Minimum.....	57
Gambar 3.20	Rancangan Antarmuka Rekap Laporan Data Akademik Siswa.....	58
Gambar 3.21	Rancangan Antarmuka Proses Rekomendasi Beasiswa.....	59
Gambar 3.22	Rancangan Antarmuka Proses Pengujian Perbandingan.....	60
Gambar 3.23	Kurva Derajat Keanggotaan Nilai Siswa .....	62
Gambar 3.24	Kurva Derajat Keanggotaan Penghasilan Orang Tua .....	63
Gambar 3.25	Kurva Derajat Keanggotaan Kedisiplinan.....	64
Gambar 4.1	Skema ETL dim_siswa .....	78
Gambar 4.2	Hasil Proses ETL dim_siswa pada <i>Kettle</i> .....	79
Gambar 4.3	Hasil Proses ETL dim_siswa pada MySQL.....	80
Gambar 4.4	Skema ETL dim_kelas .....	80
Gambar 4.5	Hasil Proses ETL dim_kelas pada <i>Kettle</i> .....	81
Gambar 4.6	Hasil Proses ETL dim_kelas pada MySQL.....	82
Gambar 4.7	Skema ETL dim_mapel.....	82
Gambar 4.8	Hasil Proses ETL dim_mapel pada <i>Kettle</i> .....	83
Gambar 4.9	Hasil Proses ETL dim_mapel pada MySQL.....	84
Gambar 4.10	Skema ETL dim_periode .....	84
Gambar 4.11	Hasil Proses ETL dim_periode pada <i>Kettle</i> .....	85
Gambar 4.12	Hasil Proses ETL dim_periode pada MySQL.....	86
Gambar 4.13	Skema ETL dim_tatib .....	86
Gambar 4.14	Hasil Proses ETL dim_tatib pada <i>Kettle</i> .....	87
Gambar 4.15	Hasil Proses ETL dim_tatib pada MySQL.....	88
Gambar 4.16	Skema ETL fact_nilai.....	88
Gambar 4.17	Hasil Proses ETL fact_nilai pada <i>Kettle</i> .....	91
Gambar 4.18	Hasil Proses ETL fact_nilai pada MySQL.....	91
Gambar 4.19	Skema ETL fact_tatib.....	91
Gambar 4.20	Hasil Proses ETL fact_tatib pada <i>Kettle</i> .....	93
Gambar 4.21	Hasil Proses ETL fact_tatib pada MySQL.....	94
Gambar 4.22	Skema ETL fact_penghasilan.....	94
Gambar 4.23	Hasil Proses ETL fact_penghasilan pada <i>Kettle</i> .....	95



Gambar 4.24 Hasil Proses ETL fact_penghasilan pada MySQL.....	96
Gambar 4.25 Skema ETL fact_penghasilan.....	96
Gambar 4.26 Hasil Proses ETL job_beasiswa pada <i>Kettle</i> .....	97
Gambar 4.27 Konfigurasi koneksi <i>database</i> MySQL pada Mondrian. ....	98
Gambar 4.28 Struktur Kubus Data Penghasilan .....	99
Gambar 4.29 Struktur Kubus Data Nilai.....	100
Gambar 4.30 Struktur Kubus Data Kedisiplinan .....	101
Gambar 4.31 Rekap Laporan Penghasilan Orang Tua.....	110
Gambar 4.32 Grafik Batang Rekap Laporan Penghasilan Orang Tua .....	111
Gambar 4.33 Rekap Laporan Nilai .....	112
Gambar 4.34 Grafik Batang Rekap Laporan Nilai.....	112
Gambar 4.35 Rekap Laporan Kedisiplinan (Pelanggaran Tata Tertib).....	113
Gambar 4.36 Grafik Batang Rekap Laporan Kedisiplinan (Pelanggaran Tata Tertib) .....	113
Gambar 4.37 Halaman Login.....	114
Gambar 4.38 Halaman Beranda .....	115
Gambar 4.39 Halaman Rekomendasi.....	116
Gambar 4.40 Derajat Keanggotaan Siswa .....	116
Gambar 4.41 Tabel Hasil Rekomendasi.....	117
Gambar 4.42 Halaman <i>Upload</i> File Excel .....	118
Gambar 4.43 Halaman Hasil Pengujian Perbandingan .....	118
Gambar 4.44 Halaman <i>User</i> .....	119
Gambar 4.45 Halaman Tambah Data User .....	119
Gambar 4.46 Halaman <i>Update</i> Data <i>User</i> .....	119
Gambar 5.1 Hasil Uji Perbandingan Kelas X dan Tahun Ajaran 2010-2011 .....	121
Gambar 5.2 Hasil Uji Perbandingan Kelas XI dan Tahun Ajaran 2010-2011 .....	121
Gambar 5.3 Hasil Uji Perbandingan Kelas XII dan Tahun Ajaran 2010-2011 .....	122
Gambar 5.4 Hasil Uji Perbandingan Kelas X dan Tahun Ajaran 2011-2012.....	122

Gambar 5.5 Hasil Uji Perbandingan Kelas XI dan Tahun Ajaran 2011-2012.....	122
Gambar 5.6 Hasil Uji Perbandingan Kelas XI dan Tahun Ajaran 2011-2012.....	123
Gambar 5.7 Grafik Batang Presentase Kecocokan Hasil Rekomendasi Sistem terhadap Data Aktual .....	125
Gambar 5.8 Grafik Waktu Respon Tabel dim_siswa .....	127
Gambar 5.9 Grafik Waktu Respon Tabel fact_nilai.....	127
Gambar 5.10 Grafik Waktu Respon Tabel fact_tatib .....	128



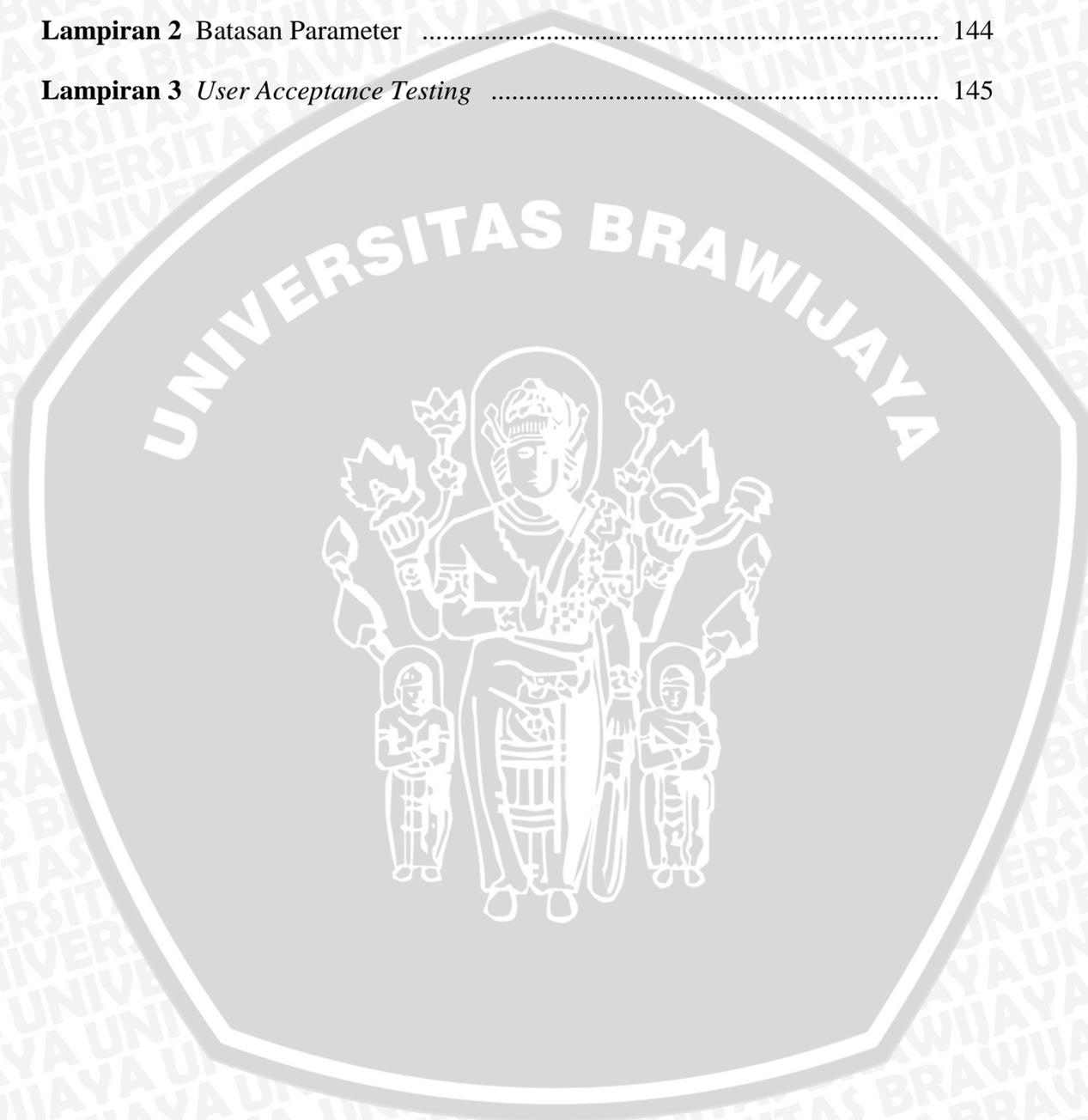
## DAFTAR TABEL

Tabel 2.1 Perbedaan Datawarehouse dengan Data Operasional.....	8
Tabel 2.2 Isi Tentang Transformasi Data.....	13
Tabel 2.3 Mekanisme ETL [TRU-03].....	15
Tabel 2.4 Karyawan Berdasarkan Umur.....	27
Tabel 2.5 Karyawan Berdasarkan Gaji .....	28
Tabel 2.6 Hasil <i>Query</i> dan Penentuan <i>Firestrength</i> .....	29
Tabel 3.1 Tabel dim_siswa .....	40
Tabel 3.2 Tabel dim_periode .....	41
Tabel 3.3 Tabel dim_kelas .....	41
Tabel 3.4 Tabel dim_tatib .....	42
Tabel 3.5 Tabel dim_mapel.....	42
Tabel 3.6 Tabel fakta_penghasilan_ortu.....	42
Tabel 3.7 Tabel fakta_nilai .....	43
Tabel 3.8 Tabel fakta_tatib .....	43
Tabel 3.9 Nilai Batasan Setiap Parameter.....	61
Tabel 3.10 Sampel Data Siswa dan Nilai Dari Setiap Parameter .....	65
Tabel 3.11 Nilai <i>Firestrength</i> .....	69
Tabel 3.12 Tabel Hasil Rekomendasi Beasiswa .....	70
Tabel 3.13 Hasil Rekomendasi Beasiswa oleh Sistem .....	72
Tabel 3.14 Data Aktual Hasil Rekomendasi Beasiswa.....	73
Tabel 3.15 Hasil Perbandingan Waktu Eksekusi ETL setiap Tabel .....	74
Tabel 3.16 Rancangan Pengujian <i>Black Box</i> .....	75
Tabel 3.17 Angket Terhadap Kemudahan Dalam Olah Data .....	76
Tabel 3.18 Angket Terhadap Hasil Rekomendasi Beasiswa .....	76
Tabel 3.19 Angket Terhadap Tampilan Laporan .....	76
Tabel 3.20 Angket Terhadap Antarmuka Aplikasi .....	76
Tabel 4.1 Keterangan Skema ETL dim_siswa.....	78
Tabel 4.2 Log dari Proses ETL dim_siswa .....	79
Tabel 4.3 Keterangan Skema ETL dim_kelas.....	80

Tabel 4.4 Log dari Proses ETL dim_kelas.....	81
Tabel 4.5 Keterangan Skema ETL dim_mapel.....	82
Tabel 4.6 Log dari Proses ETL dim_mapel.....	83
Tabel 4.7 Keterangan Skema ETL dim_periode.....	84
Tabel 4.8 Log dari Proses ETL dim_periode.....	85
Tabel 4.9 Keterangan Skema ETL dim_tatib.....	86
Tabel 4.10 Log dari Proses ETL dim_tatib.....	87
Tabel 4.11 Keterangan Skema ETL fact_nilai.....	89
Tabel 4.12 Log dari Proses ETL fact_nilai.....	90
Tabel 4.13 Keterangan Skema ETL fact_tatib.....	92
Tabel 4.14 Log dari Proses ETL fact_tatib.....	93
Tabel 4.15 Keterangan Skema ETL fact_penghasilan.....	94
Tabel 4.16 Log dari Proses ETL fact_tatib.....	95
Tabel 4.17 Keterangan Skema ETL job_basiswa.....	97
Tabel 4.18 <i>Source Code</i> Menentukan Batasan Parameter.....	102
Tabel 4.19 <i>Source Code</i> Mengambil Nilai Setiap Parameter dari Semua Siswa.....	103
Tabel 4.20 <i>Source Code Query</i> Mengambil Nilai Setiap Parameter.....	104
Tabel 4.21 <i>Source Code</i> Inisialisasi Derajat Keanggotaan.....	105
Tabel 4.22 <i>Source Code</i> Menentukan Rumus Derajat Keanggotaan.....	106
Tabel 4.23 <i>Source Code</i> Menghitung <i>Firestrength</i> .....	107
Tabel 4.24 <i>Source Code</i> Mendapatkan Calon Penerima Beasiswa.....	109
Tabel 4.25 <i>Source Code</i> Menghitung Presentase Perbandingan Hasil.....	109
Tabel 5.1 Hasil <i>Integration Testing</i> skema ETL.....	124
Tabel 5.2 Tabel Hasil Perhitungan Uji Perbandingan.....	125

## DAFTAR LAMPIRAN

<b>Lampiran 1</b> Implementasi Pengujian <i>Black Box</i> .....	136
<b>Lampiran 2</b> Batasan Parameter .....	144
<b>Lampiran 3</b> <i>User Acceptance Testing</i> .....	145



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan dan kemajuan dalam bidang teknologi informasi telah memberikan dampak yang begitu besar dalam kehidupan, salah satunya adalah terciptanya sebuah Sistem Informasi yang menyediakan data dan informasi secara cepat dan lengkap. Dengan adanya data dan informasi yang tersaji secara cepat dan lengkap itulah, maka beberapa pihak dapat menggunakannya dalam proses pengambilan keputusan, salah satu contohnya adalah sistem pengambilan keputusan dalam merekomendasikan siswa yang berhak menerima beasiswa pada SMAN 7 Malang. Dengan demikian diperlukan data dan informasi yang berkaitan dengan data akademik siswa pada SMA Negeri 7 Malang. Data dan informasi tersebut akan digunakan dalam proses pengambilan keputusan yang terkait dengan proses rekomendasi beasiswa pada SMAN 7 Malang.

Data akademik siswa yang begitu banyak akan menjadi suatu permasalahan tersendiri dalam proses analisa akademik siswa dan proses rekomendasi beasiswa, sehingga akan mempersulit pihak akademik dalam mengambil sebuah keputusan yang terkait dengan kedua aspek tersebut. Dengan hanya menggunakan data-data yang belum terintegrasi secara maksimal, pihak akademik harus mengkategorikan siswa berdasarkan parameter yang digunakan dalam proses seleksi, antara lain : nilai akademik siswa pada rapor, penghasilan kedua orang tua, dan perilaku siswa yang berupa tingkat kedisiplinan siswa berdasarkan banyaknya pelanggaran yang dilakukan oleh siswa. Selain itu, pihak akademik juga harus menganalisa setiap siswa yang telah dikategorikan berdasarkan parameter tersebut untuk menentukan daftar siswa yang berhak menerima beasiswa. Hal ini juga akan membutuhkan waktu yang lebih lama, karena pihak akademik harus membandingkan data satu siswa dengan yang lain berdasarkan parameter-parameter yang dibuat.

Berdasarkan permasalahan tersebut, maka dibuatkanlah sebuah sistem yang dapat memberikan informasi berupa laporan hasil analisa data akademik siswa yang nantinya akan digunakan dalam proses pengambilan keputusan untuk

merekomendasikan siswa yang berhak menerima beasiswa pada SMAN 7 Malang. Data operasional yang telah tersedia akan disimpan dalam sebuah *datawarehouse* agar data-data tersebut dapat terintegrasi secara maksimal. Ada beberapa hal yang dapat dilakukan untuk analisis data dari *datawarehouse* agar menghasilkan informasi yang dapat membantu pihak akademik dalam mengambil keputusan yang terkait dengan proses seleksi penerimaan beasiswa. Salah satunya adalah menggunakan OLAP (*On-Line Analytical Processing*). Dengan menggunakan OLAP akan dapat memberikan tingkatan analisis data dengan kapabilitas *query* yang kompleks, perbandingan kecenderungan data, serta *reporting*. OLAP menghasilkan informasi secara multidimensi, artinya mampu melihat data dari berbagai sudut pandang, sehingga memungkinkan pengguna menganalisis data secara interaktif [BOU-09].

Sedangkan dalam merekomendasikan siswa yang berhak menerima beasiswa akan digunakan metode *Fuzzy Database* model Tahani. Dalam proses rekomendasi beasiswa tersebut terdapat beberapa variabel linguistik yang tidak dijabarkan dalam bentuk angka, misalnya adalah parameter kedisiplinan siswa yang mempunyai nilai linguistik yaitu kurang, cukup, dan baik. Dengan demikian, maka akan terjadi ambiguitas dan akan sulit jika menggunakan variabel linguistik tersebut untuk proses rekomendasi beasiswa. Oleh karena itu, diperlukan logika *Fuzzy* agar variabel linguistik tersebut dapat dimodelkan dalam menentukan nilai dari setiap parameter yang digunakan dalam proses rekomendasi beasiswa.

Logika *Fuzzy* adalah suatu proses pengambilan keputusan berbasis aturan yang bertujuan untuk memecahkan masalah, dimana sistem tersebut sulit untuk dimodelkan atau terdapat ambiguitas dan ketidakjelasan yang berlimpah. Logika *Fuzzy* ditentukan oleh persamaan logika bukan dari persamaan diferensial kompleks. Sistem logika *fuzzy* terdiri dari himpunan *fuzzy* dan aturan *fuzzy*. *Subset fuzzy* merupakan himpunan bagian yang berbeda dari variabel *input* dan *output*. Aturan *fuzzy* berhubungan dengan variabel masukan dan variabel output melalui *subset* [HEL-01].

Pada penelitian sebelumnya, metode *Fuzzy Database* model Tahani pernah digunakan dalam proses rekomendasi pemilihan produk *handphone*. Dengan logika *fuzzy*, data akan dikelompokkan ke dalam beberapa himpunan *Fuzzy*. Pada

dasarnya untuk memilih suatu tipe handphone yang layak direkomendasikan ke Customer dapat dilakukan dengan system logika klasik yang diterapkan pada server-server database yang ada sekarang ini. Namun hal ini akan menyebabkan data akan diolah secara tegas, sehingga tipe-tipe handphone yang memiliki kriteria yang mendekati pendefinisian tidak akan direkomendasikan [AHM-2011]. Dalam proses rekomendasi beasiswa juga terdapat suatu kriteria yang diberikan dalam pengambilan keputusan yang melibatkan penggunaan variabel linguistik. Untuk itu sistem *database* perlu diintegrasikan dengan *fuzzy logic* (logika fuzzy) yang memiliki distribusi probabilitas yang digunakan untuk memodelkan data, sehingga didapatkan sebuah system *fuzzy database*. Dengan menggunakan *fuzzy database* maka akan didapatkan pengambilan informasi yang bersifat fleksibel.

Dalam penelitian ini, metode *fuzzy database* digunakan untuk merekomendasikan calon penerima beasiswa. Dalam proses rekomendasi beasiswa tersebut diperlukan suatu adanya pengelompokan dari ketiga parameter menjadi beberapa himpunan berdasarkan sifat dan ukurannya. Dari pengelompokan tersebut, maka dapat dihitung derajat keanggotaan dari setiap parameter serta nilai *firestrength* yang mewakili seberapa besar rekomendasi siswa tersebut untuk memperoleh beasiswa. Sehingga untuk setiap siswa akan mempunyai nilai yang berupa presentase besarnya rekomendasi beasiswa dalam sistem pengambilan keputusan ini.

Kemudian dari hasil proses-proses tersebut data akan disajikan dalam bentuk grafik dan tabel, sehingga dapat memudahkan pihak akademik dalam menganalisa data siswa dan membantu dalam proses pengambilan keputusan yang terkait dengan seleksi penerimaan beasiswa.

Berdasarkan latar belakang yang telah disebutkan, maka digunakanlah judul penelitian “**Implementasi Datawarehouse dalam Menganalisa Data Akademik Siswa untuk Proses Rekomendasi Beasiswa dengan Menggunakan Fuzzy Database**”.

## 1.2 Rumusan Masalah

Permasalahan yang dibahas dalam penelitian ini adalah:

1. Bagaimana mengintegrasikan dan mengagregasikan data-data pada sistem database menjadi sebuah informasi yang dibutuhkan dalam proses pengambilan keputusan (dalam hal ini adalah perekomendasi penerima beasiswa).
2. Bagaimana menganalisa data-data akademik siswa dengan menggunakan OLAP (On-Line Analytical Processing) sehingga membantu pihak akademik dalam mendapatkan informasi dan data-data siswa yang terkait dengan proses seleksi penerimaan beasiswa.
3. Bagaimana menerapkan metode *Fuzzy Database* dalam merekomendasikan siswa yang berhak menerima beasiswa.
4. Bagaimana melakukan proses pengujian terhadap metode *Fuzzy Database* dalam sistem pendukung keputusan proses perekomendasi beasiswa pada SMAN 7 Malang.
5. Bagaimana melakukan proses pengujian fungsional terhadap sistem pendukung keputusan rekomendasi beasiswa dengan menggunakan pengujian *balck box*.

## 1.3 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini, antara lain :

1. Data yang digunakan untuk dianalisa berasal dari data akademik siswa pada SMA Negeri 7 Malang.
2. Metode yang digunakan dalam menganalisa data akademik siswa adalah dengan menggunakan OLAP, sedangkan dalam merekomendasikan penerima beasiswa digunakan metode *Fuzzy Database*.
3. Hal yang dibahas dalam penelitian ini adalah untuk dapat mengetahui daftar siswa yang berhak menerima beasiswa.

4. Parameter yang digunakan dalam menentukan calon penerima beasiswa adalah nilai akademik siswa, penghasilan kedua orang tua, nilai kedisiplinan siswa.

#### 1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini, antara lain :

1. Mengintegrasikan dan mengagregasikan data-data pada sistem database menjadi sebuah informasi yang dibutuhkan dalam proses pengambilan keputusan.
2. Menganalisa data-data akademik siswa dengan menggunakan OLAP (*On-Line Analytical Processing*) sehingga membantu pihak akademik dalam mendapatkan informasi dan data-data siswa yang terkait dengan proses seleksi penerimaan beasiswa.
3. Menerapkan metode *Fuzzy Database* dalam merekomendasikan siswa yang berhak menerima beasiswa.
4. Melakukan proses pengujian apakah metode *Fuzzy Database* ini dapat diterapkan secara maksimal dalam sistem pendukung keputusan proses perekomendasi beasiswa pada SMAN 7 Malang.
5. Melakukan proses pengujian fungsional terhadap sistem pendukung keputusan rekomendasi beasiswa dengan menggunakan pengujian *balck box*.

#### 1.5 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah :

1. Mempermudah penyediaan informasi yang dibutuhkan pihak akademik dalam pengambilan keputusan terkait dengan seleksi penerimaan beasiswa.
2. Mempercepat proses penyeleksian terhadap calon penrima beasiswa.
3. Analisa terhadap data-data yang ada dapat dilakukan secara multidimensional.

## 1.6 Metode Penelitian

Metode yang akan digunakan dalam penelitian ini terdiri dari langkah-langkah berikut:

### 1. Studi Literatur

Pada tahap ini dilakukan pemahaman kepustakaan mengenai perancangan dan pembuatan ETL, datawarehouse, OLAP, serta metode Fuzzy.

### 2. Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan data dari sistem dan mengidentifikasi kebutuhan dari pengguna terhadap sistem yang akan dibuat.

### 3. Perancangan perangkat lunak

Pada tahap ini dilakukan perancangan dari hasil analisis kebutuhan yang telah dilakukan, meliputi desain rancangan proses ETL, datawarehouse dan teknologi OLAP yang akan dibuat, serta digunakan metode Fuzzy dalam proses rekomendasi penerima beasiswa.

### 4. Implementasi perangkat lunak

Pada tahap ini dilakukan implementasi rancangan sistem yang telah dibuat. Dalam tahap ini akan direalisasikan apa yang sudah menjadi rancangan sistem sehingga menjadi aplikasi yang sesuai dengan apa yang sudah direncanakan dan dibutuhkan pihak yang terkait.

### 5. Uji coba dan evaluasi

Pada tahap ini, dilakukan uji coba terhadap sistem yang telah dibuat kemudian dilakukan perbaikan apabila terdapat kesalahan sehingga dapat dilakukan evaluasi terhadap hasil uji coba tersebut.

### 6. Penyusunan Laporan Penelitian

Pada tahap ini merupakan penyusunan laporan yang memuat dokumentasi mengenai pembuatan aplikasi serta hasil dari implementasi perangkat lunak yang telah dibuat.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 *Datawarehouse*

*Datawarehouse* adalah sekumpulan informasi yang disimpan dalam basis data yang digunakan untuk mendukung pengambilan keputusan dalam sebuah organisasi. Data dikumpulkan dari berbagai aplikasi yang telah ada. Data yang telah dikumpulkan tersebut kemudian divalidasi dan direstrukturisasi lagi, untuk selanjutnya di simpan dalam *datawarehouse*. Pengumpulan data ini memungkinkan para pengambil keputusan untuk pergi hanya ke satu tempat untuk mengakses seluruh data yang ada tentang organisasinya [WAY-00].

*Datawarehouse* merupakan *database* relasional yang didesain lebih kepada *query* dan analisa dari pada proses transaksi, biasanya mengandung *history* data dari proses transaksi dan bisa juga data dari sumber lainnya. *Datawarehouse* memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan organisasi menggabung/konsolidasi data dari berbagai macam sumber [PAU-02].

Menurut [HAN-01], dalam membangun sebuah *datawarehouse*, informasi yang diambil dari berbagai *processing system* harus relevan, dalam periode waktu yang konsisten dan tidak mengalami perubahan secara cepat. Keuntungan yang diberikan sehubungan dengan adanya *datawarehouse* antara lain :

- a. Memberikan *competitive advantages* dengan menampilkan informasi yang relevan dalam mengukur *performance* dan membuat keputusan kritis untuk menghadapi persaingan dengan kompetitor.
- b. Meningkatkan produktivitas bisnis dikarenakan *datawarehouse* secara cepat dan mudah mengumpulkan informasi secara tepat.
- c. Memberikan fasilitas *Customer Relationship Management* (CRM) karena *datawarehouse* mampu memberikan pandangan yang konsisten tentang customer dan barang yang dimiliki perusahaan.

## 2.2 Karakteristik *Datawarehouse*

Menurut [INM-02], Karakteristik *Datawarehouse* adalah :

### 1. *Subject oriented*

Karakteristik pertama dari *datawarehouse* adalah berorientasi pada subjek yang memiliki makna bahwa *datawarehouse* berfokus pada masing-masing subjek yang terdapat dalam suatu perusahaan, bukan berdasarkan proses sehari – hari yang dilakukan perusahaan. Untuk memudahkan pemahaman, dapat dilihat pada tabel perbandingan antara subjek pada *datawarehouse* dengan subjek yang ada pada proses sehari-hari atau kegiatan operasional.

**Tabel 2.1** Perbedaan *Datawarehouse* dengan Data Operasional

Perbedaan	<i>Datawarehouse</i>	Data Operasional
Maksud Perancangan	Untuk kegiatan analisis perusahaan.	Untuk kegiatan operasional perusahaan.
Lingkungan perancangan	Pada subjek utama.	Pada proses yang dilakukan perusahaan.
Data yang disimpan	Data analisis dari kumpulan data operasional dan bersifat statis.	Data detil dari hasil kegiatan perusahaan dan bersifat dinamis.
Pengguna	Managerial user dalam jumlah yang relatif sedikit.	Operasional user dalam jumlah yang relatif banyak.

### 2. *Time Variant*

Dalam pembuatan *datawarehouse*, dimensi waktu adalah suatu hal yang penting, dimana rentang waktu yang digunakan merupakan rentang waktu yang panjang atau bentuk variasi waktu. Horizon waktu yang digunakan dapat berkisar antara 5-10 tahun dan semakin banyak data yang kita miliki, semakin baik kita menganalisa masalah dalam perusahaan kita. Karena

dimensi ini merupakan hal yang penting maka pada bagian waktu diperlukan suatu pembagian dengan aturan-aturan tertentu seperti per bulan, per kuartal, per tahun dan sebagainya.

### 3. *Integrated*

Sifat ini berarti bahwa data yang digunakan dalam pembuatan *datawarehouse* harus data yang terpadu atau terintegrasi, sehingga tercipta konsistensi yang tinggi pada *datawarehouse*. Konsistensi yang ditunjukkan *datawarehouse* dapat dilihat pada :

#### a. *Encoding* (pengkodean)

Sebagai contoh, *software developer* harus memberi kode “m” untuk jenis kelamin pria, “f” untuk jenis kelamin wanita. Dapat juga memberi kode “1” atau “male” untuk pria serta “0” atau “female” untuk wanita.

#### b. *Attribute Measurement* (pengukuran atribut)

Sebagai contoh, ada beberapa satuan ukuran yang digunakan untuk satuan panjang dalam database seperti cm, inchi, meter, dan yard. Dengan karakteristik integrasi data, maka ukuran tersebut harus konsisten seperti menetapkan ukuran satuan panjang yaitu cm.

#### c. *Multiple Source* (banyak sumber)

Dalam *database* ada kemungkinan banyaknya deskripsi dari suatu informasi, namun dengan prinsip integrasi data, seluruh informasi tersebut harus memiliki kesamaan deskripsi yang konsisten.

#### d. *Confilcting Keys* (kunci yang berbeda)

Sebagai contoh, dalam *database* ada beberapa tipe data yang berbeda dalam *field* yang sama seperti *field* kode barang dalam tabel penjualan memiliki tipe data character (char) dengan *field size* 10 sedangkan dalam tabel lain berbeda, misal char (12). Semua perbedaan itu harus dintegrasikan menjadi satu tipe data yaitu char dengan ukuran 12.

#### 4. *Non Volatile*

Data pada *datawarehouse* tidak dapat mengalami perubahan, lain halnya pada database operasional dimana dapat dilakukan operasi update, insert, dan delete terhadap data yang menyebabkan perubahan isi pada *database* namun pada *datawarehouse* hanya ada dua kegiatan memanipulasi data yaitu *loading* data (mengambil data yang dibutuhkan data warehouse) dan akses data (proses mengakses *datawarehouse*, seperti melakukan *query* atau menampilkan laporan yang dibutuhkan), tidak ada kegiatan update data.

### 2.3 Struktur *Datawarehouse*

Metadata merupakan suatu bentuk informasi yang berisi data yang akan digunakan, tipe data, panjang datanya serta sumber datanya yang akan digunakan dalam *datawarehouse*. Metadata adalah data tentang data yang berguna sebagai pusat penyimpanan informasi untuk menjelaskan kepada user tentang apakah *datawarehouse* itu, dari mana asalnya dan siapa yang bertanggung jawab atas hal tersebut dan sebagainya [MAL-02].

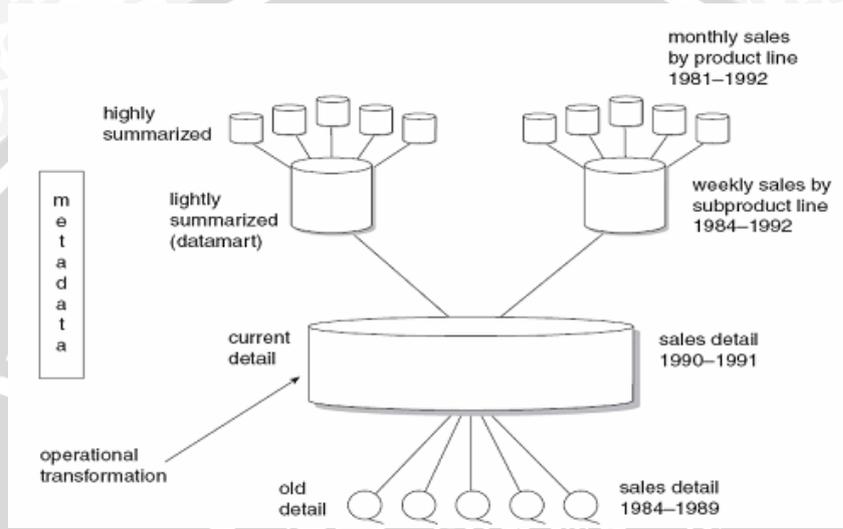
Metadata digunakan untuk berbagai tujuan meliputi [CON-05]:

- a. Proses ekstraksi dan *loading*. Metadata digunakan untuk memetakan sumber data ke dalam pandangan umum dari data dalam warehouse.
- b. Proses manajemen *warehouse*. Metadata digunakan untuk mengotomatiskan pembuatan tabel ringkasan.
- c. Sebagai bagian dari proses manajemen *query*. Metadata digunakan untuk menghubungkan suatu *query* dengan sumber data yang tepat.

Metadata dalam *datawarehouse* dibagi menjadi 3 kategori, yaitu [PON-01]:

- a. Metadata operasional, berisi mengenai informasi tentang sumber data operasional yang memiliki struktur data yang berbeda, ukuran *field* yang berbeda, dan tipe data yang berbeda.

- b. Metadata ekstraksi dan transformasi, berisi mengenai data ekstraksi dari sumber data, penamaan, frekuensi ekstraksi, metode ekstraksi, dan peraturan untuk ekstraksi.
- c. Metadata pengguna akhir, adalah sebuah peta navigasi dari *datawarehouse*. Ini memampukan pengguna akhir untuk menemukan informasi dari *datawarehouse*.



**Gambar 2.1** Struktur Datawarehouse [INM-02]

## 2.4 OLAP (*Online Analytical Processing*)

*Online Analytical Processing* (OLAP) merupakan salah satu *tools* yang digunakan untuk mengakses informasi dalam datawarehouse. Teknologi OLAP memungkinkan datawarehouse digunakan secara efektif untuk proses *online analysis*, memberikan respon yang cepat terhadap *analytical queries* yang kompleks. Multidimensional data model dan teknik agregasi data yang dimiliki oleh OLAP dapat mengatur dan membuat kesimpulan dari data dalam jumlah besar, sehingga dapat dievaluasi secara cepat dengan menggunakan *online analysis* dan *graphical tool*. Sistem OLAP menyediakan kecepatan dan fleksibilitas untuk melakukan *support* analisis secara *real time*. [WIL-02].

Menurut [PON-01] Karakteristik utama dari OLAP, meliputi:

- a. Mendukung pemanfaatan *datawarehouse* yang memiliki data multidimensional.

- b. Menyediakan fasilitas *query interaktif* dan analisis yang kompleks.
- c. Menyediakan fasilitas *drill-down* untuk memperoleh informasi yang rinci, dan *roll-up* untuk memperoleh agregat dalam multidimensi.
- d. Mampu menghasilkan perhitungan dan perbandingan.
- e. Menyajikan hasil dalam angka yang mudah dimengerti, maupun penyajian grafik.

OLAP digunakan untuk memproses informasi dan menampilkannya dalam bentuk multidimensi. Walaupun data yang ditampilkan dalam satu bentuk namun tidak berarti data-data yang ada disimpan dalam satu bentuk pula. Ada tiga teknik dasar untuk menyimpan data OLAP [GOL-09]:

1. *Multidimensional* OLAP (MOLAP)

Salah satu cara umum yang digunakan untuk menyimpan data ialah dalam vasis data multidimensional. Tidak seperti basis data relasional yang menyimpan informasi dalam sejumlah baris dalam tabel, sebuah basis data multidimensional menyimpan informasi dalam sejumlah *array* multidimensional. Karena dimensi dapat diakses secara mudah, maka *user* dapat melakukan *query* pada basis data MOLAP dengan sangat cepat. Selain mengandung data mentah, basis data MOLAP juga mengandung agregasi data sehingga dapat memberikan respon yang cepat terhadap *query*.

2. *Relational* OLAP (ROLAP)

Partisi menggunakan tabel relational dalam datawarehouse untuk menyimpan agregasi, sedangkan detail dari tabel fakta tetap tersimpan dalam datawarehouse. Pada sebuah basis data yang terdiri dari ribuan atau ratusan ribu *records* maka menampilkan informasi OLAP akan menjadi sebuah masalah karena banyak data yang harus dihapus di *query*. Dan hal inilah yang menjadi keterbatasan partisi ROLAP.

3. *Hybrid* OLAP (HOLAP)

Partisi menggunakan struktur MOLAP untuk menyimpan agregasi dan meninggalkan detail dari *fact table* dalam partisi ROLAP. OLAP service dapat menggunakan data MOLAP dan ROLAP secara simultan untuk memecahkan suatu *query*.

## 2.5 ETL (*Extraction, Transformation, dan Loading*)

Tiga fungsi utama yang perlu dilakukan untuk membuat data siap digunakan pada datawarehouse adalah *extraction*, *transformation*, dan *loading*. Ketiga fungsi ini terdapat pada *staging area*. Pada data *staging* ini, disediakan tempat dan area dengan beberapa fungsi seperti data *cleansing*, *change*, *convert*, dan menyiapkan data untuk disimpan serta digunakan oleh datawarehouse. Berikut ini merupakan penjelasan dari proses ETL dalam Datawarehouse [ANO-10].

### 1. *Extraction*

Data Extraction adalah proses pengambilan data yang diperlukan dari sumber datawarehouse dan selanjutnya dimasukkan pada *staging area* untuk diproses pada tahap berikutnya. Pada fungsi ini, kita akan banyak berhubungan dengan berbagai tipe sumberdata. Format data, mesin yang berbeda, software dan arsitektur yang tidak sama. Sehingga sebelum proses ini kita lakukan, sebaiknya perlu kita definisikan *requirement* terhadap sumber data yang akan kita butuhkan untuk lebih memudahkan pada *extraction* data ini.

### 2. *Transformation*

Pada kenyataannya, pada proses transaksional data disimpan dalam berbagai format sehingga jarang kita temui data yang konsisten antara aplikasi-aplikasi yang ada. Transformasi data ditujukan untuk mengatasi masalah ini. Dengan proses transformasi data ini, kita melakukan standarisasi terhadap data pada satu format yang konsisten. Beberapa contoh ketidak konsistenan data tersebut dapat diakibatkan oleh tipe data yang berbeda, *data length* dan lain sebagainya. Untuk lebih jelasnya dapat dilihat pada tabel 2.2 yang menunjukkan permasalahan dasar pada proses pengintegrasian data.

**Tabel 2.2** Isi Tentang Transformasi Data

	<b>Sales</b>	<b>Inventori</b>	<b>Trasaksi Penjualan</b>
<i>Format</i>	Key Text	Key Integer	Key Yes/No

<i>Description</i>	Nama pelanggan U.N.I.J.O.Y.O	Nama pelanggan UNJOYO	Nama pelanggan Universitas Trunojoyo
<i>Unit</i>	Tinggi centimeter	Tinggi meter	Tinggi inch
<i>Encoding</i>	Sex Yes = Laki-laki No = Perempuan	Sex L = Laki-laki P = Perempuan	Sex 1 = Laki-laki 0 = Perempuan

Keterangan :

*Format*, Pada transaksional data dapat disimpan dalam berbagai format. Elemen data tersebut dapat tersimpan dalam format text, integer dan sebagainya. Untuk itu standarisasi perlu dilakukan dengan melihat kegunaan pokok dari elemen data pada proses transaksional dan datawarehouse.

*Description*, Pada tabel terlihat representasi ketiga nama pelanggan adalah sama. Namun dengan penulisan yang beda terlihat adanya perbedaan format pada data. Oleh karena itu perlu diambil salah satu dari deskripsi tersebut untuk konsistensi data.

*Unit*, Adanya perbedaan satuan ukuran dapat menimbulkan permasalahan yang kompleks. Jika user tidak mengetahui adanya perbedaan ini dan menganggap sama, maka akan terjadi kesalahan ketika kita melakukan penghitungan matematis.

*Encoding*, huruf atau nomor dapat dijadikan label sebagai identifikasi suatu objek. Seperti pada tabel diatas, kesalahan dapat terjadi karenanya.

### 3. *Loading*

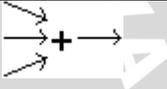
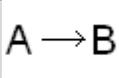
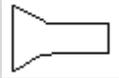
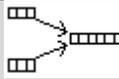
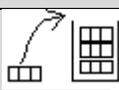
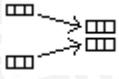
Data *loading* adalah memindahkan data ke datawarehouse. Ada dua loading data yang kita lakukan pada datawarehouse. Pertama adalah *initial load*, proses ini dilakukan pada saat kita telah selesai mendesign dan membangun datawarehouse. Kedua *Incremental load*, dilakukan ketika datawarehouse telah dioperasikan. Kita melakukan data *extraction*, *transformation* dan *loading* terhadap data tersebut.

Tahap *loading* adalah men-load data ke dalam target akhir, yang pada umumnya adalah datawarehouse. Pemilihan waktu dan lingkup untuk

menggantikan atau menambahkan aneka pilihan desain strategi bergantung pada waktu yang tersedia dan kebutuhan bisnis tersebut. Kebanyakan sistem yang kompleks dapat memelihara suatu histori dan jejak audit dari semua perubahan yang ada ke data yang di-load ke dalam datawarehouse.

Dalam perancangan proses ETL, agar mekanisme lebih mudah dipahami maka digambarkan dengan suatu UML (*Unified Modeling Language*). UML tersebut akan menjelaskan mekanisme apa saja yang dilakukan pada tahapan ETL. Tabel 2.3 menunjukkan jenis-jenis mekanisme ETL beserta simbol dan keterangannya untuk masing-masing proses [TRU-03].

**Tabel 2.3** Mekanisme ETL [TRU-03]

Mekanisme ETL	Deskripsi	Icon
<i>Aggregation</i>	Agregat data berdasarkan beberapa kriteria	
<i>Conversion</i>	Merubah tipe data dan <i>format</i> dari data yang sudah ada	
<i>Filter</i>	Mem- <i>filter</i> atau menyaring dan memverifikasi data	
<i>Incorrect</i>	Mengganti data yang tidak benar atau tidak <i>valid</i>	
<i>Join</i>	Menggabungkan dua sumber data yang terkait satu sama lain dengan beberapa atribut	
<i>Loader</i>	Memasukkan data ke target dari proses ETL	
<i>Log</i>	Mencatat aktivitas dari mekanisme ETL	
<i>Merge</i>	Mengintegrasikan dua atau lebih sumber data dengan atribut yang kompatibel	

<i>Surrogate</i>	Menghasilkan kunci pengganti yang unik	123 →
<i>Wrapper</i>	Merubah sumber data asli menjadi sumber data berbasis <i>record</i>	☐☐ → ☐☐

## 2.6 Model Data Multidimensi

Pembuatan *datawarehouse* didasarkan pada model data multidimensi. Model ini menampilkan data dalam bentuk kubus. Model data multidimensi terdiri dari dimensi (*dimensions*) dan fakta (*facts*) Objek utama yang disimpan dalam sebuah basis data OLAP ialah *cube* (kubus). Sebuah kubus merupakan representasi multidimensi dari sekumpulan data, yang mengandung data secara detail maupun rangkumannya. Sebuah basis data OLAP dapat memiliki beberapa buah kubus sesuai dengan yang dibutuhkan, yang menggambarkan data yang ada dalam *datawarehouse*. Sebuah kubus dibangun menggunakan dua komponen utama yaitu ukuran (*measure*) dan dimensi. Ukuran merupakan nilai numerik dari *fact table* dalam *datawarehouse* seperti harga dari unit maupun kualitas dari item. Sedangkan dimensi menggambarkan kategori dari ukuran yang ada, seperti bagaimana ukuran berubah setiap waktu. Beberapa operasi yang dapat dilakukan pada kubus atau report multidimensi ini adalah [HAN-06]:

### 1. *Processing*

Operasi *Processing* pada kubus antara lain melakukan *loading* dan *refreshing* data pada kubus yang dilakukan oleh OLAP *service engine*. *Processing* akan dilakukan terhadap kubus hanya jika terjadi perubahan terhadap struktur kubus tersebut atau terjadi perubahan terhadap data yang ada dalam *datawarehouse*. Ketika akan melakukan *processing* terhadap kubus, terdapat dua pilihan. Pertama dengan melakukan *refresh* terhadap kubus secara total, yang berarti menghapus seluruh data yang ada sebelumnya dan mengambil data yang baru dari *datawarehouse*. Dan yang kedua dengan melakukan *incremental update*

yang berarti mengambil data terbaru yang ada dalam datawarehouse dan menambahkannya pada data dalam kubus. Pilihan kedua akan memakan waktu yang jauh lebih singkat, namun *incremental update* tidak dapat dilakukan jika struktur kubus telah mengalami perubahan.

2. *Slice And Dice*

*Slice and Dice* memungkinkan untuk melihat kubus dari sudut pandang yang berbeda. Dengan *slice and dice* dapat ditentukan dimensi apa yang hendak ditampilkan dan bagaimana mereka ditampilkan. Hal inilah yang menjadi keunggulan OLAP. Dengan melihat kubus dari sudut pandang yang berbeda maka akan dapat dipelajari banyak hal dari data yang dimiliki.

3. *Drill Down*

Sebagian besar informasi yang ditampilkan dalam OLAP merepresentasikan kesimpulan yang lebih detail. *Drilling down* merupakan teknik untuk memecahkan sebuah informasi menjadi beberapa informasi yang lebih detail. Sebagai contoh, jika dilakukan *drill down* terhadap data tahunan maka akan dapat diperoleh data dalam catur wulan atau tri wulan.

4. *Consolidation (Roll Up)*

*Consolidation* atau yang lebih dikenal *roll up* merupakan kebalikan dari *Drill Down*. Data-data sebelumnya dilihat dalam format triwulan akan dapat dilihat dalam format tahunan. Dengan *roll up* data dapat dilihat secara lebih global.

5. *Pivotting*

*Pivotting* merupakan suatu teknik untuk saling menukarkan dimensi data. Dengan melakukan pivotting maka dapat diamati suatu informasi atau data dari sudut pandang yang berbeda, sehingga diharapkan akan dapat memperjelas analisis yang dilakukan.

## 6. *Filtering*

*Filtering* merupakan suatu teknik untuk menyaring informasi yang dibutuhkan, sehingga para analisis tidak dibingungkan oleh banyaknya informasi yang tersedia.

### 2.6.1 Tabel Fakta dan Tabel Dimensi

Tabel fakta merupakan fondasi dari datawarehouse. Tabel fakta mengandung ukuran fundamental dari perusahaan, dan ia merupakan target utama dari kebanyakan *query datawarehouse* [KIM-04].

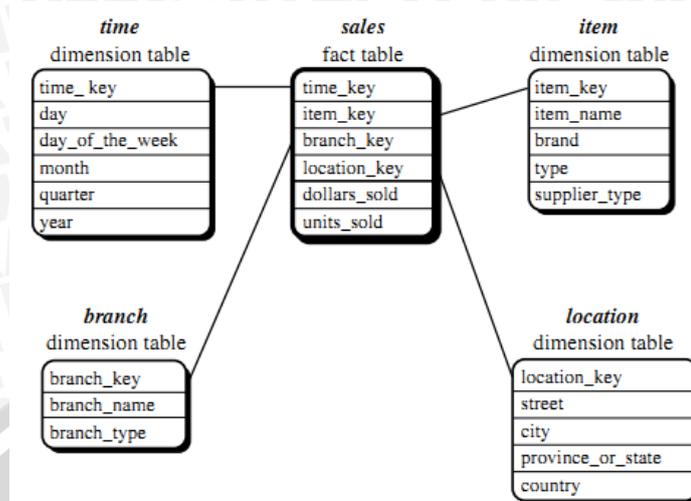
Tabel Fakta merupakan sebuah tabel yang memiliki sebuah *composit primary key* dimana tabel tersebut akan membentuk sebuah model dimensional. Tabel dimensi merupakan sekumpulan dari tabel-tabel yang lebih kecil yang memiliki sebuah *primary key* sederhana yang merespon secara benar terhadap salah satu komponen dari *composit key* yang ada pada tabel fakta [CON-05].

### 2.6.2 Skema Model Data Multidimensi

Pada *datawarehouse* sekumpulan tabel yang saling berhubungan dinamakan skema. Skema digunakan untuk menunjukkan hubungan antar tabel dimensi dengan tabel fakta. Tipe-tipe skema multidimensi antara lain [HAN-06]:

#### 1. *Star Scheme* (Skema Bintang)

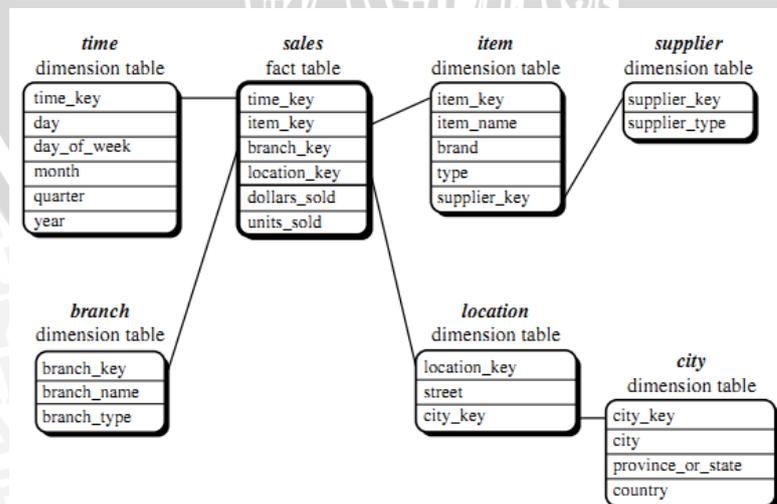
Skema yang paling umum digunakan pada *datawarehouse* adalah skema bintang. Skema bintang merupakan struktur logikan yang memiliki tabel fakta di tengah dan dikelilingi oleh tabel dimensi sebagai referensi data. Keuntungan dari skema bintang adalah peningkatan kinerja *datawarehouse*, pemrosesan *query* yang lebih efisien, dan waktu respon yang cepat. Bentuk skema bintang dapat dilihat pada Gambar 2.2.



Gambar 2.2 Star Scheme [HAN-06]

2. Snowflake Scheme (Skema Kepingan Salju)

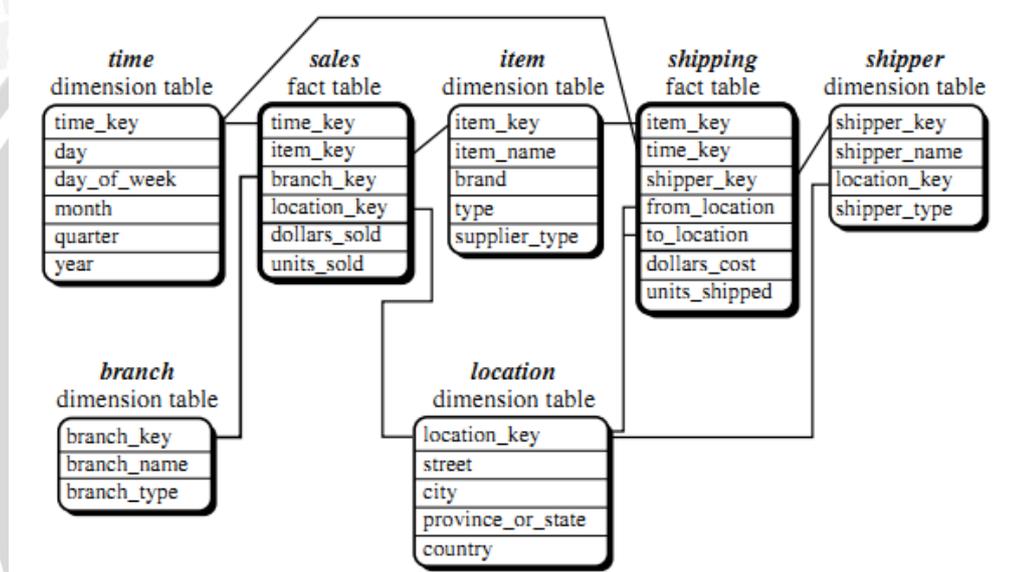
Skema kepingan salju (*Snowflake Scheme*) adalah variasi dari skema bintang yang beberapa tabel dimensi dinormalisasi sehingga membagi data kedalam tabel tambahan. Perbedaan utama antara skema kepingan salju dengan skema bintang adalah tabel dimensi dari model kepingan salju dapat disimpan dalam bentuk normalisasi untuk mengurangi redudansi. Selanjutnya struktur kepingan salju dapat menghemat penggunaan memori, tapi waktu yang dibutuhkan untuk pemrosesan *query* menjadi lebih lama akibat kinerja sistem dapat berdampak buruk. Bentuk skema kepingan salju dapat dilihat pada Gambar 2.3.



Gambar 2.3 Snowflake Scheme [HAN-06]

### 3. Galaxy Scheme (Skema Galaksi)

Skema galaksi dapat dikatakan sebagai kumpulan dari skema bintang atau bisa disebut gabungan dari skema bintang dan skema kepingan salju. Skema ini biasanya digunakan pada aplikasi yang canggih. Keuntungan menggunakan skema ini adalah menghemat memori dan mengurangi kesalahan yang mungkin terjadi. Ciri utama skema galaksi ini adalah mempunyai lebih dari satu tabel fakta dan tabel dimensi bisa berelasi dengan tabel dimensi lain. Bentuk skema galaksi dapat dilihat pada Gambar 2.4.



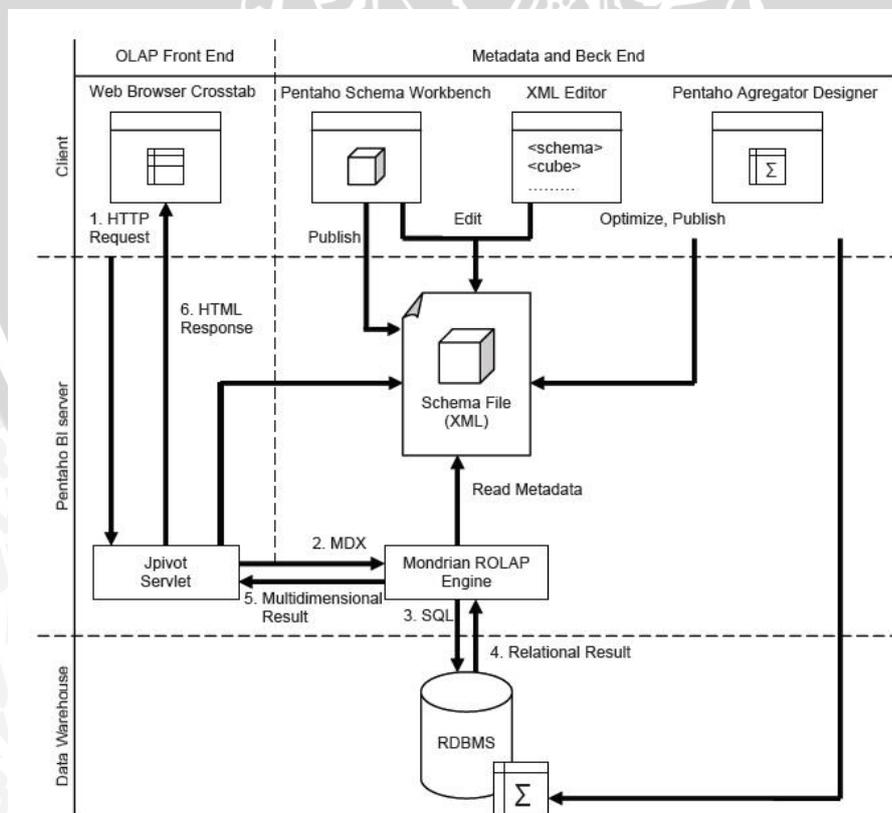
Gambar 2.4 Galaxy Scheme [HAN-06]

### 2.7 Arsitektur Komponen OLAP Pentaho

Mondrian merupakan aplikasi *server* OLAP berbasis *open source* yang sangat populer dan saat ini bernaung di bawah Pentaho Corporation. Proyek Mondrian ini merupakan kombinasi yang sangat baik dengan JPivot - *interface* berbasis web yang juga merupakan proyek *open source*. Pada gambar 2.5 dijelaskan urutan peristiwa yang terjadi ketika pengguna menggunakan aplikasi Pentaho OLAP [BOU-09].

1. Pengguna web browser menggunakan sebuah HTTP *request* untuk melihat, *browse* atau *drill down* kedalam tabel pivot OLAP.

2. JPivot servlet menerima *request* dari pengguna dan menerjemahkan *request* tersebut kedalam sebuah MDX *query*. MDX *query* ini kemudian dikirimkan ke Mondrian *engine*.
3. Mondrian menginterpretasikan MDX *query* dan menerjemahkannya kedalam satu atau lebih SQL *query*. Teknik khusus ini disebut sebagai ROLAP, yang merupakan singkatan dari *Relational OLAP*.
4. *Relational Database Management System* (RDBMS) mengeksekusi *query* yang diberikan oleh Mondrian. Mondrian menerima hasil *query* dalam bentuk tabular (*relational*).
5. Mondrian memproses hasil yang diterima dari RDBMS dan menerjemahkannya kedalam result-set multidimensi. Ini sebenarnya adalah hasil query MDX dari query MDX yang dikirim ke Mondrian pada langkah 2.
6. JPivot menggunakan hasil multidimensi untuk membuat halaman HTML menampilkan data. Inilah yang kemudian dikirim ke browser untuk ditunjukkan kepada pengguna.



**Gambar 2.5** Gambaran Komponen OLAP Pentaho [BOU-09]

## 2.8 Sistem Pendukung Keputusan

Definisi sistem pendukung keputusan (SPK) atau *Decision Support System* (DSS) adalah sebuah sistem yang digunakan sebagai alat bantu menyelesaikan masalah untuk membantu pengambil keputusan (manajer) dalam menentukan keputusan, tetapi tidak untuk menggantikan kapasitas manajer, hanya memberikan pertimbangan. DSS ditujukan untuk keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma [TUR-05]. Selain itu, Sistem pendukung keputusan mendayagunakan *resources* individu-individu secara intelek dengan kemampuan komputer untuk meningkatkan kualitas keputusan. Jadi ini merupakan sistem pendukung keputusan yang berhubungan dengan masalah-masalah semi terstruktur [SUB-02].

Adapun karakteristik Sistem Pendukung Keputusan menurut [KOS-02], antara lain :

1. Mendukung seluruh kegiatan organisasi.
2. Mendukung beberapa keputusan yang saling berinteraksi.
3. Dapat digunakan berulang kali dan bersifat konstan.
4. Terdapat dua komponen utama, yaitu data dan model.
5. Menggunakan baik data internal maupun eksternal.
6. Memiliki kemampuan *what-if analysis* dan *goal seeking analysis*  
Menggunakan beberapa model kuantitatif.

## 2.9 Logika Fuzzy

FDSS (*Fuzzy Decision Support System*) atau SPKF (Sistem Pendukung Keputusan dengan logika Fuzzy) merupakan sistem pembuat keputusan manusia-komputer untuk mendukung keputusan manajerial, dan intuisi untuk memecahkan masalah manajerial dengan memberikan informasi yang diperlukan, menghasilkan, mengevaluasi dan memberikan putusan alternatif [NAD-03].

Logika *Fuzzy* memiliki beberapa karakteristik yaitu himpunan *Fuzzy* dan fungsi keanggotaan. Pada logika *boolean*, sebuah individu dipastikan sebagai anggota dari salah satu himpunan saja, sedangkan pada himpunan *fuzzy* sebuah individu dapat masuk pada dua himpunan yang berbeda. Seberapa besar

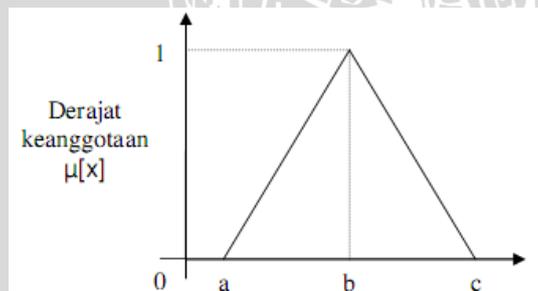
eksistensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaannya [CRI-05].

Menurut [HEL-01], Himpunan *fuzzy*  $A$  pada semesta  $X$  dinyatakan sebagai himpunan pasangan berurutan (*set of ordered pairs*) baik diskrit maupun kontinu.

$$A = \{(x, \mu_A(x)) \mid x \in X\} \quad (2.1)$$

Dimana  $\mu_A(x)$  fungsi keanggotaan himpunan *fuzzy*  $A$ . Fungsi keanggotaan memetakan setiap  $x \in X$  pada suatu nilai antara  $[0,1]$  yang disebut derajat keanggotaan (*membership grade* atau *membership value*).

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (disebut juga dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Derajat keanggotaan dalam himpunan (*degree of membership*) dilambangkan dengan  $\mu$ . Dalam kasus yang dibahas, fungsi keanggotaan yang dipakai adalah Representasi Kurva Segitiga dan Representasi Kurva Bahu. Representasi Kurva Segitiga pada dasarnya merupakan gabungan antara dua garis.

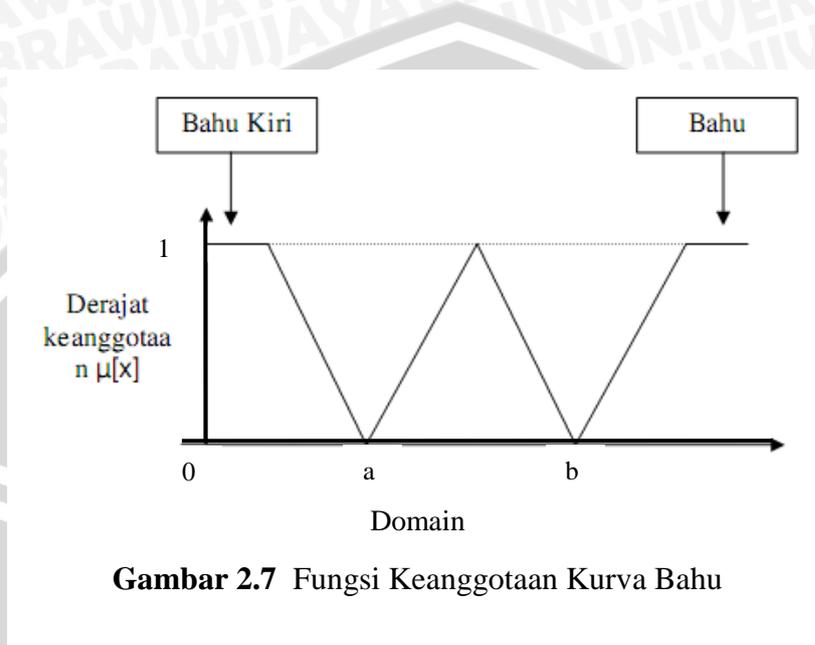


**Gambar 2.6** Fungsi Keanggotaan Kurva Segitiga

Fungsi Keanggotaan :

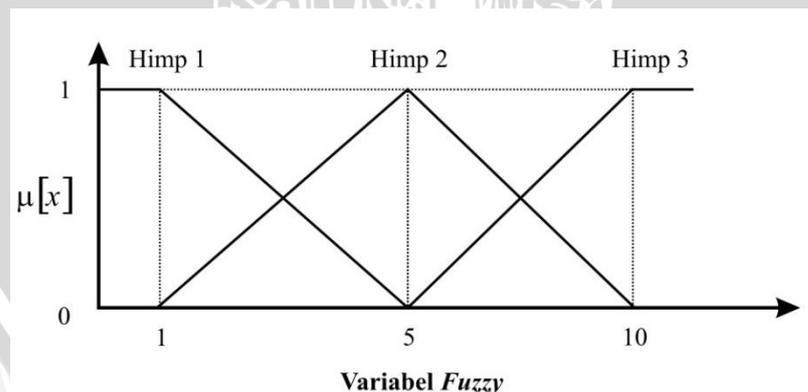
$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}; & a < x \leq b \\ \frac{c-x}{c-b}; & b < x < c \end{cases}$$

Sedangkan representasi kurva bahu merupakan daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan sisi kirinya akan naik dan turun. Himpunan *fuzzy* bahu digunakan untuk mengakhiri variabel suatu daerah *fuzzy*.



**Gambar 2.7** Fungsi Keanggotaan Kurva Bahu

Fungsi keanggotaan pada kurva segitiga dan fungsi keanggotaan pada kurva bahu dapat dilihat pada gambar 2.8



**Gambar 2.8** Himpunan Fuzzy Untuk Suatu Variabel

Akan dilakukan perhitungan untuk memperoleh nilai *fire strength fuzzy* dengan menggunakan gabungan fungsi keanggotaan kurva segitiga dan kurva bahu. Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan

fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau  $\alpha$ -predikat. Operator AND merupakan operator yang berhubungan dengan operasi interseksi pada himpunan. *Firestrength* sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y]) \quad (2.2)$$

Operator OR merupakan operator yang berhubungan dengan operasi union pada himpunan. *Fire strength* sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antara elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[y]) \quad (2.3)$$

Operator ini berhubungan dengan operasi komplemen pada himpunan. *Fire strength* sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

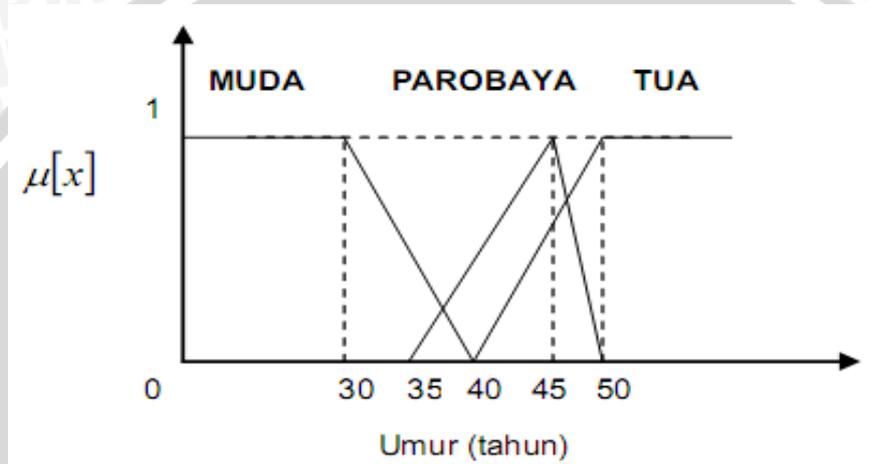
$$\mu_{A^1} = 1 - \mu_A[x] \quad (2.4)$$

## 2.10 Fuzzy Database

Menurut [KUS-04] *Database system* adalah suatu sistem informasi yang mengintegrasikan kumpulan data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi dalam suatu organisasi. Sebagian besar database standar diklasifikasikan berdasarkan bagaimana data tersebut dipandang oleh user. Dengan menggunakan database standar, kita dapat mencari data-data dengan spesifikasi tertentu dengan menggunakan query. Namun pada kenyataannya, kadang kita membutuhkan informasi dari data-data yang bersifat ambiguous. Apabila hal ini terjadi, maka kita bisa menggunakan *fuzzy database*. Salah satu diantara model sistem *fuzzy database* adalah model Tahani.

Fuzzy database model Tahani masih tetap menggunakan relasi standar, hanya saja model ini menggunakan teori himpunan fuzzy untuk mendapatkan informasi query-nya.

Misalkan terdapat sebuah contoh variabel usia karyawan yang kemudian dapat dikategorikan ke dalam himpunan : MUDA, PAROBAYA, dan TUA (Gambar 2.8)



Gambar 2.9 Fungsi Keanggotaan untuk Variabel Usia

Fungsi Keanggotaan :

$$\mu_{MUDA}[x] = \begin{cases} 1; & x \leq 30 \\ \frac{40-x}{10}; & 30 \leq x \leq 40 \\ 0; & x \geq 40 \end{cases}$$

$$\mu_{TUA}[x] = \begin{cases} 0; & x \leq 40 \\ \frac{x-40}{10}; & 40 \leq x \leq 50 \\ 1; & x \geq 50 \end{cases}$$

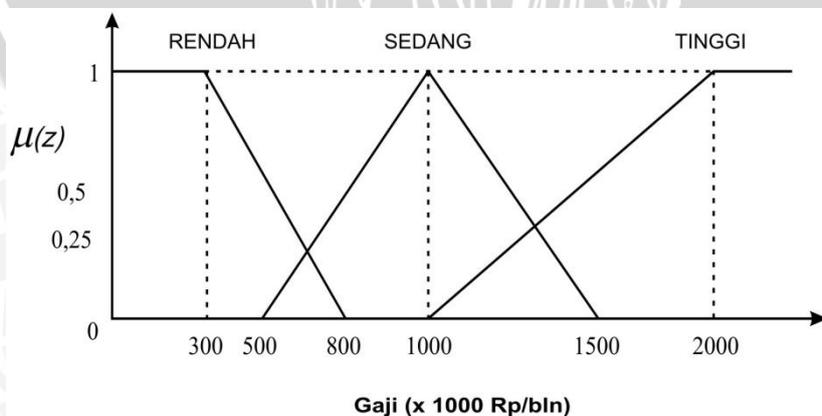
$$\mu_{\text{PAROBAYA}} [x] = \begin{cases} 0; & x \leq 35 \text{ atau } x \geq 50 \\ \frac{x-35}{10}; & 35 \leq x \leq 45 \\ \frac{50-x}{5}; & 45 \leq x \leq 50 \end{cases}$$

Tabel karyawan berdasarkan umur dengan derajat keanggotaannya pada setiap himpunan dapat dilihat pada Tabel 2.4.

**Tabel 2.4** Karyawan Berdasarkan Umur

No.	Nama	Umur	Derajat Keanggotaan ( $\mu(x)$ )		
			MUDA	PAROBAYA	TUA
01	Lia	30	1	0	0
02	Iwan	48	0	0,4	0,8
03	Sari	36	0,4	0,1	0
04	Andi	37	0,3	0,2	0
05	Budi	42	0	0,7	0,2
06	Amir	39	0,1	0,4	0
07	Rian	37	0,3	0,2	0
08	Kiki	32	0,8	0	0
09	Alda	35	0,5	0	0
10	Yoga	25	1	0	0

Selain itu terdapat juga sebuah contoh variabel gaji karyawan yang kemudian dapat dikategorikan ke dalam himpunan RENDAH, SEDANG, dan TINGGI (Gambar 2.9)



**Gambar 2.10** Fungsi Keanggotaan untuk Variabel Gaji

Fungsi Keanggotaan :

$$\mu_{RENDAH} [z] = \begin{cases} 1; & z \leq 300 \\ \frac{800 - z}{500}; & 300 \leq z \leq 800 \\ 0; & z \geq 800 \end{cases}$$

$$\mu_{SEDANG} [z] = \begin{cases} 0; & z \leq 500 \text{ atau } z \geq 1500 \\ \frac{z - 500}{500}; & 500 \leq z \leq 1000 \\ \frac{1500 - z}{500}; & 1000 \leq z \leq 1500 \end{cases}$$

$$\mu_{TINGGI} [z] = \begin{cases} 0; & z \leq 1000 \\ \frac{z - 1000}{1000}; & 1000 \leq z \leq 2000 \\ 1; & z \geq 2000 \end{cases}$$

Tabel karyawan berdasarkan gaji dengan derajat keanggotaannya pada setiap himpunan dapat dilihat pada Tabel 2.5.

**Tabel 2.5** Karyawan Berdasarkan Gaji

No.	Nama	Gaji	Derajat Keanggotaan ( $[x]$ )		
			RENDAH	SEDANG	TINGGI
01	Lia	750.000	0,1	0,50	0
02	Iwan	1.500.000	0	0,49	0,255
03	Sari	1.255.000	0	0	0,500
04	Andi	1.040.000	0	0,92	0,040
05	Budi	950.000	0	0,90	0
06	Amir	1.600.000	0	0	0,600
07	Rian	1.250.000	0	0,50	0,250
08	Kiki	550.000	0,50	0	0
09	Alda	735.000	0,13	0	0
10	Yoga	860.000	0	0	0

Dari penentuan Fungsi Keanggotaan tersebut, terdapat beberapa *query* yang bisa diberikan dari data di atas, selain itu juga dapat dicari *Firestrength* dari hasil

query tersebut, *firestrength* ini merupakan nilai hasil pengoperasian dari *query* yang telah dibuat dan akan digunakan sebagai nilai akhir dalam proses perhitungan *fuzzy*, misalkan:

*Query:*

Siapa sajakah karyawan yang masih muda tapi memiliki gaji tinggi?

```
SELECT Nama
```

```
FROM Karyawan
```

```
WHERE (Umur="MUDA") and (Gaji = "TINGGI")
```

**Tabel 2.6** Hasil *Query* dan Penentuan *Firestrength*

No.	Nama	Umur	Gaji	Derajat Keanggotaan ([x])		
				MUDA	TINGGI	<i>Fire Strength</i>
03	Sari	36	1.500.000	0,4	0,5	0,4
07	Rian	37	1.250.000	0,3	0,25	0,25
06	Amir	39	1.600.000	0,1	0,6	0,1
04	Andi	37	1.040.000	0,3	0,04	0,04
01	Lia	30	750.000	1	0	0
02	Iwan	48	1.255.000	0	0,255	0
05	Budi	42	950.000	0	0	0
08	Kiki	32	550.000	0,8	0	0
09	Alda	35	735.000	0,5	0	0
10	Yoga	25	860.000	1	0	0

### 2.11 Akurasi dan Evaluasi

Akurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (*true value* atau *reference value*). Dalam penelitian ini akurasi keputusan dihitung dari jumlah keputusan yang tepat dibagi dengan jumlah data. Tingkat akurasi dan prosentase akurasi diperoleh dengan perhitungan seperti pada persamaan 2.1 dan persamaan 2.2. [NUG-06].

$$\text{Tingkat Akurasi} = \frac{\sum \text{dataujibenar}}{\sum \text{totaldatauji}} \quad (2.5)$$

$$\text{Akurasi}(\%) = \frac{\sum \text{dataujibenar}}{\sum \text{totaldatauji}} \times 100\% \quad (2.6)$$

## 2.12 Pengujian *Black Box*

Menurut [PRE-02], Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* merupakan pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangkat lunak dan kemudian keluar dari perangkat lunak dicek apakah telah sesuai yang diharapkan.

Pengujian *black box* bukan merupakan alternative dari teknik *White-Box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white-box*. Pengujian *black box* cenderung diaplikasikan selama tahap akhir pengujian, karena pengujian *black-box* memperhatikan struktur kontrol maka perhatian berfokus pada domain informasi.

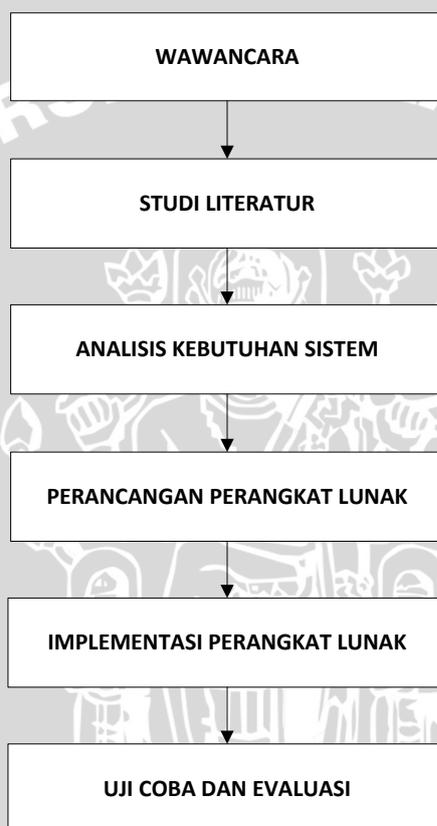
Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminas

### BAB III

## METODOLOGI DAN PERANCANGAN SISTEM

Pada bab Metodologi dan Perancangan Sistem ini berisi tentang penjelasan metode yang digunakan serta bagaimana proses perancangan sistem dalam penelitian ini. Metode penelitian yang dilakukan ditunjukkan pada gambar 3.1.



**Gambar 3.1** Langkah Penelitian

Berdasarkan pada gambar 3.1, langkah-langkah yang dilakukan dalam penelitian ini antara lain :

#### 1. Wawancara

Pada tahap awal dilakukan proses wawancara kepada pihak SMA Negeri 7 Malang mengenai parameter-parameter yang diperlukan dalam proses

seleksi penerimaan beasiswa. Selain itu, tahap ini dilakukan untuk mengetahui batasan-batasan yang ditentukan untuk setiap parameternya.

## **2. Studi Literatur**

Pada tahap ini dilakukan pemahaman kepustakaan mengenai perancangan dan pembuatan ETL, *datawarehouse*, OLAP, dan metode *Fuzzy Database*.

## **3. Analisis Kebutuhan Sistem**

Pada tahap ini dilakukan proses analisis terhadap kebutuhan yang diperlukan oleh sistem dan mengidentifikasi kebutuhan dari pengguna terhadap sistem yang akan dibuat.

## **4. Perancangan Perangkat Lunak**

Pada tahap ini dilakukan perancangan dari hasil analisis kebutuhan yang telah dilakukan, meliputi desain rancangan proses ETL (*Extract, Transform, Load*), *datawarehouse* dan teknologi OLAP yang akan dibuat, serta desain antar muka pengguna aplikasi.

## **5. Implementasi perangkat lunak**

Pada tahap ini dilakukan implementasi rancangan sistem yang telah dibuat. Dalam tahap ini akan direalisasikan apa yang sudah menjadi rancangan sistem sehingga menjadi aplikasi yang sesuai dengan apa yang sudah direncanakan dan dibutuhkan pihak yang terkait.

## **6. Uji coba dan evaluasi**

Pada tahap ini, dilakukan uji coba terhadap sistem yang telah dibuat kemudian dilakukan perbaikan apabila terdapat kesalahan sehingga dapat dilakukan evaluasi terhadap hasil uji coba tersebut.

### **3.1 Analisa Sistem**

#### **3.1.1 Deskripsi Sistem**

Aplikasi yang akan dibuat merupakan sebuah sistem aplikasi pendukung keputusan yang digunakan untuk proses rekomendasi beasiswa pada siswa. Dari database akademik siswa (data operasional) akan dibuat sebuah *datawarehouse* agar dapat memberikan tingkat analisa pada sistem dengan lebih efektif dan efisien. Sebelum membentuk sebuah *datawarehouse*, terlebih dahulu dilakukan

proses ETL (*Extract, Load, Transform*) yang digunakan untuk menganalisa sumber data (*database* akademik siswa) sehingga data-data dan atribut-atribut yang diproses hanya yang berkaitan dengan proses rekomendasi beasiswa dan merupakan parameter-parameter yang digunakan dalam proses rekomendasi. Parameter yang digunakan dalam proses rekomendasi beasiswa adalah nilai akademik siswa yang diambil dari rapor, biodata orang tua khususnya yang berkaitan dengan pekerjaan serta penghasilan kedua orang tua, serta sikap atau perilaku siswa yang berupa tingkat kedisiplinan siswa berdasarkan banyaknya pelanggaran yang dilakukan oleh siswa.

Dengan menggunakan teknik *fuzzy database* yang diproses dalam *datawarehouse*, maka proses rekomendasi beasiswa akan bisa dilakukan dengan cepat, sehingga *user* akan lebih mudah dalam menyeleksi para siswa yang berhak menerima beasiswa.

### 3.1.2 Analisa Kebutuhan Fungsional Sistem

Dari penjelasan tentang deskripsi sistem, maka dapat diketahui beberapa kebutuhan dari sistem yang sekaligus menggambarkan beberapa fungsional atau layanan yang akan diberikan sistem kepada pengguna. Kebutuhan dari sistem ini meliputi :

- a. Analisa akademik siswa yang diberikan sistem adalah analisa nilai akademik siswa dan tingkat kedisiplinan siswa.
- b. Sistem akan menampilkan laporan hasil analisa akademik siswa berdasarkan beberapa dimensi, antara lain : dimensi kelas, dimensi periode (tahun ajaran dan semester), dimensi mata pelajaran, dimensi tata tertib, dan dimensi
- c. Sistem membantu dalam proses pengambilan keputusan terkait dengan proses rekomendasi beasiswa dengan menggunakan metode *Fuzzy Database*, sehingga akan didapatkan hasil calon penerima beasiswa.
- d. Parameter yang digunakan dalam proses rekomendasi didapatkan dari database akademik siswa SMAN 7 Malang, yaitu nilai akademik, penghasilan orang tua, dan tingkat kedisiplinan siswa.

- e. Sistem akan menampilkan hasil dari proses rekomendasi beasiswa dalam bentuk tabel.

### 3.1.3 Analisa Kebutuhan Non-Fungsional Sistem

Kebutuhan non-fungsional mengacu pada atribut perilaku yang harus dimiliki oleh sebuah sistem, seperti *performance* dan *useability*. Kebutuhan non-fungsional ini ditinjau dari segi operasional, keamanan, informasi dan *performance* (kinerja).

#### 1. Operasional

Kebutuhan non-fungsional yang ditinjau dari segi operasional ini menjelaskan secara teknis bagaimana sistem beroperasi dan perangkat lunak apa saja yang digunakan. Penjelasan mengenai perangkat lunak dibutuhkan sebagai dukungan proses instalasi sebelum sistem pendukung keputusan dibuat.

- a. *System existing* : sistem pendukung keputusan rekomendasi beasiswa pada SMA Negeri 7 Malang dan penerapan *datawarehouse* dalam menganalisa data akademik siswa.
- b. Proses rekomendasi beasiswa yang diimplementasikan pada sistem menggunakan metode *fuzzy database*.
- c. Sumber data berasal dari database akademik siswa SMAN 7 Malang yang menggunakan *database MySQL*.
- d. Sedangkan proses perancangan *datawarehouse* menggunakan OLAP Pentaho.

#### 2. Keamanan

Kebutuhan non-fungsional yang ditinjau dari segi keamanan dilakukan dengan adanya sistem password untuk menjaga keamanan dan kenyamanan bagi pengguna untuk dapat masuk ke dalam sistem.

#### 3. Informasi

Informasi yang diberikan oleh sistem merupakan sebuah laporan dari data akademik siswa, serta data calon penerima beasiswa.

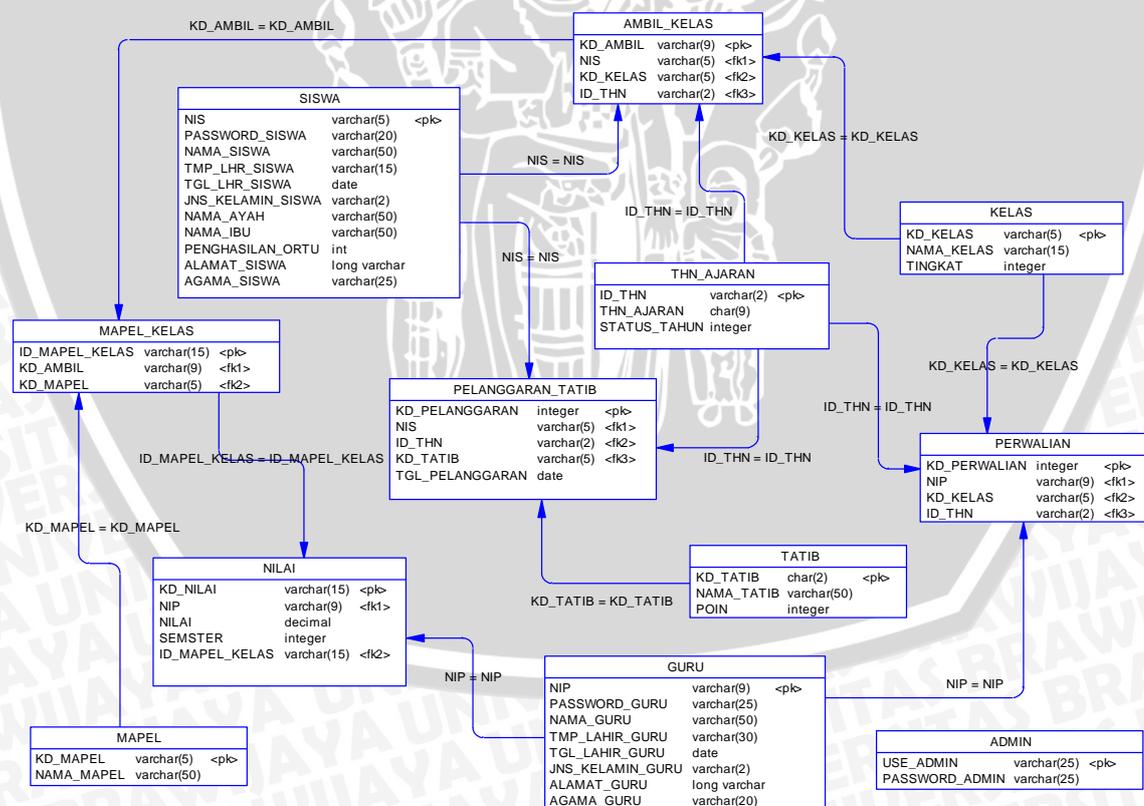
#### 4. Kinerja

Kinerja dapat dilihat dari kemampuan sebuah data untuk tetap aman dan dapat diakses dengan mengintegrasikan beberapa komponen database yang sudah ada. Dalam proses pemberian informasi yang mendukung pengambilan keputusan, diharapkan mampu memberikan kemudahan bagi pengguna untuk mengakses data.

### 3.2 Perancangan *Datawarehouse*

#### 3.2.1 Analisa Sumber Data

Sumber data yang digunakan dalam penelitian ini berasal dari *database* akademik siswa SMA Negeri 7 Malang. Data awal yang digunakan dalam proses perancangan sistem ini merupakan data relasional yang tersimpan dalam *database MySQL*. Diagram relasional pada data akademik SMAN 7 Malang ditunjukkan pada gambar 3.2



**Gambar 3.2** Diagram Relasional Data Akademik SMAN 7 Malang

Data akademik tersebut memiliki 13 tabel sebagai pembentuk database, dan setiap tabel memiliki atribut-atribut yang berelasi terhadap tabel lainnya. Berikut ini merupakan penjelasan setiap tabel yang terdapat pada database akademik siswa SMAN 7 Malang.

1. Tabel admin

Tabel admin merupakan tabel yang berfungsi untuk menyimpan data administrator sistem tersebut, tabel admin ini hanya berdiri sendiri dan tidak berelasi dengan tabel lainnya.

2. Tabel guru

Tabel guru merupakan tabel yang berfungsi untuk menyimpan semua data guru termasuk akun (*username* dan *password*) dari setiap guru yang digunakan dalam proses *log in* pada sistem. Tabel guru ini berelasi dengan tabel nilai, tabel perwalian, dan tabel jadwal.

3. Tabel siswa

Tabel siswa merupakan tabel yang digunakan untuk menyimpan seluruh informasi mengenai data diri setiap siswa (NIS, nama, alamat, pekerjaan orang tua, dan sebagainya). Tabel siswa ini juga menyimpan data akun setiap siswa yang akan digunakan dalam proses *log in*. Tabel siswa berelasi dengan tabel *ambil\_kelas*, dan tabel *pelanggaran\_tatib*.

4. Tabel kelas

Tabel kelas merupakan tabel yang berfungsi untuk menyimpan data semua kelas yang terdapat pada SMAN 7 Malang. Di dalam tabel ini terdapat sebuah atribut “tingkat”, atribut ini digunakan untuk menyimpan informasi tentang tingkatan kelas, yaitu tingkat 1, tingkat 2, dan tingkat 3. Tabel ini berelasi dengan tabel *ambil\_kelas*, tabel *jadwal*, dan tabel *perwalian*.

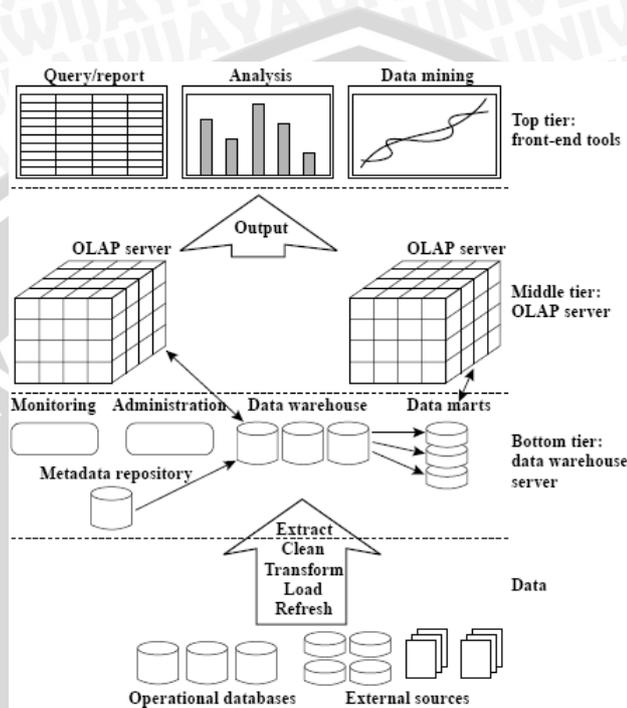
5. Tabel mapel

Tabel mapel merupakan tabel yang berfungsi untuk menyimpan semua data mata pelajaran yang terdapat pada SMAN 7 Malang. Tabel ini berelasi dengan tabel *jadwal*, dan tabel *mapel\_kelas*.

6. Tabel tatib  
Tabel tatib merupakan tabel yang berfungsi untuk menyimpan semua jenis peraturan atau tata tertib sekolah beserta besarnya poin yang dikenakan kepada siswa jika melanggar tata tertib tersebut. Tabel tatib ini hanya berelasi dengan tabel pelanggaran\_tatib.
7. Tabel thn\_ajaran  
Tabel thn\_ajaran ini berfungsi untuk menyimpan tahun ajaran baik yang sedang ditempuh ataupun yang telah ditempuh. Tabel ini berelasi dengan tabel perwalian, tabel jadwal, tabel pelanggaran\_tatib, dan tabel ambil\_kelas.
8. Tabel ambil\_kelas  
Tabel ambil\_kelas merupakan sebuah tabel relasional atau tabel penghubung yang menyimpan semua data siswa yang berada pada kelas tertentu dalam tahun ajaran tertentu. Tabel ini berelasi dengan tabel siswa, tabel thn\_ajaran, dan tabel kelas.
9. Tabel mapel\_kelas  
Tabel mapel\_kelas juga merupakan tabel relasional atau tabel penghubung yang berfungsi untuk menyimpan data mata pelajaran di suatu kelas tertentu. Tabel ini berelasi dengan tabel mapel, dan tabel nilai.
10. Tabel nilai  
Tabel nilai merupakan tabel yang digunakan untuk menyimpan data nilai dari setiap siswa dari suatu mata pelajaran. Tabel nilai ini berelasi dengan tabel guru, dan tabel mapel\_kelas.
11. Tabel perwalian  
Tabel perwalian merupakan suatu tabel penghubung yang berfungsi untuk menyimpan data perwalian kelas dalam suatu tahun ajaran. Tabel ini berelasi dengan tabel guru, tabel kelas, dan tabel thn\_ajaran.
12. Tabel pelanggaran\_tatib  
Tabel pelanggaran\_tatib merupakan tabel yang berfungsi untuk menyimpan informasi mengenai data semua siswa yang melakukan suatu pelanggaran tata tertib dalam tahun ajaran tertentu. Tabel ini berelasi dengan tabel siswa, tabel tatib, dan tabel thn\_ajaran.

### 3.2.2 Arsitektur *Datawarehouse*

Dalam pembuatan *datawarehouse* metode yang digunakan dalam penelitian ini adalah mengikuti arsitektur *datawarehouse* tiga tingkat seperti pada Gambar 3.3 :



**Gambar 3.3** Arsitektur Tiga Tingkat *Datawarehouse* [HAN-06]

1. Tingkat bawah (*bottom tier*)

Tingkat bawah merupakan suatu server sistem basis data relasional yang berfungsi sebagai tempat penyimpanan dan pengolahan data. Pada tahap ini data diambil dari data operasional, kemudian dilakukan proses ETL (*Extract, Transform, Load*) agar data yang digunakan *valid*. Data yang sudah *valid* tersebut disimpan sebagai *datawarehouse*.

2. Tingkat tengah (*middle tier*)

Lapisan tengah merupakan tempat penyimpanan suatu struktur kubus OLAP yang disebut OLAP Server. Pada tahap ini akan dibuat suatu struktur kubus OLAP.

### 3. Tingkat atas (*top tier*)

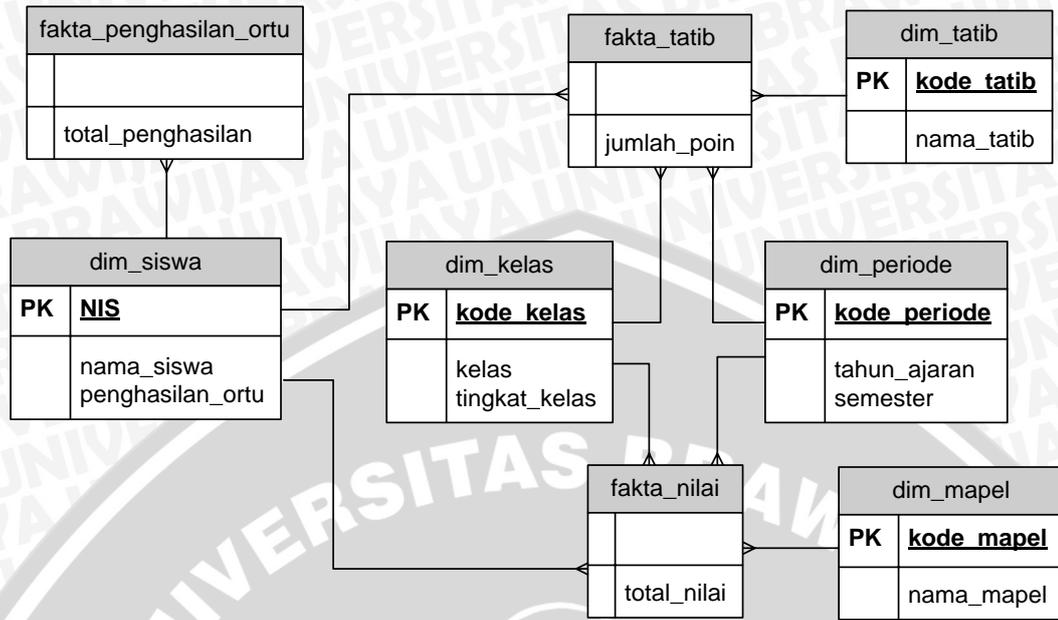
Pada tingkat atas merupakan lapisan untuk *end user* yang berfungsi untuk menampilkan ringkasan dari isi *datawarehouse* yang merupakan hasil dari operasi OLAP. Ringkasan dari hasil operasi OLAP tersebut akan ditampilkan dalam bentuk  *Crosstab* tabel dan grafik agar memudahkan dalam proses analisis data.

### 3.2.3 Perancangan Skema *Datawarehouse*

Terdapat tiga tipe pemodelan data multidimensi atau *datawarehouse*, yaitu *star schema*, *snowflake schema*, dan *galaxy schema*. Sedangkan tipe pemodelan yang digunakan dalam penelitian ini adalah *galaxy scheme*. Karena sesuai dengan kebutuhan penelitian yang membutuhkan lebih dari satu tabel fakta, sehingga terdapat proses penggabungan dari beberapa tabel fakta dan tabel dimensi menjadi satu skema. *Galaxy scheme* didesain berdasarkan hasil dari analisis kebutuhan dengan menentukan apa saja *fact tabel* dan *dimension tabel* yang diperlukan.

Skema *galaxy* ini mempunyai struktur beberapa tabel dimensi dan tabel fakta. Tabel dimensi adalah tabel yang berisi informasi secara mendetil yang membantu untuk menyimpulkan informasi dalam tabel fakta. Tabel fakta adalah tabel yang berisi data suatu transaksi pada periode tertentu. Data dalam tabel fakta pada umumnya sering berubah-ubah dan bertipe numerik.

Berdasarkan hasil analisis kebutuhan dalam proses analisis data akademik siswa, ditentukan lima tabel dimensi dan tiga tabel fakta. Kelima tabel dimensi tersebut antara lain dimensi siswa, dimensi tata\_tertib, dimensi kelas, dimensi mata\_pelajaran, dan dimensi periode. Sedangkan untuk tabel fakta antara lain fakta tata\_tertib, fakta nilai, dan fakta penghasilan. Ketiga tabel fakta tersebut akan digunakan dalam proses rekomendasi beasiswa. Rancangan *galaxy scheme* yang akan digunakan pada *datawarehouse* ditunjukkan pada gambar 3.4.



**Gambar 3.4** Galaxy Scheme

Berikut ini penjelasan dari tabel fakta dan dimensi pada *Galaxy Scheme* yang digunakan :

6. Tabel dim\_siswa

Tabel dim\_siswa merupakan tabel yang menyimpan informasi dari data siswa yang dibutuhkan dalam proses analisis dan rekomendasi beasiswa. Tabel dimensi ini hanya berisi NIS sebagai *primary key*, nama siswa, dan penghasilan orangtua. Tabel dim\_siswa ini berelasi terhadap tabel fakta\_penghasilan\_ortu, tabel fakta\_tatib, dan tabel fakta\_nilai. Penjelasan tabel dim\_siswa ditunjukkan oleh tabel 3.1.

**Tabel 3.1** Tabel dim\_siswa

Atribut	Type Data	Keterangan
NIS	varchar	Menyimpan Nomor Induk Siswa ( <i>primary key</i> ).
nama_siswa	varchar	Menyimpan nama siswa.
penghasilan_ortu	int	Menyimpan jumlah penghasilan orang tua siswa.

## 7. Tabel dim\_periode

Tabel dim\_periode merupakan tabel yang menyimpan informasi berkenaan dengan tahun ajaran dan semester. Tabel dimensi ini berisi kode\_periode sebagai *primary key*, tahun ajaran, dan semester. Tabel dim\_periode ini berelasi terhadap tabel fakta\_nilai dan tabel fakta\_tatib. Penjelasan tabel dim\_periode ditunjukkan oleh tabel 3.2.

**Tabel 3.2** Tabel dim\_periode

Atribut	Type Data	Keterangan
kode_periode	varchar	Menyimpan kode periode ( <i>primary key</i> ).
tahun_ajaran	varchar	Menyimpan data tahun ajaran.
semester	int	Menyimpan data semester tiap tahun ajaran.

## 8. Tabel dim\_kelas

Tabel dim\_kelas merupakan tabel yang menyimpan informasi berkenaan dengan kelas yang terdapat pada SMAN 7 malang. Tabel dimensi ini berisi kode\_kelas sebagai *primary key*, kelas, dan tingkat\_kelas. Tabel dim\_kelas ini berelasi terhadap tabel fakta\_nilai dan tabel fakta\_tatib. Penjelasan tabel dim\_kelas ditunjukkan oleh tabel 3.3.

**Tabel 3.3** Tabel dim\_kelas

Atribut	Type Data	Keterangan
kode_kelas	varchar	Menyimpan kode kelas ( <i>primary key</i> )
kelas	varchar	Menyimpan semua kelas yang ada.
tingkat_kelas	int	Memeberikan informasi tentang tingkatan kelas, yaitu tingkat 1, 2, 3.

## 9. Tabel dim\_tatib

Tabel dim\_tatib merupakan tabel yang menyimpan informasi berkenaan dengan peraturan atau tata tertib pada SMAN 7 Malang. Tabel dimensi ini berisi kode\_tatib sebagai *primary key*, nama\_tatib, dan poin. Tabel dim\_tatib ini hanya berelasi terhadap tabel fakta\_tatib. Penjelasan tabel dim\_tatib ditunjukkan oleh tabel 3.4.

**Tabel 3.4** Tabel dim\_tatib

Atribut	Tipe Data	Keterangan
kode_tatib	varchar	Menyimpan kode tata tertib sekolah ( <i>primary key</i> ).
nama_tatib	varchar	Menyimpan semua jenis tata tertib sekolah.
poin	int	Menyimpan jumlah poin pelanggaran terhadap setiap tata tertib.

## 10. Tabel dim\_mapel

Tabel dim\_mapel merupakan tabel yang menyimpan informasi berkenaan dengan mata pelajaran pada SMAN 7 Malang. Tabel dimensi ini berisi kode\_mapel sebagai *primary key*, dan nama\_mapel. Tabel dim\_mapel ini hanya berelasi terhadap tabel fakta\_nilai. Penjelasan tabel dim\_mapel ditunjukkan oleh tabel 3.5.

**Tabel 3.5** Tabel dim\_mapel

Atribut	Tipe Data	Keterangan
kode_mapel	varchar	Menyimpan kode mata pelajaran ( <i>primary key</i> ).
nama_mapel	varchar	Menyimpan semua jenis mata pelajaran.

## 11. Tabel fakta\_penghasilan\_ortu

Tabel fakta\_penghasilan\_ortu merupakan tabel yang menyimpan informasi tentang fakta total penghasilan orang tua dari setiap siswa. Di dalam tabel ini terdapat sebuah *measure* yaitu total penghasilan orang tua yang telah diproses dari tabel dim\_siswa. Penjelasan tabel fakta\_penghasilan\_ortu ditunjukkan oleh tabel 3.6.

**Tabel 3.6** Tabel fakta\_penghasilan\_ortu

Atribut	Tipe Data	Keterangan
NIS	varchar	Menyimpan Nomor Induk Siswa ( <i>foreign key</i> ).
nama_siswa	varchar	Menyimpan nama semua siswa.
Total_penghasilan	int	Menyimpan hasil total penghasilan kedua orang tua untuk setiap siswa.

## 12. Tabel fakta\_nilai

Tabel fakta\_nilai merupakan tabel yang menyimpan informasi tentang fakta besarnya nilai pelajaran dari setiap siswa dalam satu periode. Di dalam tabel ini juga terdapat sebuah *measure* yaitu rata-rata nilai setiap siswa yang telah diproses dari tabel dim\_siswa, tabel dim\_kelas, tabel dim\_mapel, dan tabel dim\_periode. Penjelasan tabel fakta\_nilai ditunjukkan oleh tabel 3.7.

**Tabel 3.7** Tabel fakta\_nilai

Atribut	Type Data	Keterangan
NIS	varchar	Menyimpan Nomor Induk Siswa ( <i>foreign key</i> ).
Kode_kelas	varchar	Menyimpan kode_kelas ( <i>foreign key</i> ).
Kode_mapel	varchar	Menyimpan kode_mapel ( <i>foreign key</i> ).
Kode_periode	varchar	Menyimpan kode_periode ( <i>foreign key</i> ).
Nilai	decimal	Menyimpan total nilai pelajaran untuk setiap siswa.

## 13. Tabel fakta\_tatib

Tabel fakta\_tatib merupakan tabel yang menyimpan informasi tentang fakta besarnya poin pelanggaran yang dilakukan setiap siswa dalam satu periode. Di dalam tabel ini juga terdapat sebuah *measure* yaitu besarnya poin pelanggaran setiap siswa yang telah diproses dari tabel dim\_siswa, tabel dim\_kelas, tabel dim\_tatib, dan tabel dim\_periode. Penjelasan tabel fakta\_tatib ditunjukkan oleh tabel 3.8.

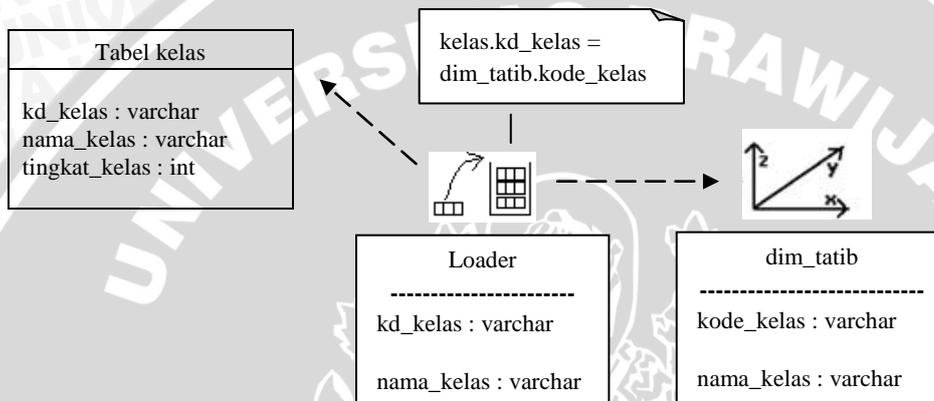
**Tabel 3.8** Tabel fakta\_tatib

Atribut	Type Data	Keterangan
NIS	varchar	Menyimpan Nomor Induk Siswa ( <i>foreign key</i> ).
kode_kelas	varchar	Menyimpan kode_kelas ( <i>foreign key</i> ).
kode_tatib	varchar	Menyimpan kode_tatib ( <i>foreign key</i> ).
kode_periode	varchar	Menyimpan kode_periode ( <i>foreign key</i> ).
jumlah_poin	int	Menyimpan jumlah poin pelanggaran untuk setiap siswa.

### 3.2.4 Proses ETL (*Extraction, Transformation, Loading*)

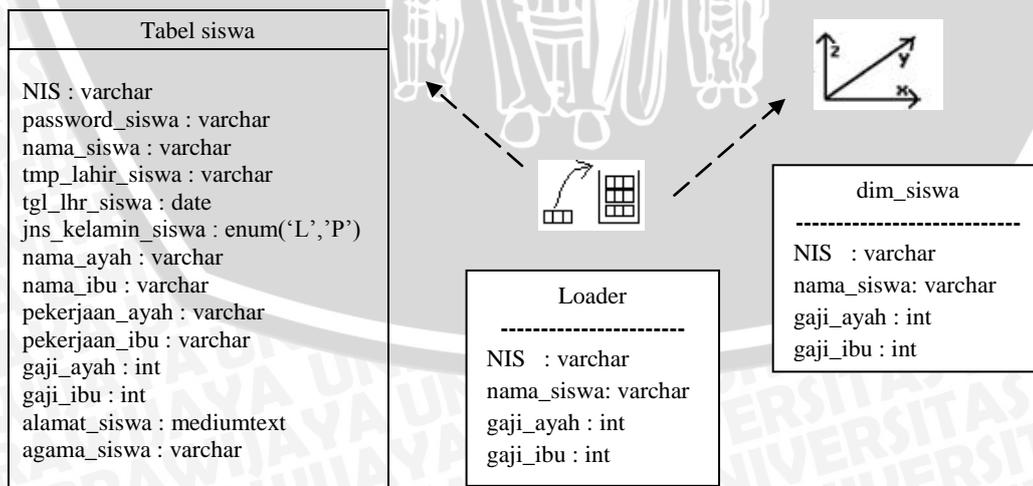
Untuk mentransformasi data awal atau sumber data menjadi sebuah datawarehouse, diperlukan suatu proses ETL (*Extraction, Transformation, Loading*). Berikut ini adalah rancangan proses ETL yang digambarkan dalam bentuk UML.

Proses memasukkan (*Loading*) data ke dalam tabel `dim_kelas` ditunjukkan oleh gambar 3.5.



**Gambar 3.5** UML Proses *Loading* Tabel `dim_kelas`

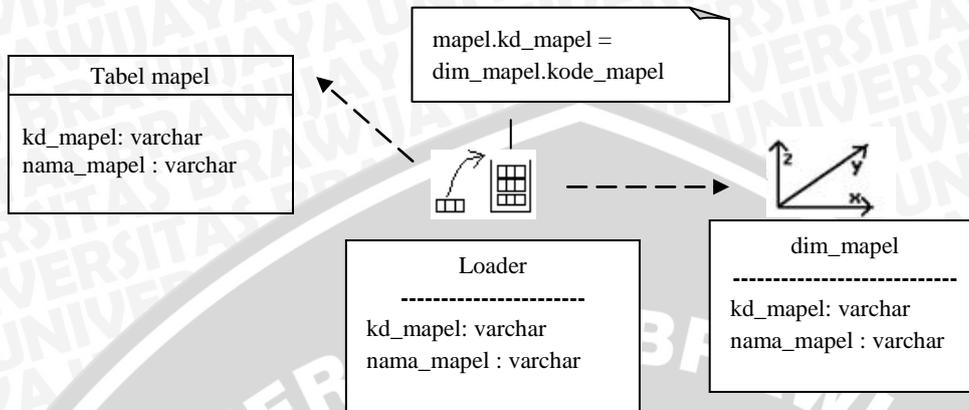
Proses memasukkan (*Loading*) data ke dalam tabel `dim_siswa` ditunjukkan oleh gambar 3.6.



**Gambar 3.6** UML Proses *Loading* Tabel `dim_siswa`

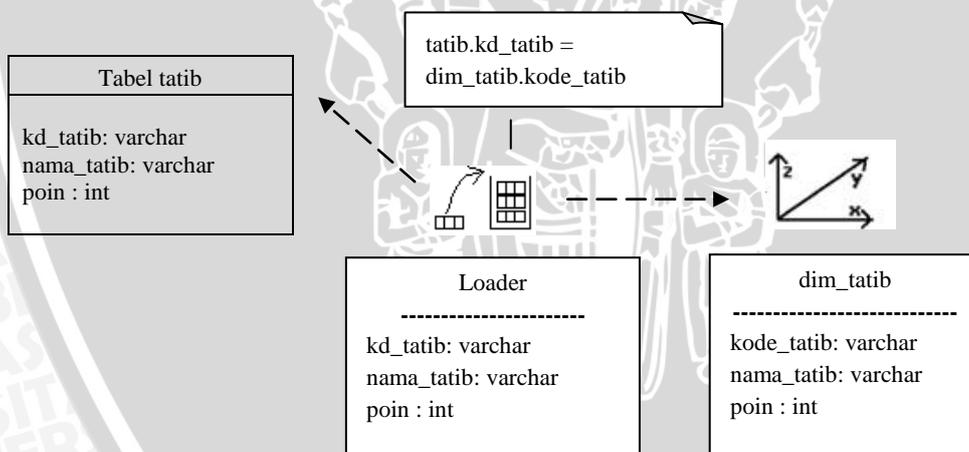


Proses memasukkan (*Loading*) data ke dalam tabel `dim_mapel` ditunjukkan oleh gambar 3.7.



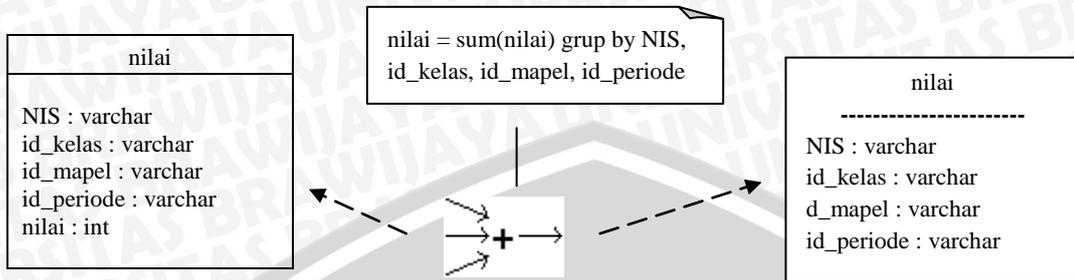
**Gambar 3.7** UML Proses *Loading* Tabel `dim_mapel`

Proses memasukkan (*Loading*) data ke dalam tabel `dim_tatib` ditunjukkan oleh gambar 3.8.



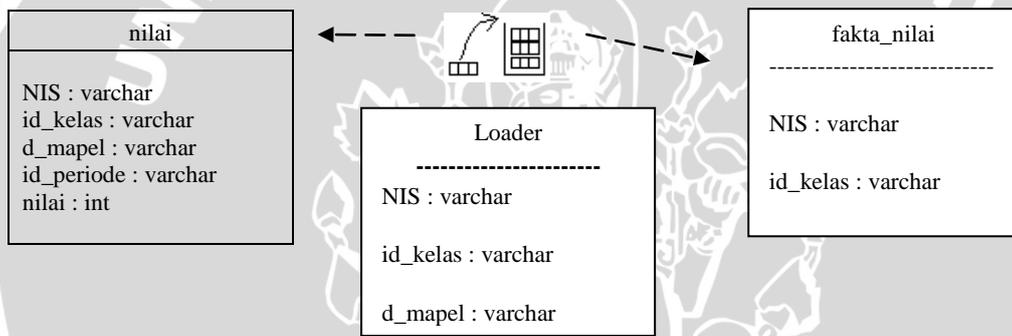
**Gambar 3.8** UML Proses *Loading* Tabel `dim_tatib`

Proses agregasi untuk mendapatkan total nilai dari tabel\_nilai ditunjukkan oleh gambar 3.9.



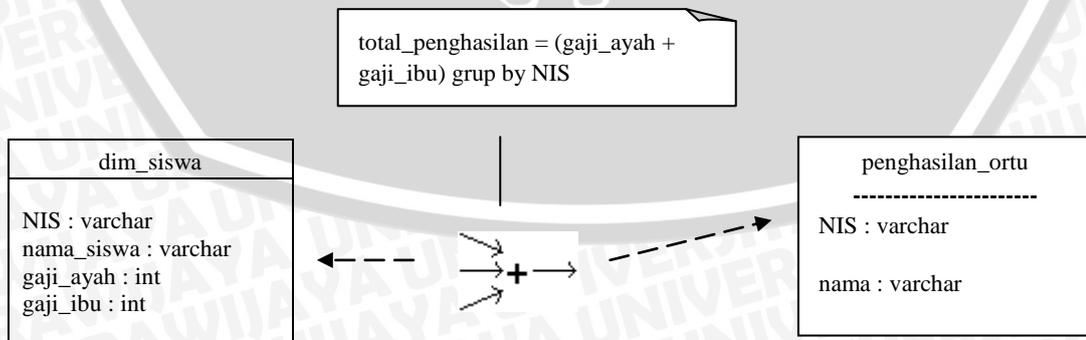
**Gambar 3.9** UML Proses Agregasi Nilai Siswa

Proses memasukkan (*Loading*) data total nilai ke dalam tabel fakta\_nilai ditunjukkan oleh gambar 3.10.



**Gambar 3.10** UML Proses *Loading* Tabel fakta\_nilai

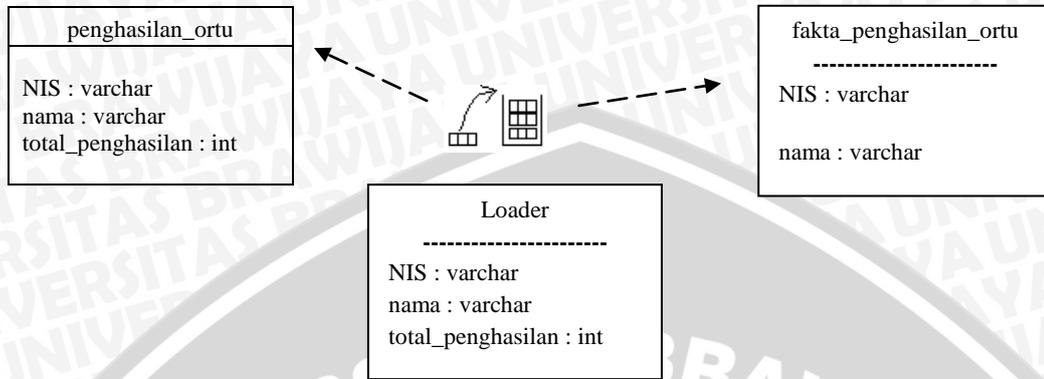
Proses agregasi untuk mendapatkan jumlah penghasilan kedua orang tua dari dim\_siswa ditunjukkan oleh gambar 3.11.



**Gambar 3.11** UML Proses Agregasi Nilai Penghasilan Orang Tua Siswa



Proses memasukkan (*Loading*) data jumlah penghasilan orang tua ke dalam tabel `fakta_penghasilan_ortu` ditunjukkan oleh gambar 3.12.



**Gambar 3.12** UML Proses *Loading* Tabel `fakta_penghasilan_ortu`

### 3.2.5 Struktur Kubus Data

Setelah semua tahapan ETL dan pembuatan *datawarehouse* selesai, tahapan selanjutnya yaitu merancang struktur kubus data di *OLAP Server*. Rancangan struktur kubus data disesuaikan dengan skema *datawarehouse* yang telah dibuat yaitu dengan memanfaatkan tabel `dim_siswa`, tabel `dim_kelas`, tabel `dim_mapel`, tabel `dim_tatib`, tabel `dim_periode`, tabel `fakta_penghasilan_ortu`, tabel `fakta_tatib`, dan tabel `fakta_nilai`. Dalam pembuatan kubus data (*cube*) ditentukan juga suatu hirarki data dan *measure*. Terdapat tiga *measure* yang digunakan pada kubus data ini yang mewakili dari setiap tabel fakta, yaitu *measure* `total_penghasilan`, *measure* `jumlah_poin`, dan *measure* `total_nilai`. Sehingga terdapat tiga struktur kubus data untuk masing-masing tabel fakta. Misalnya : *Measure* `total_nilai` digunakan untuk mengetahui besarnya nilai siswa pada kondisi atau sudut pandang tertentu, misalnya pada kelas tertentu atau pada suatu periode tertentu. Suatu hirarki data berfungsi agar hasil dari proses analisis data akademik siswa dapat dilihat dari hirarki yang paling tinggi sampai yang paling rendah. Sehingga diperoleh suatu struktur kubus data seperti berikut ini :

### 1. *Struktur Kubus Data Nilai*

- a. Dimensi Siswa (semua data siswa)
  - ↳ Nama Siswa (*drill down* dari semua data siswa)
- b. Dimensi Kelas (semua kelas)
  - ↳ Tingkat Kelas (*drill down* dari semua kelas)
  - ↳ Kelas (*drill down* dari tingkat kelas)
- c. Dimensi Mata Pelajaran (semua mata pelajaran)
  - ↳ Mata Pelajaran (*drill down* dari semua mata pelajaran)
- d. Dimensi Periode (semua periode)
  - ↳ Tahun Ajaran (*drill down* dari semua periode)
  - ↳ Semester (*drill down* dari tahun ajaran)

### 2. *Struktur Kubus Data Tata Tertib*

- a. Dimensi Periode (semua periode)
  - ↳ Tahun Ajaran (*drill down* dari semua periode)
  - ↳ Semester (*drill down* dari tahun ajaran)
- b. Dimensi Kelas (semua kelas)
  - ↳ Tingkat Kelas (*drill down* dari semua kelas)
  - ↳ Kelas (*drill down* dari tingkat kelas)
- c. Dimensi Tata Tertib (semua tata tertib)
  - ↳ Nama tata tertib (*drill down* dari semua tata tertib)
- d. Dimensi Siswa (semua data siswa)
  - ↳ Nama Siswa (*drill down* dari semua data siswa)

### 3. Struktur Kubus Data Penghasilan Orang Tua

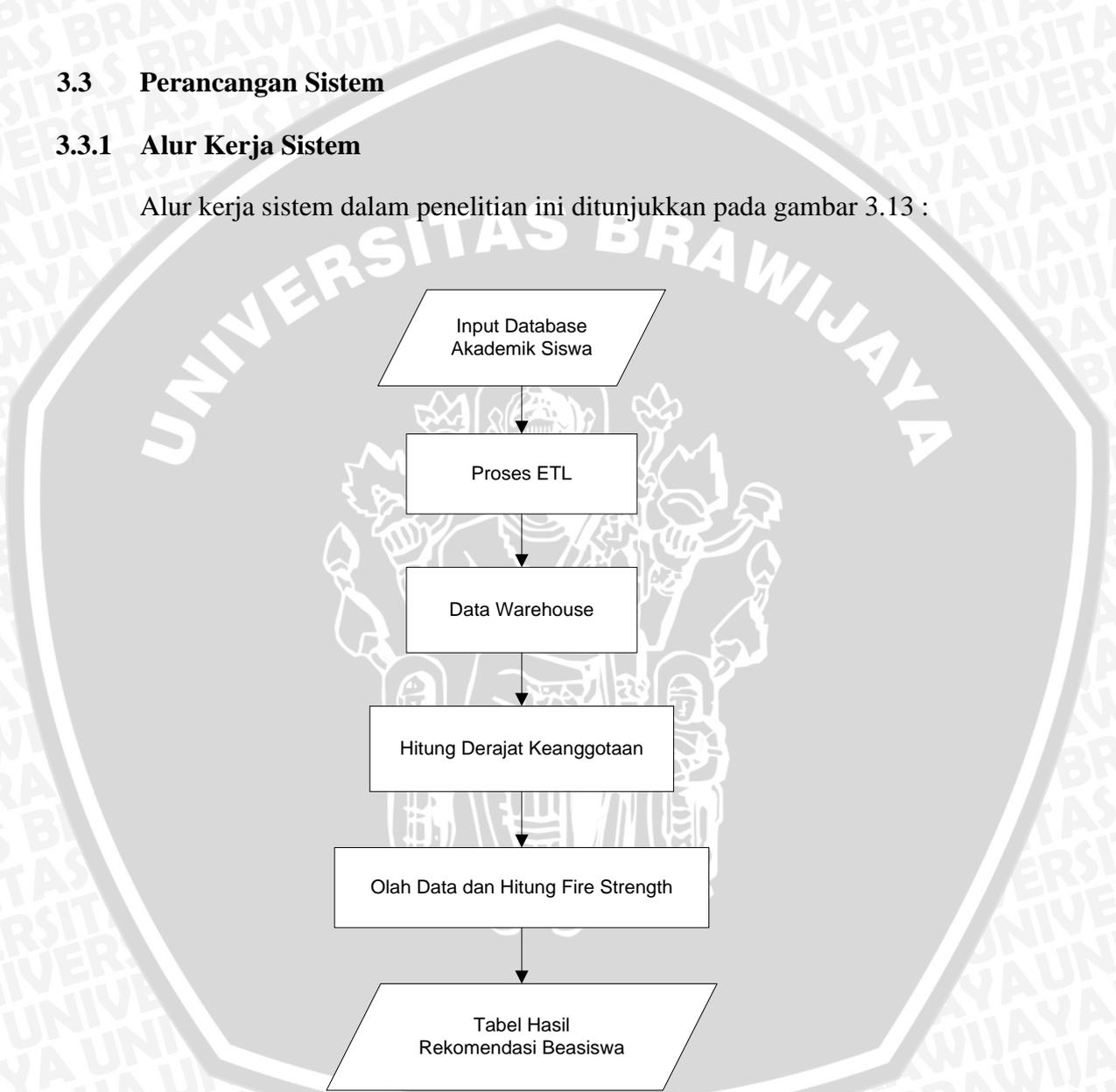
#### a. Dimensi Siswa (semua data siswa)

↳ Nama Siswa (*drill down* dari semua data siswa)

## 3.3 Perancangan Sistem

### 3.3.1 Alur Kerja Sistem

Alur kerja sistem dalam penelitian ini ditunjukkan pada gambar 3.13 :



**Gambar 3.13** Alur Kerja Sistem

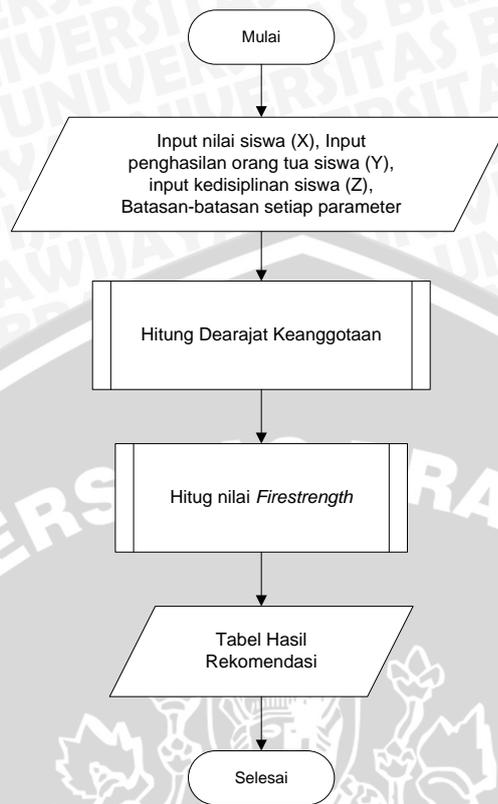
Pada penelitian ini, hal pertama yang harus dilakukan adalah menentukan *database* yang akan digunakan dalam proses analisa dalam rekomendasi beasiswa.

*Database* (data operasional) yang digunakan adalah *database* akademik siswa SMA Negeri 7 Malang. Kemudian dilakukan proses ETL (*Extraction, Loading, Transformation*) pada *database* tersebut untuk menyeleksi data dan tabel-tabel yang akan digunakan dalam proses rekomendasi beasiswa, sehingga hanya atribut dan variabel yang berkaitan dengan parameter beasiswa yang akan diproses. Dari *database* akademik siswa yang telah melalui proses ETL, maka terbentuklah sebuah *datawarehouse* yang akan digunakan dalam proses rekomendasi beasiswa dengan metode *fuzzy database*.

Langkah selanjutnya adalah menentukan dan menghitung fungsi keanggotaan dari setiap parameter yang digunakan dalam proses rekomendasi beasiswa yang telah dipetakan dan ditentukan batasan-batasannya. Dari rumus fungsi keanggotaan yang telah ditentukan, maka *firestrenght* untuk setiap siswa dapat dihitung. Penghitungan *firestrenght* dilakukan dengan menggunakan operasi *And*, yaitu membandingkan semua parameter pada masing-masing siswa yang telah dihitung derajat keanggotaannya, kemudian dicari nilai terkecil dari hasil perbandingan tersebut. Nilai *firestrenght* untuk setiap siswa akan disimpan pada tabel dan akan disajikan dalam bentuk grafik atau *chart*. *Firestrenght* ini menunjukkan seberapa besar rekomendasi yang diberikan oleh sistem untuk setiap siswa.

### 3.3.2 Diagram Alir

Perangkat lunak yang akan dibuat ini memiliki tiga proses utama, yaitu proses perancangan *datawarehouse*, proses menentukan dan menghitung fungsi keanggotaan untuk setiap parameter yang ada, dan yang ketiga adalah menghitung nilai *firestrenght* untuk setiap siswa. Nilai *firestrenght* ini digunakan untuk menunjukkan seberapa besar nilai rekomendasi pada masing-masing siswa. Diagram alir utama perancangan sistem ditunjukkan oleh gambar 3.14.



**Gambar 3.14** Diagram Alir Utama Perancangan Sistem

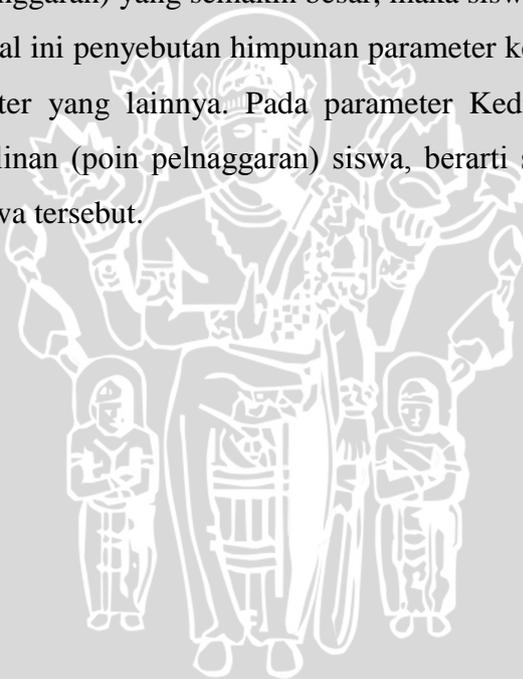
Berikut ini adalah penjelasan dari diagram alir utama perancangan sistem :

1. Nilai siswa, penghasilan orang tua siswa, dan kedisiplinan siswa yang tersimpan dalam *datawarehouse* dimasukkan dan siproses ke dalam sistem.
2. Dimasukkan batasan-batasan setiap parameter ke dalam sistem.
3. Dihitung derajat keanggotaan setiap parameter yang mempengaruhi proses rekomendasi beasiswa.
4. Dihitung nilai *firestrength* untuk masing-masing siswa.
5. Dihasilkan tabel rekomendasi beasiswa.

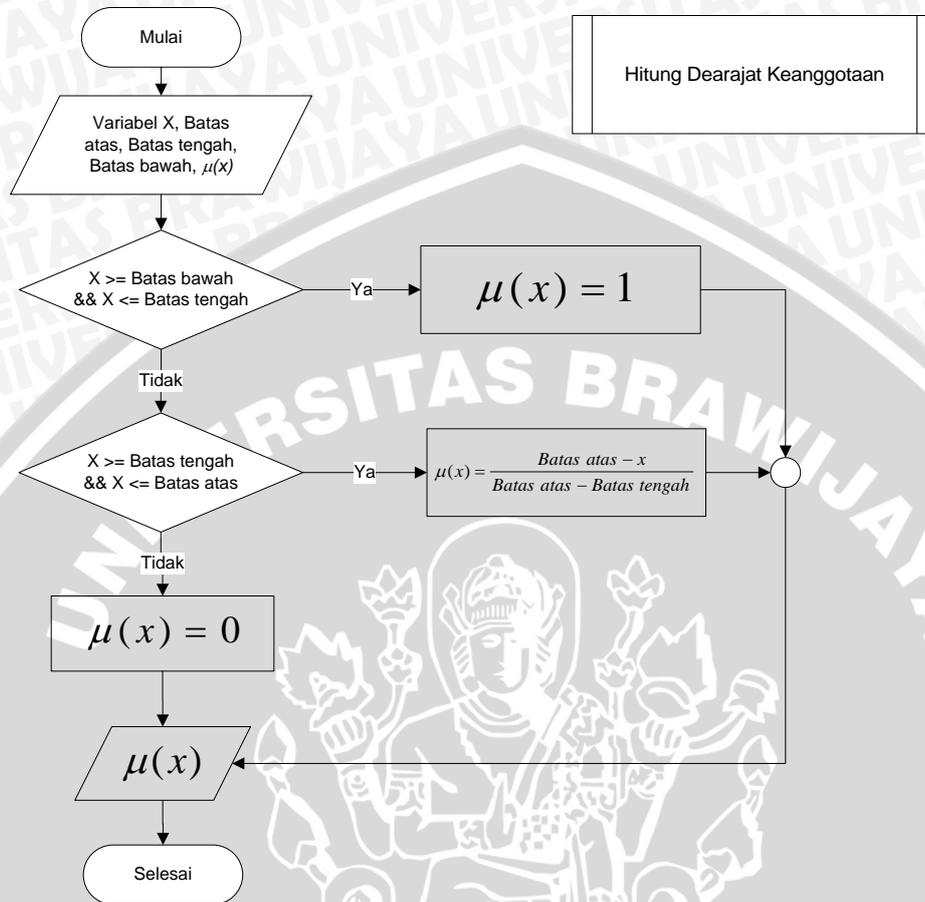
### 3.3.2.1 Menentukan Fungsi Keanggotaan

Pada penelitian ini terdapat tiga parameter yang akan dibuat fungsi keanggotaan. Setiap parameter tersebut dibentuk kembali menjadi tiga himpunan yang masing-masing akan dihitung fungsi keanggotaannya, antara lain : Kurang, Sedang, dan Baik. Tetapi untuk penyebutan bagian-bagian pada setiap parameter tersebut berbeda. Untuk parameter Nilai Akademik dibagi menjadi Nilai Kurang, Nilai Cukup, dan Nilai Baik. Untuk parameter Penghasilan Orang Tua dibagi menjadi Kurang Mampu, Mampu, dan Kaya. Sedangkan untuk Tingkat Kedisiplinan dibagi menjadi Kurang Disiplin, Cukup Disiplin, dan Disiplin.

Pada parameter tingkat kedisiplinan, jika seorang siswa memiliki nilai kedisiplinan (poin pelanggaran) yang semakin besar, maka siswa tersebut semakin tidak disiplin. Dalam hal ini penyebutan himpunan parameter kedisiplinan berbeda dengan kedua parameter yang lainnya. Pada parameter Kedisiplinan, semakin “kurang” nilai kedisiplinan (poin pelanggaran) siswa, berarti semakin baik pula tingkat kedisiplinan siswa tersebut.



1. Menentukan Fungsi Keanggotaan “Parameter Kurang”

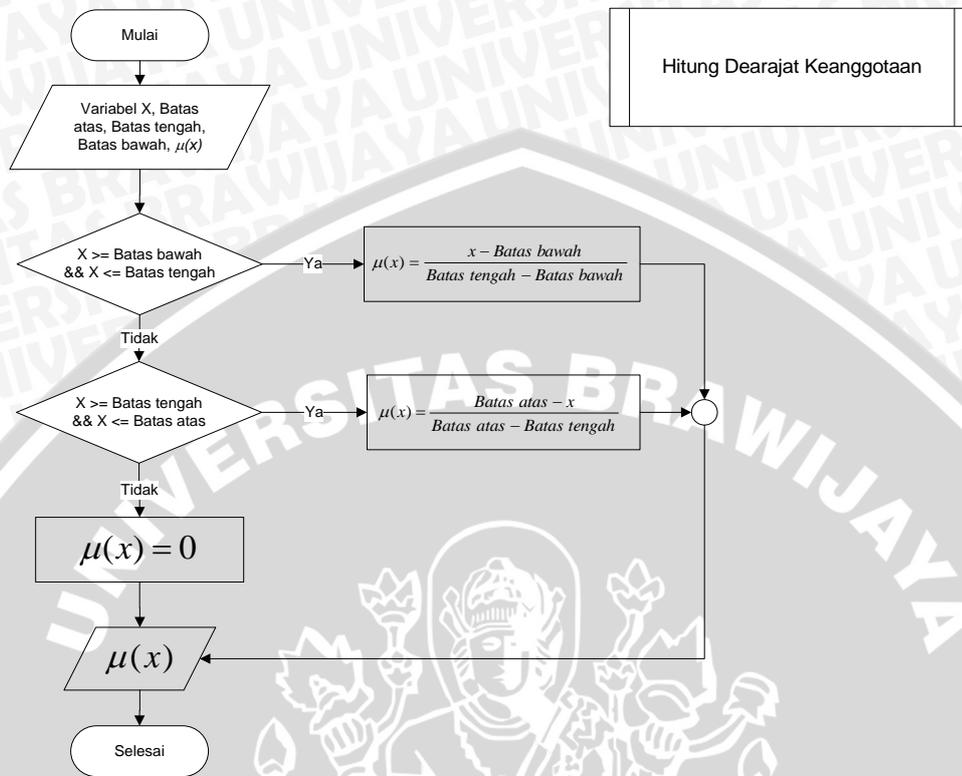


Gambar 3.15 Diagram Alir Menentukan Fungsi Keanggotaan “Parameter Kurang”

Berikut ini adalah penjelasan dari proses penentuan fungsi keanggotaan “Parameter Kurang” :

1. Dimasukkan sebuah variabel x, batas atas, batas tengah, dan batas bawah dari parameter kurang. Dimana variabel x ini merupakan variabel yang mewakili besarnya nilai dari semua parameter pada setiap siswa yang diinputkan.
2. Selanjutnya dilakukan pengecekan terhadap variabel x, jika  $x \geq$  batas bawah dan  $x \leq$  batas tengah. Maka nilai  $\mu(x)$  sama dengan 1.
3. Dan jika  $x \geq$  batas tengah dan  $x \leq$  batas atas, maka nilai 
$$\mu(x) = \frac{\text{Batas atas} - x}{\text{Batas atas} - \text{Batas tengah}}$$
4. Dan jika tidak nilai x memenuhi semua syarat yang ada, maka nilai  $\mu(x) = 0$ .

## 2. Menentukan Fungsi Keanggotaan “Parameter Cukup”



**Gambar 3.16** Diagram Alir Menentukan Fungsi Keanggotaan “Parameter Cukup”

Berikut ini adalah penjelasan dari proses penentuan fungsi keanggotaan “Parameter Cukup” :

1. Dimasukkan sebuah variabel x, batas atas, batas tengah, dan batas bawah dari parameter cukup. Dimana variabel x ini merupakan variabel yang mewakili besarnya nilai dari semua parameter pada setiap siswa yang diinputkan.

2. Jika  $x \geq$  batas bawah dan  $x \leq$  batas tengah, maka nilai

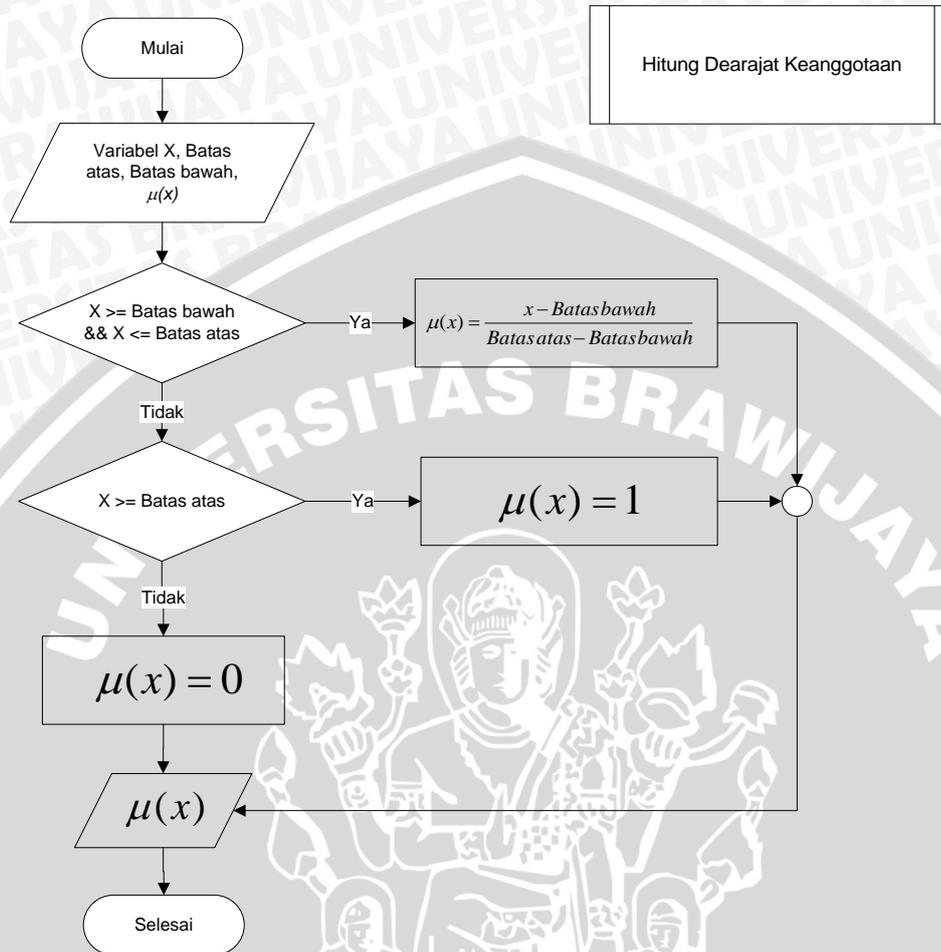
$$\mu(x) = \frac{x - \text{Batas bawah}}{\text{Batas tengah} - \text{Batas bawah}}$$

3. Dan jika  $x \geq$  batas tengah dan  $x \leq$  batas atas, maka nilai

$$\mu(x) = \frac{\text{Batas atas} - x}{\text{Batas atas} - \text{Batas tengah}}$$

4. Dan jika nilai x tidak memenuhi semua syarat yang ada, maka nilai  $\mu(x) = 0$ .

### 3. Menentukan Fungsi Keanggotaan “Parameter Baik”



**Gambar 3.17** Diagram Alir Menentukan Fungsi Keanggotaan “Parameter Baik”

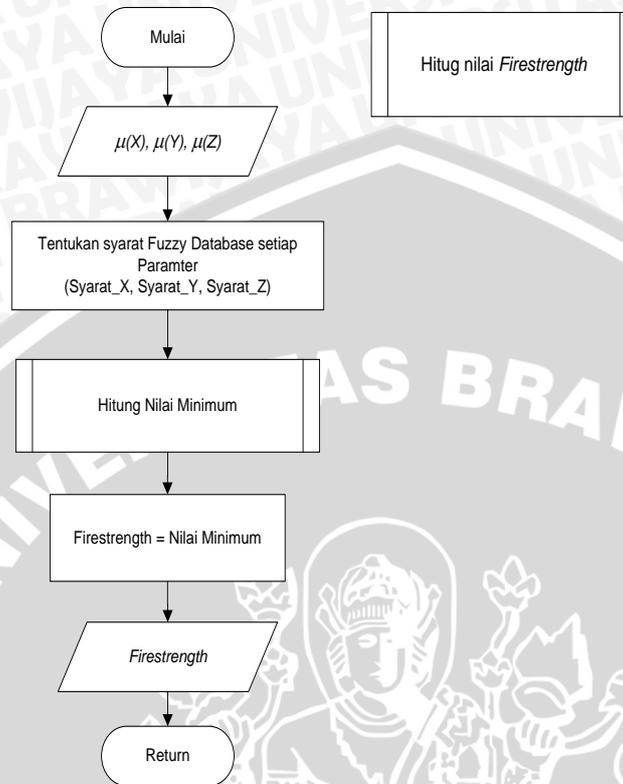
Berikut ini adalah penjelasan dari proses penentuan fungsi keanggotaan “Parameter Baik” :

1. Dimasukkan sebuah variabel x, batas atas dan batas bawah dari parameter baik. Dimana variabel x ini merupakan variabel yang mewakili besarnya nilai dari semua parameter pada setiap siswa yang diinputkan.
2. Jika  $x \geq$  batas bawah dan  $x \leq$  batas atas, maka nilai

$$\mu(x) = \frac{x - \text{Batas bawah}}{\text{Batas atas} - \text{Batas bawah}}$$

3. Dan jika  $x =$  batas atas, maka  $\mu(x) = 1$ .
4. Selanjutnya dilakukan pengecekan terhadap variabel x, jika nilai x tidak memenuhi semua syarat, maka nilai  $\mu(x) = 0$ .

### 3.3.2.2 Menghitung *Firestrength*

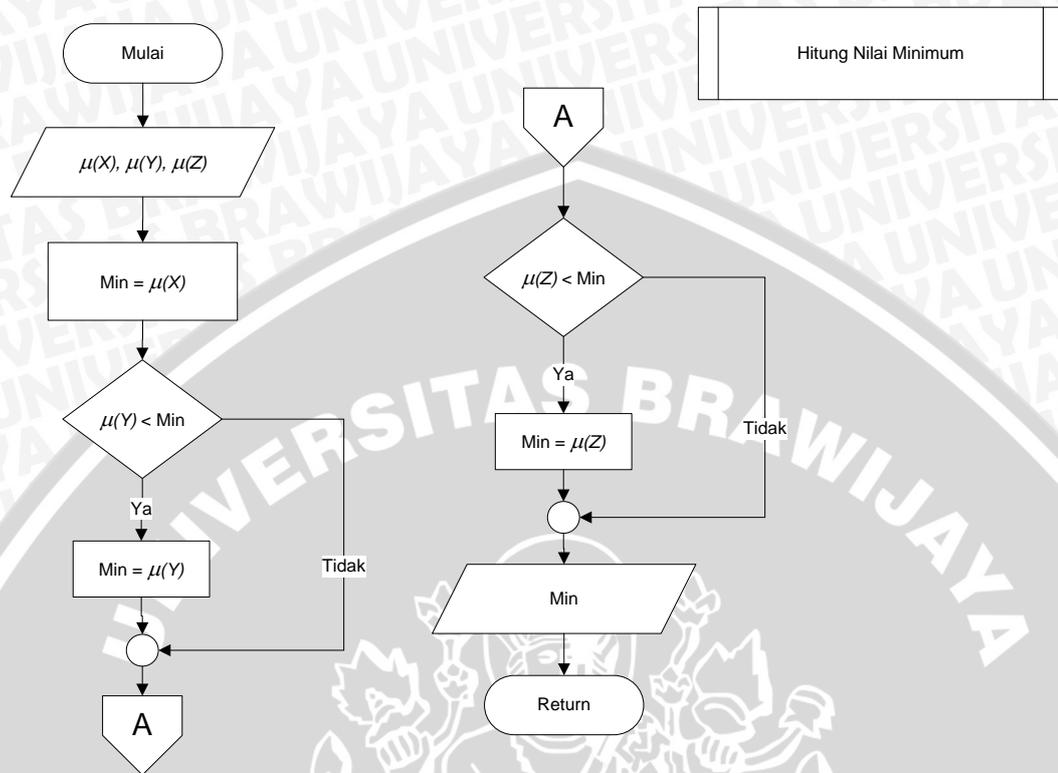


**Gambar 3.18** Diagram Alir Menghitung *Firestrength*

Berikut ini adalah penjelasan dari proses menghitung *firestrength* :

1. Dimasukkan nilai  $\mu(X)$ ,  $\mu(Y)$ ,  $\mu(Z)$
2. Ditentukan syarat-syarat *fuzzy database* setiap parameter untuk proses rekomendasi beasiswa. Misalnya rekomendasi beasiswa diproses dengan syarat Rata-rata “Nilai Baik”, Penghasilan Orang Tua “Kurang Mampu”, dan Tingkat Kedisiplinan “Disiplin”.
3. Dihitung Nilai Minimum dari semua  $\mu(X)$ ,  $\mu(Y)$ ,  $\mu(Z)$  berdasarkan parameter yang telah ditentukan.
4. Dihasilkan Nilai Minimum yang merupakan nilai *firestrength*.
5. Dihasilkan nilai *firestrength* untuk setiap siswa.

### 3.3.2.3 Menghitung Nilai Minimum



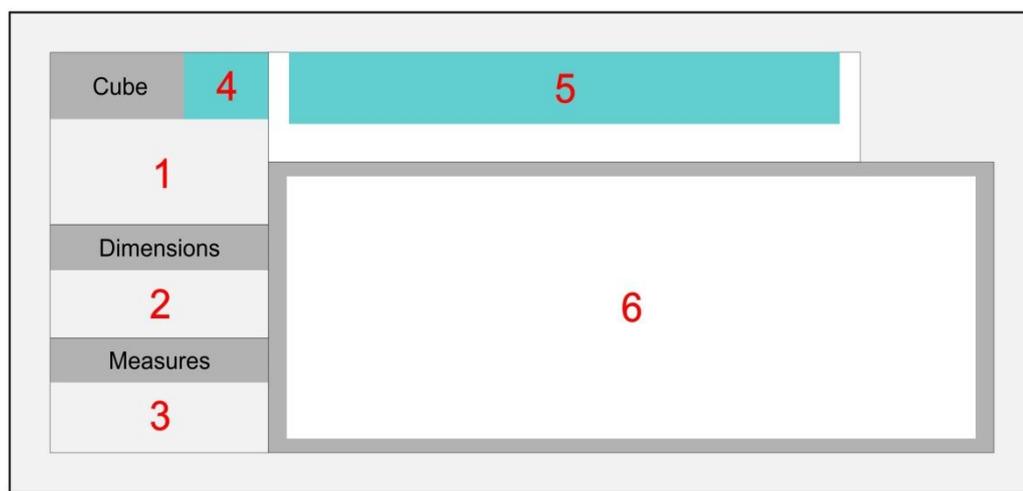
**Gambar 3.19** Diagram Alir Menghitung Nilai Minimum

Berikut ini adalah penjelasan dari proses menghitung nilai minimum :

1. Dimasukkan nilai  $\mu(X)$ ,  $\mu(Y)$ ,  $\mu(Z)$
2. Dilakukan proses penentuan nilai minimum, yaitu  $Min = \mu(X)$ .
3. Dilakukan proses pengecekan, jika  $\mu(Y) < Min$ , maka nilai minimum akan berubah menjadi  $\mu(Y)$  dan kemudian nilai minimum tersebut akan dilakukan proses pengecekan kembali terhadap  $\mu(Z)$ , sehingga akan didapatkan nilai minimum baru. Tapi jika tidak memenuhi syarat, maka nilai minimum akan tetap pada nilai sebelumnya.
4. Dihasilkan nilai Minimum.

### 3.4 Perancangan Antarmuka

Antarmuka atau *interface* pada sistem ini terdiri dari dua bagian utama, yaitu bagian yang berfungsi untuk proses rekap laporan data akademik siswa dan bagian yang berfungsi untuk proses rekomendasi beasiswa dengan menggunakan beberapa parameter sebagai inputan yang akan diproses. Untuk bagian antarmuka rekap laporan data akademik siswa ditunjukkan pada gambar 3.20.



**Gambar 3.20** Rancangan Antarmuka Rekap Laporan Data Akademik Siswa

Keterangan :

1. Bagian yang menandakan *drop down list* yang berisi daftar nama *cube* yang tersedia dan yang akan dipilih oleh pengguna.
2. Daftar nama dimensi yang sesuai dengan *cube* yang telah dipilih (pada nomor 1).
3. Daftar nama *measure* yang sesuai dengan *cube* yang telah dipilih (pada nomor 1).
4. Tombol *Refresh* yang berfungsi untuk me-*refresh* atau menampilkan ulang data dalam kondisi kosong.
5. Bagian menu berisi *button* atau tombol-tombol yang berfungsi untuk menampilkan visualisasi hasil rekap laporan yang diinginkan.

- Menampilkan hasil dari visualisasi laporan dalam bentuk *crosstab* tabel atau *chart* sesuai dengan *cube* (nomor 1), dimensi (nomor 2), dan *measure* (nomor 3) yang telah dipilih.

Sedangkan untuk bagian *interface* atau antarmuka dari proses rekomendasi beasiswa ditunjukkan oleh gambar 3.21

**Gambar 3.21** Rancangan Antarmuka Proses Rekomendasi Beasiswa

Keterangan :

- Bagian antarmuka yang merupakan syarat atau kriteria yang harus dipenuhi dalam proses rekomendasi beasiswa. Berupa *drop down list* yang akan memberikan beberapa pilihan syarat atau kriteria pada setiap parameter yang harus diproses dalam sistem.
- Drop down list* yang memberikan pilihan berupa kelas yang akan diproses dalam sistem sehingga akan memberikan hasil rekomendasi beasiswa pada kelas tersebut.
- Drop down list* yang memberikan pilihan berupa tahun ajaran yang akan diproses dalam sistem sehingga akan memberikan hasil rekomendasi beasiswa pada tahun ajaran tersebut.
- Tombol untuk melakukan proses rekomendasi beasiswa berdasarkan syarat atau kriteria parameter (nomor 1), kelas (nomor 2), dan tahun ajaran kelas (nomor 3) yang telah dipilih.

5. Menampilkan hasil rekomendasi beasiswa dalam bentuk tabel dan dapat *download* dalam bentuk *microsoft excel*.

Untuk perancangan antarmuka proses pengujian perbandingan antara hasil rekomendasi sistem dengan data aktual ditunjukkan oleh gambar 3.22.

Upload File Excel :

Choose File No File Chosen 1

Upload

**Tabel Hasil Upload Excel**

2

**Tabel Data yang Sama antara Sistem dengan Data Upload**

3

4

Terdapat sebanyak ..... data yang sama dari ..... data yang dibandingkan  
Sehingga Presentase Kecocokan sistem dengan data Aktual dari kelas ..... dan tahun ajaran ..... adalah sebesar ..... %

**Gambar 3.22** Rancangan Antarmuka Proses Pengujian Perbandingan

Keterangan :

1. Merupakan tampilan yang berfungsi untuk melakukan proses upload data aktual dalam bentuk *microsoft excel*.
2. Menampilkan data hasil upload file ke dalam bentuk tabel.
3. Menampilkan jumlah data yang cocok atau sama antara hasil rekomendasi sistem dengan data aktual dalam bentuk tabel.
4. Keterangan mengenai hasil pengujian perbandingan hasil rekomendasi sistem terhadap data aktual.

### 3.5 Perhitungan Manual

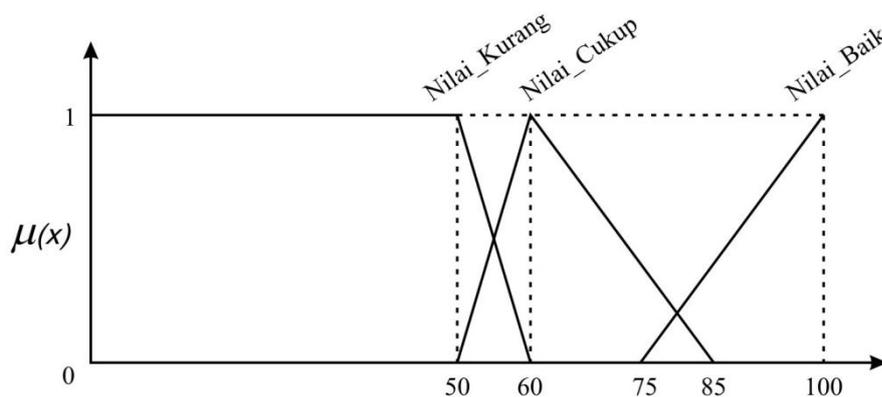
Sebelum melakukan perhitungan manual pada penelitian ini, dibuat batasan-batasan yang akan digunakan dalam membuat kurva fungsi keanggotaan dan menghitung derajat keanggotaan untuk masing-masing parameter yang digunakan dalam proses rekomendasi beasiswa. Nilai batasan yang telah ditentukan ini merupakan hasil diskusi penulis dengan pihak SMA Negeri 7 Malang, sehingga penulis hanya mengimplementasikan nilai batasan pada sistem rekomendasi beasiswa. Nilai batasan pada setiap parameter ditunjukkan pada tabel 3.9.

**Tabel 3.9** Nilai Batasan Setiap Parameter

Parameter	Kriteria	Batasan
Nilai Siswa	Nilai_Kurang	0 - 60
	Nilai_Cukup	50 - 85
	Nilai_Baik	75- 100
Penghasilan Orang Tua	Kurang_Mampu	0 - Rp. 4.000.000
	Mampu	Rp. 2.000.000 - Rp. 6.000.000
	Kaya	$\geq$ Rp. 5.000.000
Kedisiplinan	Disiplin	0 - 50
	Cukup_Disiplin	20 - 80
	Kurang_Disiplin	$\geq$ 60

Dari batasan-batasan setiap parameter yang telah ditentukan, maka dapat dibuat kurva fungsi keanggotaan dari setiap parameter tersebut. Berikut ini adalah kurva derajat keanggotaan untuk masing-masing parameter :

Kurva derajat keanggotaan untuk parameter Nilai siswa ditunjukkan pada gambar 3.23.



**Gambar 3.23** Kurva Derajat Keanggotaan Nilai Siswa

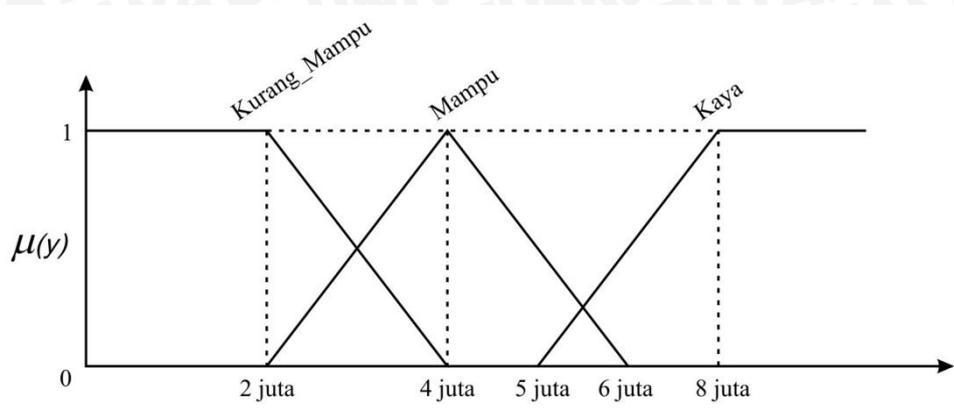
Selanjutnya dari kurva yang telah dibuat, maka dapat ditentukan fungsi derajat keanggotaan untuk parameter Nilai siswa. Berikut ini adalah Fungsi Keanggotaan Nilai :

$$\mu_{\text{Nilai\_Kurang}}[x] = \begin{cases} 0; & x \geq 60 \\ \frac{60-x}{60-50}; & 50 \leq x \leq 60 \\ 1; & x \leq 50 \end{cases}$$

$$\mu_{\text{Nilai\_Cukup}}[x] = \begin{cases} 0; & x \leq 50 \text{ atau } x \geq 85 \\ \frac{x-50}{60-50}; & 50 \leq x \leq 60 \\ \frac{85-x}{85-60}; & 60 \leq x \leq 85 \end{cases}$$

$$\mu_{\text{Nilai\_Baik}}[x] = \begin{cases} 0; & x \leq 75 \\ \frac{x-75}{100-75}; & 75 \leq x \leq 100 \\ 1; & x = 100 \end{cases}$$

Kurva derajat keanggotaan untuk parameter Penghasilan Orang Tua ditunjukkan pada gambar 3.24.



**Gambar 3.24** Kurva Derajat Keanggotaan Penghasilan Orang Tua

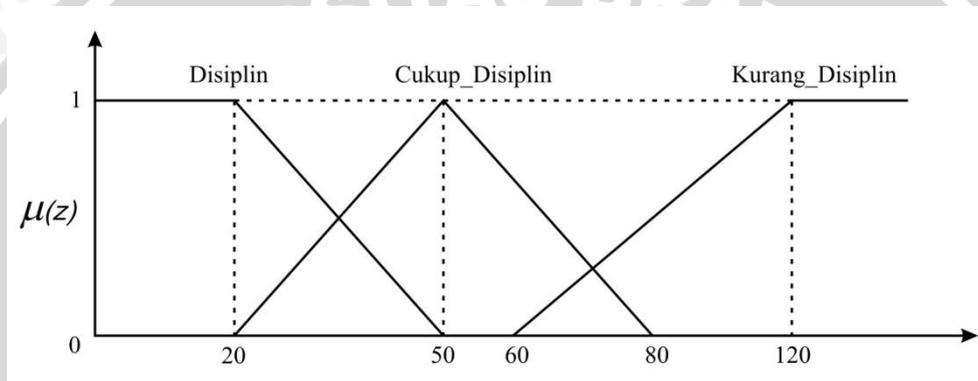
Selanjutnya dari kurva yang telah dibuat, maka dapat ditentukan fungsi derajat keanggotaan untuk parameter Penghasilan Orang Tua. Berikut ini adalah Fungsi Keanggotaan Penghasilan Orang Tua :

$$\mu_{\text{Kurang\_Mampu}} [y] = \begin{cases} 0; & y \geq 4 \text{ juta} \\ \frac{4 \text{ juta} - y}{4 \text{ juta} - 2 \text{ juta}}; & 2 \text{ juta} \leq y \leq 4 \text{ juta} \\ 1; & y \leq 2 \text{ juta} \end{cases}$$

$$\mu_{\text{Mampu}} [y] = \begin{cases} 0; & y \leq 2 \text{ juta} \text{ atau } y \geq 6 \text{ juta} \\ \frac{y - 2 \text{ juta}}{4 \text{ juta} - 2 \text{ juta}}; & 2 \text{ juta} \leq y \leq 4 \text{ juta} \\ \frac{6 \text{ juta} - y}{6 \text{ juta} - 4 \text{ juta}}; & 4 \text{ juta} \leq y \leq 6 \text{ juta} \end{cases}$$

$$\mu_{Kaya} [y] = \begin{cases} 0; & y \leq 5 \text{ juta} \\ \frac{y - 5 \text{ juta}}{8 \text{ juta} - 5 \text{ juta}}; & 5 \text{ juta} \leq y \leq 8 \text{ juta} \\ 1; & y \geq 8 \text{ juta} \end{cases}$$

Kurva derajat keanggotaan untuk parameter Kedisiplinan ditunjukkan pada gambar 3.25.



**Gambar 3.25** Kurva Derajat Keanggotaan Kedisiplinan

Selanjutnya dari kurva yang telah dibuat, maka dapat ditentukan fungsi derajat keanggotaan untuk parameter Kedisiplinan. Berikut ini adalah Fungsi Keanggotaan Kedisiplinan :

$$\mu_{Disiplin} [z] = \begin{cases} 0; & z \geq 50 \\ \frac{50 - z}{50 - 20}; & 20 \leq z \leq 50 \\ 1; & z \leq 20 \end{cases}$$

$$\mu_{Cukup\_Disiplin}[z] = \begin{cases} 0; & z \leq 20 \text{ atau } z \geq 80 \\ \frac{z-20}{80-20}; & 20 \leq z \leq 50 \\ \frac{80-z}{80-50}; & 50 \leq z \leq 80 \end{cases}$$

$$\mu_{Kurang\_Disiplin}[z] = \begin{cases} 0; & z \leq 60 \\ \frac{z-60}{120-60}; & 60 \leq z \leq 120 \\ 1; & z \geq 120 \end{cases}$$

Pada tabel 3.10 menunjukkan beberapa siswa yang memiliki nilai, penghasilan orang tua, dan nilai kedisiplinan tertentu. Kemudian dari beberapa contoh data siswa yang diambil dari *datawarehouse* tersebut akan dihitung derajat keanggotaan setiap parameter dan selanjutnya dihitung *firesterngth* dari setiap siswa untuk mengetahui berapa besar nilai rekomendasi yang diberikan sistem kepada setiap siswa.

**Tabel 3.10** Sampel Data Siswa dan Nilai Dari Setiap Parameter

NIS	NAMA	RATA-RATA NILAI	PENGHASILAN ORANG TUA	KEDISIPLINAN
10087	CICI MAULIDYA S.	82,34	Rp 3.378.000	15
10088	FATWA ANBIYAA	80,54	Rp 2.066.000	15
10091	ILHAM ROSYIDI	87,08	Rp 3.875.000	20
10095	JANUAR WAHYU R.	78,22	Rp 2.780.000	40
10104	KRESNA DHIPAYANA P.	77,45	Rp 3.200.000	25

10108	LAILATUL QUDSIAH	88,70	Rp 3.050.000	20
10120	IRA HAYANI	83,34	Rp 3.780.000	15
10199	NABILA RAHMANTIKA	76,33	Rp 3.100.000	45
10249	RISKY WULANDARI	71,45	Rp 5.906.000	40
10153	M. FIRMANSYAH PUTRA	70,04	Rp 4.200.000	75
10164	M.SANDHYKA WAHYUDI	75,60	Rp 3.957.000	15
10310	YUDI HERMAWAN	70,84	Rp 3.400.000	20
10170	MUHAMAD NAFI RIZALDI	85,42	Rp 2.860.000	15
10178	MUHAMMAD KEVIN A.	88,32	Rp 8.500.000	25
10061	ANITA KUSUMA DEWI	78,85	Rp 3.800.000	45

Untuk menghitung derajat keanggotaan dari setiap parameter yang menjadi acuan untuk proses rekomendasi beasiswa, maka terlebih dahulu ditentukan kriteria dari ketiga parameter yang akan digunakan sebagai pertimbangan dalam proses rekomendasi beasiswa. Misalnya, dengan membuat suatu kriteria untuk Rata-Rata Nilai Mahasiswa yang termasuk dalam NILAI\_BAIK. Kemudian Penghasilan Orang Tua yang termasuk dalam KURANG\_MAMPU. Dan Tingkat Kedisiplinan yang mencakup DISIPLIN. Dari ketiga kriteria atau parameter yang telah ditentukan, maka dapat dihitung derajat keanggotaan setiap parameter berdasarkan kriteria-kriteria tersebut. Sehingga syarat dalam menghitung derajat keanggotaan adalah sesuai dengan kriteria yang ditentukan.

$$\mu_{\text{Nilai_Baik}} [x] = \begin{cases} 0; & x \leq 75 \\ \frac{x - 75}{100 - 75}; & 75 \leq x \leq 100 \\ 1; & x = 100 \end{cases}$$

$$\mu_{\text{Kurang\_Mampu}} [y] = \begin{cases} 0; & y \geq 4 \text{ juta} \\ \frac{4 \text{ juta} - y}{4 \text{ juta} - 2 \text{ juta}}; & 2 \text{ juta} \leq y \leq 4 \text{ juta} \\ 1; & y \leq 2 \text{ juta} \end{cases}$$

$$\mu_{\text{Disiplin}} [z] = \begin{cases} 0; & z \geq 50 \\ \frac{50 - z}{50 - 20}; & 20 \leq z \leq 50 \\ 1; & z \leq 20 \end{cases}$$

**Menghitung derajat keanggotaan :**

1. "CICI MAULIDYA S.", dengan Rata-Rata Nilai (x) = 82,34  
 Jumlah Penghasilan Orang Tua (y) = Rp 3.378.000  
 Nilai Kedisiplinan (z) = 15

$$\begin{aligned} \mu_{\text{Nilai\_Baik}} [x] &= \frac{x - 75}{100 - 75} \quad 75 \leq x \leq 100 \\ &= \frac{82,34 - 75}{25} \\ &= 0,2936 \end{aligned}$$

$$\begin{aligned} \mu_{\text{Kurang\_Mampu}} [y] &= \frac{4.000.000 - y}{4.000.000 - 2.000.000} \quad 2.000.000 \leq y \leq 4.000.000 \\ &= \frac{4.000.000 - 3.378.000}{2.000.000} \\ &= 0,311 \end{aligned}$$

$$\mu_{\text{Disiplin}} [z] = 1,00 \quad z \leq 30$$

Dari hasil perhitungan tersebut didapatkan nilai derajat keanggotaan untuk setiap parameter dari siswa “CICI MAULIDYA S.”. Selanjutnya dihitung nilai *firestrenght* dari derajat keanggotaan yang telah diketahui, dengan menggunakan operasi AND ( $\cap$ ).

$$\begin{aligned} \mu_{\text{Nilai\_Baik}} \cap \mu_{\text{Kurang\_Mampu}} \cap \mu_{\text{Disiplin}} &= \min (\mu_{\text{Nilai\_Baik}}[x]; \mu_{\text{Kurang\_Mampu}}[y]; \mu_{\text{Disiplin}}[z]) \\ &= \min (0,2936 ; 0,311 ; 1,00) \\ &= 0,2936 \end{aligned}$$

Jadi Nilai *Firestrenght* dari siswa dengan nama “CICI MAULIDYA S.” adalah sebesar **0,2936**.

2. “FATWA ANBIYAA.”, dengan Rata-Rata Nilai (x) = 80,54  
Jumlah Penghasilan Orang Tua (y) = Rp 2.066.000  
Nilai Kedisiplinan (z) = 15

$$\begin{aligned} \mu_{\text{Nilai\_Baik}} [x] &= \frac{x - 75}{100 - 75} \quad 75 \leq x \leq 100 \\ &= \frac{80,54 - 75}{25} \\ &= 0,2216 \end{aligned}$$

$$\mu_{\text{Kurang\_Mampu}} [y] = 1,00 \quad y \leq 3 \text{ juta}$$

$$\mu_{\text{Disiplin}} [z] = 1,00 \quad z \leq 30$$

Dari hasil perhitungan tersebut didapatkan nilai derajat keanggotaan untuk setiap parameter dari siswa “FATWA ANBIYAA”. Selanjutnya dihitung nilai *firestrenght* dari derajat keanggotaan yang telah diketahui, dengan menggunakan operasi AND ( $\cap$ ).

$$\begin{aligned}
 \mu_{\text{Nilai\_Baik}} \cap \mu_{\text{Kurang\_Mampu}} \cap \mu_{\text{Disiplin}} &= \min (\mu_{\text{Nilai\_Baik}}[x]; \mu_{\text{Kurang\_Mampu}}[y]; \mu_{\text{Disiplin}}[z]) \\
 &= \min (0,2216 ; 1,00 ; 1,00) \\
 &= 0,351
 \end{aligned}$$

Jadi Nilai *Firestrength* dari siswa dengan nama “FATWA ANBIYAA” adalah sebesar **0,2216**.

Selanjutnya, nilai *firestrength* untuk semua siswa dapat dihitung dengan menggunakan metode dan cara yang sama seperti sebelumnya, nilai *firestrength* ini merupakan nilai yang mewakili besarnya rekomendasi dari setiap siswa. Dari perhitungan nilai *firestrength*, didapatkan nilai *firestrength* untuk setiap siswa seperti yang ditunjukkan pada tabel 3.11.

**Tabel 3.11** Nilai *Firestrength*

NIS	NAMA	<i>Firestrength</i>
10087	CICI MAULIDYA S.	<b>0,2936</b>
10088	FATWA ANBIYAA	<b>0,2216</b>
10091	ILHAM ROSYIDI	<b>0,125</b>
10095	JANUAR WAHYU R.	<b>0,274</b>
10104	KRESNA DHIPAYANA P.	<b>0,248</b>
10108	LAILATUL QUDSIAH	<b>0,623</b>
10120	IRA HAYANI	<b>0,220</b>
10199	NABILA RAHMANTIKA	<b>0,211</b>
10249	RISKY WULANDARI	<b>0</b>
10153	M. FIRMANSYAH PUTRA	<b>0</b>
10164	M.SANDHYKA WAHYUDI	<b>0,043</b>

10310	YUDI HERMAWAN	<b>0,028</b>
10170	MUHAMAD NAFT' RIZALDI	<b>0,514</b>
10178	MUHAMMAD KEVIN A.	<b>0</b>
10061	ANITA KUSUMA DEWI	<b>0,200</b>

Dari hasil *firestrength* dari setiap siswa yang telah diketahui, kemudian dihitung presentase nilai *firestrength* tersebut. Hal ini digunakan untuk mengetahui seberapa besar presentase dari setiap siswa dalam memperoleh beasiswa. Untuk menghitung presentase dari setiap siswa dapat menggunakan rumus :

$$\text{Nilai\_Rekomendasi(\%)} = \frac{\text{Nilai\_Firestrength Setiap Siswa}}{\text{Jumlah Nilai\_Firestrength Seluruh Siswa}} \times 100\%$$

Sehingga didapatkan hasil rekomendasi beasiswa yang ditunjukkan pada tabel 3.12.

**Tabel 3.12** Tabel Hasil Rekomendasi Beasiswa

NIS	NAMA	<i>Firestrength</i>	Nilai Rekomendasi
10087	CICI MAULIDYA S.	0,2936	9,78%
10088	FATWA ANBIYAA	0,2216	7,38%
10091	ILHAM ROSYIDI	0,125	4,17%
10095	JANUAR WAHYU R.	0,274	9,13%
10104	KRESNA DHIPAYANA P.	0,248	8,26%
10108	LAILATUL QUDSIAH	0,623	20,76%
10120	IRA HAYANI	0,22	7,33%

10199	NABILA RAHMANTIKA	0,211	7,03%
10249	RISKY WULANDARI	0	0,00%
10153	M. FIRMANSYAH PUTRA	0	0,00%
10164	M.SANDHYKA WAHYUDI	0,043	1,43%
10310	YUDI HERMAWAN	0,028	0,93%
10170	MUHAMAD NAFI' RIZALDI	0,514	17,13%
10178	MUHAMMAD KEVIN A.	0	0,00%
10061	ANITA KUSUMA DEWI	0,2	6,66%

### 3.6 Rancangan Pengujian Penelitian

Pada penelitian ini akan dilakukan empat jenis proses pengujian, yang pertama adalah dengan melakukan proses perbandingan hasil rekomendasi sistem terhadap data aktual. Kemudian yang kedua adalah *integration testing*, yang berfungsi untuk mengetahui waktu respon yang digunakan dalam proses eksekusi skema ETL dari tabel dimensi dan tabel fakta pada *datawarehouse*. Ketiga adalah pengujian *black box*, pengujian ini merupakan pengujian yang dilakukan untuk menguji fungsional sistem, apakah sistem yang dibuat sudah dapat berjalan dengan baik atau tidak. Dan yang terakhir adalah pengujian *User Acceptance*, pengujian ini dilakukan dengan menggunakan metode kuesioner yang akan diberikan kepada pengguna untuk menilai performa dari sistem yang telah dibuat.

#### 3.6.1 Pengujian Perbandingan Hasil Sistem dengan Data Aktual

Pengujian perbandingan ini dilakukan dengan cara membandingkan data hasil rekomendasi sistem dengan data aktual, kemudian dihitung berapa jumlah data yang sama dari jumlah data yang diproses antara hasil rekomendasi sistem dengan data aktual. Hasil dari pengujian perbandingan berupa nilai presentase (%).

Tabel hasil rekomendasi beasiswa yang telah diproses oleh sistem ditunjukkan pada tabel 3.13, sedangkan untuk data aktual hasil rekomendasi beasiswa ditunjukkan oleh tabel 3.14.

**Tabel 3.13** Hasil Rekomendasi Beasiswa oleh Sistem

NIS	NAMA	Firestrength	Nilai Rekomendasi
10087	CICI MAULIDYA S.	0,2936	9,78%
10088	FATWA ANBIYAA	0,2216	7,38%
10091	ILHAM ROSYIDI	0,125	4,17%
10095	JANUAR WAHYU R.	0,274	9,13%
10104	KRESNA DHIPAYANA P.	0,248	8,26%
10108	LAILATUL QUDSIAH	0,623	20,76%
10120	IRA HAYANI	0,22	7,33%
10199	NABILA RAHMANTIKA	0,211	7,03%
10249	RISKY WULANDARI	0	0,00%
10153	M. FIRMANSYAH PUTRA	0	0,00%
10164	M.SANDHYKA WAHYUDI	0,043	1,43%
10310	YUDI HERMAWAN	0,028	0,93%
10170	MUHAMAD NAFT' RIZALDI	0,514	17,13%
10178	MUHAMMAD KEVIN A.	0	0,00%
10061	ANITA KUSUMA DEWI	0,2	6,66%

**Tabel 3.14** Data Aktual Hasil Rekomendasi Beasiswa

<i>Ranking</i>	NAMA	RATA-RATA NILAI	PENGHASILAN ORANG TUA	KEDISIPLINAN
1	CHUSNUL LIYAH	84,33	Rp2.300.000	15
2	DEWI SEKAR MIASIH	85,7	Rp3.110.000	20
3	CICI MAULIDYA S.	82,34	Rp3.378.000	15
4	FATWA ANBIYAA	80,54	Rp2.066.000	15
5	JANUAR WAHYU R.	78,22	Rp2.780.000	40
6	YUDI HERMAWAN	87,48	Rp3.075.000	20
7	IRA HAYANI	83,34	Rp3.780.000	15
8	NABILA RAHMANTIKA	76,33	Rp3.100.000	45
9	ANITA KUSUMA DEWI	78,85	Rp3.800.000	45
10	REZA AINURRACHMAN	72,45	Rp3.260.000	25
11	M.SANDHYKA WAHYUDI	75,6	Rp3.957.000	15
12	YUDI HERMAWAN	70,84	Rp3.400.000	20
13	KARINA PUSPITA SARI	88,20	Rp5.100.000	25
14	RADITYA RIO SAPUTRA	74,15	Rp2.006.000	40
15	M. FIRMANSYAH PUTRA	70,04	Rp4.200.000	75

Dari hasil rekomendasi kedua tabel di atas, dapat disimpulkan bahwa terdapat 6 perbedaan hasil rekomendasi beasiswa dan terdapat 9 data rekomendasi yang cocok, sehingga :

$$\text{Presentase Kecocokan} = \frac{9}{15} \times 100\% = 60\%$$

Dari hasil perhitungan tersebut, dapat disimpulkan bahwa presentase kecocokan hasil perhitungan dengan menggunakan *Fuzzy Database* terhadap data aktual adalah sebesar **60 %**.

### 3.6.2 Integration Testing

*Integration Testing* dilakukan dengan menganalisa hasil *record* atau *log* untuk setiap kali proses yang dilakukan. Dalam melakukan pengujian ini, hal pertama yang harus dilakukan adalah melakukan eksekusi proses ETL pada aplikasi sehingga nantinya akan muncul sebuah *Log* data yang berisi tentang waktu yang diperlukan dalam mengeksekusi jumlah *record* yang ada. Pengujian ini dilakukan terhadap setiap tabel dimensi dan fakta yang terbentuk.

Setelah melakukan proses eksekusi ETL untuk masing-masing tabel *dim\_siswa*, *dim\_tatib*, *dim\_periode*, *dim\_mapel*, *dim\_kelas*, *fakta\_penghasilan\_ortu*, *fakta\_nilai*, dan *fakta\_tatib*, kemudian akan dilakukan perbandingan terhadap waktu respon (waktu eksekusi) setiap tabel dengan jumlah data tertentu. Hasil perbandingan proses lamanya eksekusi untuk masing-masing tabel *dim\_siswa*, *dim\_tatib*, *dim\_periode*, *dim\_mapel*, *dim\_kelas*, *fakta\_penghasilan\_ortu*, *fakta\_nilai*, dan *fakta\_tatib* ditunjukkan oleh tabel 3.15.

**Tabel 3.15** Hasil Perbandingan Waktu Eksekusi ETL setiap Tabel

Banyak Baris Database Operasional	Lama Waktu (s)							
	<i>dim_siswa</i>	<i>dim_kelas</i>	<i>dim_tatib</i>	<i>dim_mapel</i>	<i>dim_periode</i>	<i>fakta_penghasilan_ortu</i>	<i>fakta_tatib</i>	<i>fakta_nilai</i>
.... data								
.... data								

### 3.6.3 Pengujian Black Box

Pengujian *black box* merupakan pengujian yang dilakukan terhadap fungsional sistem yang telah dibuat. Pengujian ini dilakukan untuk mengetahui apakah sistem yang dibuat sudah dapat digunakan secara baik atau tidak. Pengujian *black box* pada sistem dilakukan pada setiap menu atau *service* yang terdapat dalam sistem, selanjutnya dari setiap menu atau *service* tersebut dibuatlah skenario pengujian yang berfungsi untuk melakukan perintah pada sistem tersebut. Kemudian sistem akan melakukan perintah sesuai dengan skenario

pengujian tersebut dan dari perintah itu akan didapatkan sebuah hasil yang telah diproses oleh sistem, sehingga dari hal tersebut akan diketahui apakah sistem mampu melakukan proses skenario pengujian dengan baik atau tidak. Jika ternyata sistem telah mampu memenuhi semua skenario pengujian, maka sistem telah berfungsi dengan baik. Tetapi jika terdapat beberapa skenario pengujian yang gagal dilakukan oleh sistem, maka sistem membutuhkan sebuah perbaikan yang berkaitan dengan menu atau *service* pada sistem tersebut. Rancangan pengujian *black box* ditunjukkan oleh tabel 3.16.

**Tabel 3.16** Rancangan Pengujian *Black Box*

Menu atau <i>Service</i>	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Log In					
Rekomendasi					
Pengujian					
Kelola <i>User</i>					

### 3.6.4 *Acceptance Testing*

*Acceptance testing* digunakan untuk menunjukkan bahwa sistem sudah siap dalam pemakaian operasional dan apakah sistem dapat memenuhi kriteria atau kebutuhan yang diharapkan serta apakah dapat diterima dengan baik. *Acceptance testing* ini dilakukan dengan cara memberikan angket kepada pengguna setelah melakukan *testing* program. Angket-angket pada *acceptance testing* ini ditunjukkan oleh tabel 3.17, tabel 3.18, tabel 3.19, dan tabel 3.20

**Tabel 3.17** Angket Terhadap Kemudahan Dalam Olah Data

Hasil Yang Diharapkan	Hasil Nyata	Kesimpulan
Mudah dalam pengoperasian untuk proses rekomendasi beasiswa serta pengolahan data akademik siswa		<input type="checkbox"/> diterima <input type="checkbox"/> dipertimbangkan <input type="checkbox"/> ditolak

**Tabel 3.18** Angket Terhadap Hasil Rekomendasi Beasiswa

Hasil Yang Diharapkan	Hasil Nyata	Kesimpulan
Menampilkan daftar siswa yang berhak menerima beasiswa		<input type="checkbox"/> diterima <input type="checkbox"/> dipertimbangkan <input type="checkbox"/> ditolak

**Tabel 3.19** Angket Terhadap Tampilan Laporan

Hasil Yang Diharapkan	Hasil Nyata	Kesimpulan
Menampilkan informasi dan laporan data akademik siswa, informasi juga dapat ditampilkan dalam bentuk grafik		<input type="checkbox"/> diterima <input type="checkbox"/> dipertimbangkan <input type="checkbox"/> ditolak

**Tabel 3.20** Angket Terhadap Antarmuka Aplikasi

Hasil Yang Diharapkan	Hasil Nyata	Kesimpulan
Antar muka aplikasi menarik dan mudah untuk digunakan		<input type="checkbox"/> diterima <input type="checkbox"/> dipertimbangkan <input type="checkbox"/> ditolak

## BAB IV IMPLEMENTASI

### 4.1 Lingkungan Implementasi

Lingkungan implementasi sistem meliputi lingkungan perangkat keras (*Hardware*) dan lingkungan perangkat lunak (*software*).

#### 4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan untuk mengimplementasikan Sistem Pendukung Keputusan Rekomendasi Beasiswa sebagai berikut :

1. Processor Intel Core i3-2350M CPU @ 2.30 GHz.
2. *Memory* 4 GB DDR3.
3. *Harddisk* 500 GB
4. VGA Intel(R) HD Graphics (60Hz)
5. Monitor 14".

#### 4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan untuk mengimplementasikan Sistem Pendukung Keputusan Rekomendasi Beasiswa sebagai berikut :

1. Sistem Operasi *Windows 7 Ultimate*
2. *Tools* dari Pentaho Corporation. Pentaho merupakan *software* yang berbasis *open source* yang dapat melakukan proses ETL dan operasi OLAP yang dapat menyelesaikan masalah pada penelitian ini. *Tools* dari Pentaho tersebut diantaranya :
  - a. Pentaho Data Integration (Kettle) v. 4.1.0, untuk melakukan proses ETL.
  - b. Pentaho Schema Workbench (Mondrian) v. 3.2.1.13885 untuk membuat struktur kubus dan sebagai ROLAP *server*.
  - c. Pentaho BIServer Community Edition v. 4.1.0 stabil sebagai *web server* serta *front end user* atau aplikasi *user interface* untuk membuat laporan rekap data.

3. XAMPP 1.7.4
4. phpMyAdmin 3.3.9
5. *Tools programing* : NetBeans IDE 6.9.1

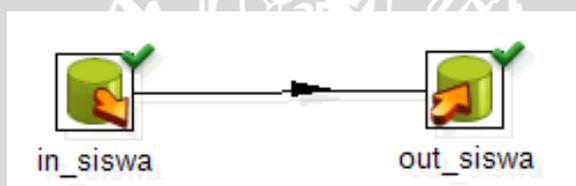
#### 4.2 Implementasi Program

Pada subbab ini akan dibahas mengenai implementasi dari perancangan yang telah dibuat pada bab 3.

##### 4.2.1 Implementasi ETL (*Extraction, Transformation, Loading*)

###### 4.2.1.1 Proses ETL dim\_siswa

Data siswa yang bersumber pada *database* MySQL akan di transformasi melalui proses ETL. Data siswa tersebut akan dimasukkan pada tabel dim\_siswa. Skema ETL dim\_siswa ditunjukkan pada gambar 4.1.



Gambar 4.1 Skema ETL dim\_siswa

Untuk penjelasan dari setiap simbol pada skema ETL dim\_siswa ditunjukkan pada tabel 4.1 dan untuk log dari proses ETL dim\_siswa ditunjukkan pada tabel 4.2.

Tabel 4.1 Keterangan Skema ETL dim\_siswa

Simbol	Keterangan
 in_siswa	Berfungsi untuk mengambil data siswa dari <i>database</i> OLTP.
 out_siswa	Berfungsi untuk melakukan proses <i>loading</i> data hasil transformasi ke dalam tabel dim_siswa pada <i>datawarehouse</i> .

**Tabel 4.2** Log dari Proses ETL dim\_siswa

Stepname	Copy nr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
in_siswa	0	0	1463	1463	0	0	0	0	Finished	0.2s	8,219	-
out_siswa	0	1463	1463	0	1463	0	0	0	Finished	0.7s	2,203	-

Pada tabel 4.2 ditunjukkan log dari proses ETL dim\_siswa, pada *stepname* in\_siswa terdapat keterangan 0 pada *copynr*, artinya jumlah langkah (*step*) yang tersalin sebanyak 0, pada *Written* terdapat nilai 1463, artinya pada *database* OLTP yang menjadi inputan sistem berhasil tertulis sebanyak 1463 baris, pada keterangan *Input* terdapat nilai 14633, artinya jumlah baris yang terbaca dari *database* OLTP sebanyak 1463 baris. Untuk menuliskan data ke *database* pada *stepname* out\_siswa proses ETL ini membutuhkan waktu 0,7 detik dengan kecepatan (*speed*) 2.203 rows/second, jumlah baris yang terbaca dari input sistem (*read*) sebanyak 1463 baris, jumlah baris yang tertulis di keluaran sistem (*written*) sebanyak 1463 baris dan jumlah baris yang tertulis di *database* (*output*) pada tabel dim\_siswa sebanyak 1463 baris.

Hasil proses ETL dim\_siswa yang dilakukan pada *Kettle* ditunjukkan pada gambar 4.2, sedangkan gambar 4.3 menunjukkan proses ETL pada *Kettle* dim\_siswa yang telah dimasukkan kedalam tabel dim\_siswa pada MySQL.

Rows of step: out_siswa (1463 rows)			
#	NIS	NAMA_SISWA	Penghasilan_ortu
1	10000	Arizal Ismoyo Wijanarko	6526000
2	10001	Aldwin Ryan Zakaria	5762000
3	10002	Annissa Kusumawardhani	7917000
4	10003	Annisa Rahmadiyah	5449000
5	10004	Dana Helmi Anggara	1147000
6	10005	Tanti Prahastuti	7670000

**Gambar 4.2** Hasil Proses ETL dim\_siswa pada *Kettle*

+ Options				
		NIS	NAMA_SISWA	Penghasilan_Ortu
<input type="checkbox"/>		10000	Arizal Ismoyo Wijanarko	6526000
<input type="checkbox"/>		10001	Aldwin Ryan Zakaria	5762000
<input type="checkbox"/>		10002	Annissa Kusumawardhani	7917000
<input type="checkbox"/>		10003	Annisa Rahmadiah	5449000
<input type="checkbox"/>		10004	Dana Helmi Anggara	1147000
<input type="checkbox"/>		10005	T. D. I. H.	7070000

Gambar 4.3 Hasil Proses ETL dim\_siswa pada MySQL

#### 4.2.1.2 Proses ETL dim\_kelas

Data kelas yang bersumber pada *database* MySQL akan di transformasi melalui proses ETL. Data kelas tersebut akan dimasukkan pada tabel *dim\_kelas*. Skema ETL *dim\_kelas* ditunjukkan pada gambar 4.4.



Gambar 4.4 Skema ETL dim\_kelas

Untuk penjelasan dari setiap simbol pada skema ETL *dim\_kelas* ditunjukkan pada tabel 4.3 dan untuk Log dari proses ETL *dim\_kelas* ditunjukkan oleh tabel 4.4.

Tabel 4.3 Keterangan Skema ETL dim\_kelas

Simbol	Keterangan
	Berfungsi untuk mengambil data kelas dari <i>database</i> OLTP.
	Berfungsi untuk melakukan proses <i>loading</i> data hasil transformasi ke dalam tabel <i>dim_kelas</i> pada <i>datawarehouse</i> .

**Tabel 4.4** Log dari Proses ETL dim\_kelas

Stepname	Copy nr	Read	Written	In put	Out put	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
in_kelas	0	0	29	29	0	0	0	0	Finished	0.2s	187	-
out_kelas	0	29	29	0	29	0	0	0	Finished	0.3s	93	-

Pada tabel 4.4 ditunjukkan log dari proses ETL dim\_kelas, pada *stepname* in\_kelas terdapat keterangan 0 pada *copynr*, artinya jumlah langkah (*step*) yang tersalin sebanyak 0, pada *Written* terdapat nilai 29, artinya pada *database* OLTP yang menjadi inputan sistem berhasil tertulis sebanyak 29 baris, pada keterangan *Input* terdapat nilai 29, artinya jumlah baris yang terbaca dari *database* OLTP sebanyak 29 baris. Untuk menuliskan data ke *database* pada *stepname* out\_kelas proses ETL ini membutuhkan waktu 0,3 detik dengan kecepatan (*speed*) 93 rows/second, jumlah baris yang terbaca dari input sistem (*read*) sebanyak 29 baris, jumlah baris yang tertulis di keluaran sistem (*written*) sebanyak 29 baris dan jumlah baris yang tertulis di *database* (*output*) pada tabel dim\_kelas sebanyak 29 baris.

Hasil proses ETL dim\_kelas yang dilakukan pada *Kettle* ditunjukkan pada gambar 4.5, sedangkan gambar 4.6 menunjukkan proses ETL pada *Kettle* dim\_kelas yang telah dimasukkan kedalam tabel dim\_siswa pada MySQL.

Rows of step: out_kelas (29 rows)			
#	KD_KELAS	NAMA_KELAS	tingkat
1	0100	X-1	1
2	0200	X-2	1
3	0300	X-3	1
4	0400	X-5	1
5	0500	X-4	1
6	0600	X-7	1

**Gambar 4.5** Hasil Proses ETL dim\_kelas pada *Kettle*

+ Options			
←T→	KD_KELAS	NAMA_KELAS	TINGKAT
<input type="checkbox"/>  	0100	X-1	1
<input type="checkbox"/>  	0200	X-2	1
<input type="checkbox"/>  	0300	X-3	1
<input type="checkbox"/>  	0400	X-5	1
<input type="checkbox"/>  	0500	X-4	1

**Gambar 4.6** Hasil Proses ETL dim\_kelas pada MySQL

### 4.2.1.3 Proses ETL dim\_mapel

Data mata pelajaran yang bersumber pada *database* MySQL akan di transformasi melalui proses ETL. Data mata pelajaran tersebut akan dimasukkan pada tabel dim\_kelas. Skema ETL dim\_mapel ditunjukkan pada gambar 4.7.



**Gambar 4.7** Skema ETL dim\_mapel

Untuk penjelasan dari setiap simbol pada skema ETL dim\_mapel ditunjukkan pada tabel 4.5 dan untuk Log dari proses ETL dim\_mapel ditunjukkan oleh tabel 4.6.

**Tabel 4.5** Keterangan Skema ETL dim\_mapel

Simbol	Keterangan
 in_mapel	Berfungsi untuk mengambil data mata pelajaran dari <i>database</i> OLTP.
 out_mapel	Befungsi untuk melakukan proses <i>loading</i> data hasil transformasi ke dalam tabel dim_mapel pada <i>datawarehouse</i> .

**Tabel 4.6** Log dari Proses ETL dim\_mapel

Stepname	Copynr	Read	Written	In put	Out put	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
in_mapel	0	0	15	15	0	0	0	0	Finished	0.1s	183	-
out_mapel	0	15	15	0	15	0	0	0	Finished	0.2s	63	-

Pada tabel 4.6 ditunjukkan log dari proses ETL dim\_mapel, pada *stepname* in\_mapel terdapat keterangan 0 pada *copynr*, artinya jumlah langkah (*step*) yang tersalin sebanyak 0, pada *Written* terdapat nilai 15, artinya pada *database* OLTP yang menjadi inputan sistem berhasil tertulis sebanyak 15 baris, pada keterangan *Input* terdapat nilai 15, artinya jumlah baris yang terbaca dari *database* OLTP sebanyak 15 baris. Untuk menuliskan data ke *database* pada *stepname* out\_mapel proses ETL ini membutuhkan waktu 0,2 detik dengan kecepatan (*speed*) 63 rows/second, jumlah baris yang terbaca dari input sistem (*read*) sebanyak 15 baris, jumlah baris yang tertulis di keluaran sistem (*written*) sebanyak 29 baris dan jumlah baris yang tertulis di *database* (*output*) pada tabel dim\_mapel sebanyak 29 baris.

Hasil proses ETL dim\_mapel yang dilakukan pada *Kettle* ditunjukkan pada gambar 4.8, sedangkan gambar 4.9 menunjukkan proses ETL pada *Kettle* dim\_mapel yang telah dimasukkan kedalam tabel dim\_mapel pada MySQL.

Rows of step: out_mapel (15 rows)		
#	KD_MAPEL	NAMA_MAPEL
1	01	Pendidikan Agama
2	02	Kewarganegaraan
3	03	Bahasa Indonesia
4	04	Bahasa Inggris
5	05	Matematika
6	06	Fisika
7	07	Pendidikan Kewarganegaraan

**Gambar 4.8** Hasil Proses ETL dim\_mapel pada *Kettle*

+ Options			KD_MAPEL	NAMA_MAPEL
<input type="checkbox"/>			01	Pendidikan Agama
<input type="checkbox"/>			02	Kewarganegaraan
<input type="checkbox"/>			03	Bahasa Indonesia
<input type="checkbox"/>			04	Bahasa Inggris
<input type="checkbox"/>			05	Matematika
<input type="checkbox"/>			06	Fisika
<input type="checkbox"/>			07	Biologi

**Gambar 4.9** Hasil Proses ETL dim\_mapel pada MySQL

#### 4.2.1.4 Proses ETL dim\_periode

Data periode (semsetr dan tahun ajaran) yang bersumber pada *database* MySQL akan di transformasi melalui proses ETL. Data periode tersebut akan dimasukkan pada tabel *dim\_periode*. Skema ETL *dim\_periode* ditunjukkan pada gambar 4.10.



**Gambar 4.10** Skema ETL *dim\_periode*

Untuk penjelasan dari setiap simbol pada skema ETL *dim\_periode* ditunjukkan pada tabel 4.7 dan untuk Log dari proses ETL *dim\_periode* ditunjukkan oleh tabel 4.8.

**Tabel 4.7** Keterangan Skema ETL *dim\_periode*

Simbol	Keterangan
 in_periode	Berfungsi untuk mengambil data periode dari <i>database</i> OLTP.
 out_periode	Berfungsi untuk melakukan proses <i>loading</i> data hasil transformasi ke dalam tabel <i>dim_periode</i> pada <i>datawarehouse</i> .

**Tabel 4.8** Log dari Proses ETL dim\_periode

Stepname	Copy nr	Read	Written	In put	Out put	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
in_periode	0	0	4	4	0	0	0	0	Finished	0.6s	6	-
out_periode	0	4	4	0	4	0	0	0	Finished	0.6s	6	-

Pada tabel 4.6 ditunjukkan log dari proses ETL dim\_periode, pada *stepname* in\_periode terdapat keterangan 0 pada *copynr*, artinya jumlah langkah (*step*) yang tersalin sebanyak 0, pada *Written* terdapat nilai 4, artinya pada *database* OLTP yang menjadi inputan sistem berhasil tertulis sebanyak 4 baris, pada keterangan *Input* terdapat nilai 4, artinya jumlah baris yang terbaca dari *database* OLTP sebanyak 4 baris. Untuk menuliskan data ke *database* pada *stepname* out\_periode proses ETL ini membutuhkan waktu 0,6 detik dengan kecepatan (*speed*) 6 rows/second, jumlah baris yang terbaca dari input sistem (*read*) sebanyak 4 baris, jumlah baris yang tertulis di keluaran sistem (*written*) sebanyak 4 baris dan jumlah baris yang tertulis di *database* (*output*) pada tabel dim\_periode sebanyak 4 baris.

Hasil proses ETL dim\_periode yang dilakukan pada *Kettle* ditunjukkan pada gambar 4.11, sedangkan gambar 4.12 menunjukkan proses ETL pada *Kettle* dim\_periode yang telah dimasukkan kedalam tabel dim\_periode pada MySQL.

Rows of step: out_periode (4 rows)		
#	thn_ajaran	semester
1	2010/2011	1
2	2010/2011	2
3	2011/2012	1
4	2011/2012	2

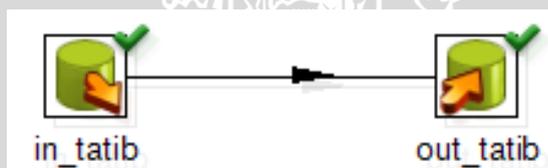
**Gambar 4.11** Hasil Proses ETL dim\_periode pada *Kettle*

+ Options			id_periode	thn_ajaran	semester
<input type="checkbox"/>			1	2010/2011	1
<input type="checkbox"/>			2	2010/2011	2
<input type="checkbox"/>			3	2011/2012	1
<input type="checkbox"/>			4	2011/2012	2

**Gambar 4.12** Hasil Proses ETL dim\_periode pada MySQL

#### 4.2.1.5 Proses ETL dim\_tatib

Data tata tertib yang bersumber pada *database* MySQL akan di transformasi melalui proses ETL. Data periode tersebut akan dimasukkan pada tabel dim\_tatib. Skema ETL dim\_tatib ditunjukkan pada gambar 4.13.



**Gambar 4.13** Skema ETL dim\_tatib

Untuk penjelasan dari setiap simbol pada skema ETL dim\_tatib ditunjukkan pada tabel 4.9 dan untuk Log dari proses ETL dim\_tatib ditunjukkan oleh tabel 4.10.

**Tabel 4.9** Keterangan Skema ETL dim\_tatib

Simbol	Keterangan
 in_tatib	Berfungsi untuk mengambil data tata tertib dari <i>database</i> OLTP.
 out_tatib	Berfungsi untuk melakukan proses <i>loading</i> data hasil transformasi ke dalam tabel dim_tatib pada <i>datawarehouse</i> .

**Tabel 4.10** Log dari Proses ETL dim\_tatib

Stepname	Copy nr	Read	Written	In put	Out put	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
in_tatib	0	0	35	35	0	0	0	0	Finished	0.2s	226	-
out_tatib	0	35	35	0	35	0	0	0	Finished	0.3s	111	-

Pada tabel 4.10 ditunjukkan log dari proses ETL dim\_tatib, pada *stepname* in\_tatib terdapat keterangan 0 pada *copynr*, artinya jumlah langkah (*step*) yang tersalin sebanyak 0, pada *Written* terdapat nilai 35, artinya pada *database* OLTP yang menjadi inputan sistem berhasil tertulis sebanyak 35 baris, pada keterangan *Input* terdapat nilai 35, artinya jumlah baris yang terbaca dari *database* OLTP sebanyak 35 baris. Untuk menuliskan data ke *database* pada *stepname* out\_tatib proses ETL ini membutuhkan waktu 0,3 detik dengan kecepatan (*speed*) 111 rows/second, jumlah baris yang terbaca dari input sistem (*read*) sebanyak 35 baris, jumlah baris yang tertulis di keluaran sistem (*written*) sebanyak 35 baris dan jumlah baris yang tertulis di *database* (*output*) pada tabel dim\_tatib sebanyak 35 baris.

Hasil proses ETL dim\_tatib yang dilakukan pada *Kettle* ditunjukkan pada gambar 4.14, sedangkan gambar 4.15 menunjukkan proses ETL dim\_tatib pada *Kettle* yang telah dimasukkan kedalam tabel dim\_tatib pada MySQL.

Rows of step: out_tatib (35 rows)			
#	KD_TATIB	NAMA_TATIB	POIN
1	1	Mencuri/ merampas barang milik orang lain	100
2	10	Merusak sarana/prasarana milik sekolah atau warga	50
3	11	Merokok atau membawa rokok di sekolah	50
4	12	Memalsu tanda tangan orang tua/wali, kepala sekolah	50
5	13	Memalsukan stempel sekolah	50
6	14	Membuat pernyataan dusta atau kebohongan palsu	50

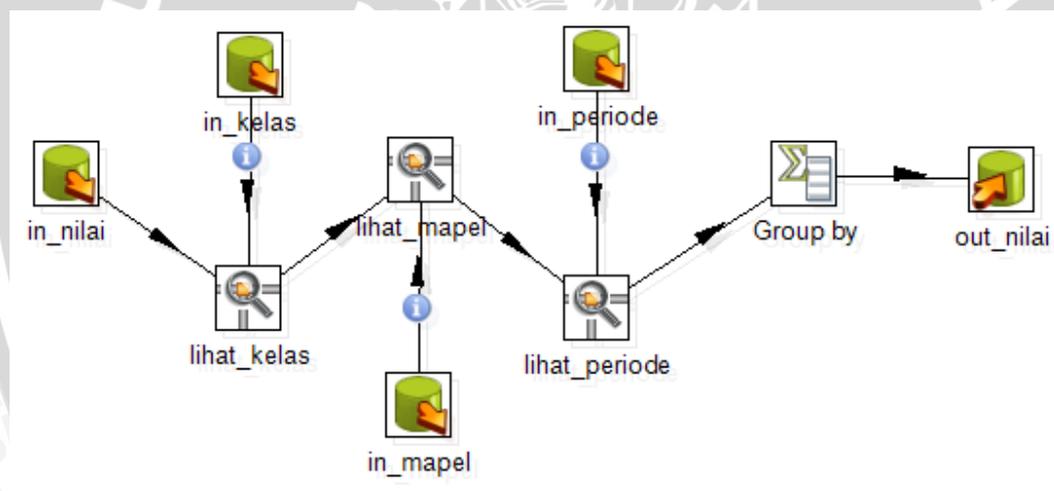
**Gambar 4.14** Hasil Proses ETL dim\_tatib pada *Kettle*

+ Options					
			KD_TATIB	NAMA_TATIB	POIN
<input type="checkbox"/>			1	Mencuri/ merampas barang milik orang lain	100
<input type="checkbox"/>			10	Merusak sarana/prasarana milik sekolah atau warga	50
<input type="checkbox"/>			11	Merokok atau membawa rokok di sekolah	50
<input type="checkbox"/>			12	Memalsu tanda tangan orang tua/wali, kepala sekola	50
<input type="checkbox"/>			13	Memalsukan stempel sekolah	50
<input type="checkbox"/>			14	Membuat pernyataan dusta atau kesaaksian palsu	50

Gambar 4.15 Hasil Proses ETL dim\_tatib pada MySQL

#### 4.2.1.6 Proses ETL fact\_nilai

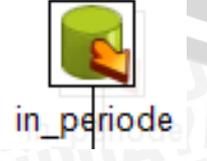
Data nilai yang bersumber pada *database* MySQL akan di transformasi melalui proses ETL. Data nilai tersebut akan dimasukkan pada tabel fact\_nilai. Skema ETL fact\_nilai ditunjukkan pada gambar 4.16.



Gambar 4.16 Skema ETL fact\_nilai

Untuk penjelasan dari setiap simbol pada skema ETL fact\_nilai ditunjukkan pada tabel 4.11 dan untuk Log dari proses ETL fact\_nilai ditunjukkan oleh tabel 4.12.

**Tabel 4.11** Keterangan Skema ETL fact\_nilai

Simbol	Keterangan
 <p>in_nilai</p>	<p>Berfungsi untuk mengambil data nilai dari <i>database</i> OLTP.</p>
 <p>lihat_kelas</p>	<p>Befungsi untuk menghubungkan atribut nama_kelas dan tingkat pada in_nilai dengan atribut nama_kelas dan tingkat pada dim_kelas, agar dapat mengambil primary key pada tabel dim_kelas.</p>
 <p>in_kelas</p>	<p>Berfungsi untuk mengambil data kelas dari dim_siswa pada <i>datawarehouse</i>.</p>
 <p>lihat_mapel</p>	<p>Befungsi untuk menghubungkan atribut nama_mapel pada in_nilai dengan atribut nama_mapel pada dim_mapel, agar dapat mengambil primary key pada tabel dim_mapel.</p>
 <p>in_mapel</p>	<p>Berfungsi untuk mengambil data mata pelajaran dari dim_mapel pada <i>datawarehouse</i>.</p>
 <p>lihat_periode</p>	<p>Befungsi untuk menghubungkan atribut thn_ajaran dan semester pada in_nilai dengan atribut thn_ajaran dan semester pada dim_periode, agar dapat mengambil primary key pada tabel dim_periode.</p>
 <p>in_periode</p>	<p>Berfungsi untuk mengambil data periode dari dim_periode pada <i>datawarehouse</i>.</p>
 <p>Group by</p>	<p>Berfungsi untuk mengelompokkan data nilai berdasarkan nis, kd_mapel, kd_kelas, id_periode, dan tingkat, serta mengagregasikan nilai dengan fungsi <i>average</i>.</p>

 out_nilai	<p>Berfungsi untuk memasukkan data hasil transformasi ke dalam tabel fact_nilai pada <i>datawarehouse</i>.</p>
--	--

**Tabel 4.12** Log dari Proses ETL fact\_nilai

Stepname	Copy nr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
in_nilai	0	0	57054	57054	0	0	0	0	Finished	3.1s	18,682	-
lihat_kelas	0	57083	57054	0	0	0	0	0	Finished	3.1s	18,673	-
in_kelas	0	0	29	29	0	0	0	0	Finished	0.0s	806	-
lihat_mapel	0	57069	57054	0	0	0	0	0	Finished	4.0s	14,408	-
in_mapel	0	0	15	15	0	0	0	0	Finished	0.1s	214	-
lihat_periode	0	57058	57054	0	0	0	0	0	Finished	5.2s	11,071	-
in_periode	0	0	4	4	0	0	0	0	Finished	0.1s	73	-
Group by	0	57054	57054	0	0	0	0	0	Finished	6.5s	8,740	-
out_nilai	0	57054	57054	0	57054	0	0	0	Finished	7.8s	7,323	-

Pada tabel 4.12 ditunjukkan log dari proses ETL fact\_nilai, pada *stepname* in\_nilai terdapat keterangan 0 pada *copynr*, artinya jumlah langkah (*step*) yang tersalin sebanyak 0, pada *Written* terdapat nilai 57054, artinya pada *database* OLTP yang menjadi inputan sistem berhasil tertulis sebanyak 57054 baris, pada keterangan *Input* terdapat nilai 57054, artinya jumlah baris yang terbaca dari *database* OLTP sebanyak 57054 baris. Untuk menuliskan data ke *database* pada *stepname* out\_nilai proses ETL ini membutuhkan waktu 7,8 detik dengan kecepatan (*speed*) 7.323 rows/second, jumlah baris yang terbaca dari input sistem (*read*) sebanyak 57054 baris, jumlah baris yang tertulis di keluaran sistem (*written*) sebanyak 57054 baris dan jumlah baris yang tertulis di *database* (*output*) pada tabel fact\_nilai sebanyak 57054 baris.

Hasil proses ETL fact\_nilai yang dilakukan pada *Kettle* ditunjukkan pada gambar 4.17, sedangkan gambar 4.18 menunjukkan proses ETL fact\_nilai pada *Kettle* yang telah dimasukkan kedalam tabel fact\_nilai pada MySQL.

Rows of step: out\_nilai (57054 rows)

#	nis	KD_KELAS	KD_MAPEL	id_periode	tingkat	Nilai
1	10019	0100	01	1	1	81
2	10019	0100	01	2	1	92
3	10019	0100	02	1	1	81
4	10019	0100	02	2	1	82
5	10019	0100	03	1	1	84
6	10019	0100	03	2	1	88
7	10019	0100	04	1	1	92
8	10019	0100	04	2	1	86

Gambar 4.17 Hasil Proses ETL fact\_nilai pada *Kettle*

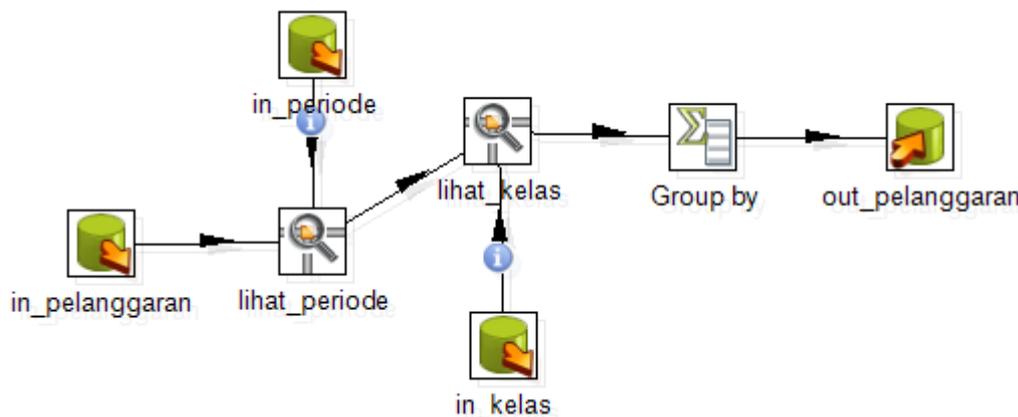
+ Options

	nis	KD_KELAS	KD_MAPEL	id_periode	tingkat	Nilai
<input type="checkbox"/>	10019	0100	01	1	1	81
<input type="checkbox"/>	10019	0100	01	2	1	92
<input type="checkbox"/>	10019	0100	02	1	1	81
<input type="checkbox"/>	10019	0100	02	2	1	82
<input type="checkbox"/>	10019	0100	03	1	1	84
<input type="checkbox"/>	10019	0100	03	2	1	88
<input type="checkbox"/>	10019	0100	04	1	1	92
<input type="checkbox"/>	10019	0100	04	2	1	86

Gambar 4.18 Hasil Proses ETL fact\_nilai pada MySQL

#### 4.2.1.7 Proses ETL fact\_tatib

Data pelanggaran tata tertib yang bersumber pada *database* MySQL akan di transformasi melalui proses ETL. Data pelanggaran tata tertib tersebut akan dimasukkan pada tabel fact\_tatib. Skema ETL fact\_tatib ditunjukkan pada gambar 4.19.

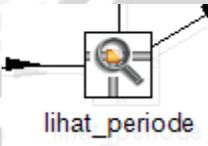
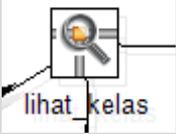


Gambar 4.19 Skema ETL fact\_tatib



Untuk penjelasan dari setiap simbol pada skema ETL fact\_tatib ditunjukkan pada tabel 4.13 dan untuk Log dari proses ETL fact\_nilai ditunjukkan oleh tabel 4.14.

**Tabel 4.13** Keterangan Skema ETL fact\_tatib

Simbol	Keterangan
 in_pelanggaran	Berfungsi untuk mengambil data pelanggaran tata tertib dari <i>database</i> OLTP.
 lihat_periode	Berfungsi untuk menghubungkan thn_ajaran pada in_pelanggaran dengan atribut thn_ajaran pada dim_periode, agar dapat mengambil primary key pada tabel dim_periode.
 in_periode	Berfungsi untuk mengambil data periode dari dim_periode pada <i>datawarehouse</i> .
 lihat_kelas	Berfungsi untuk menghubungkan atribut nama_kelas pada in_pelanggaran dengan atribut nama_kelas pada dim_kelas, agar dapat mengambil primary key pada tabel dim_kelas.
 in_kelas	Berfungsi untuk mengambil data kelas dari dim_kelas pada <i>datawarehouse</i> .
 Group by	Berfungsi untuk mengelompokkan data pelanggaran tata tertib berdasarkan nis, kd_kelas, kd_tatib, dan id_periode, serta mengagregasikan poin pelanggaran dengan fungsi <i>sum</i> .
 out_pelanggaran	Berfungsi untuk memasukkan data hasil transformasi ke dalam tabel fact_tatib pada <i>datawarehouse</i> .

**Tabel 4.14** Log dari Proses ETL fact\_tatib

Stepname	Copy nr	Read	Written	In put	Out put	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
in_pelanggaran	0	0	2041	2041	0	0	0	0	Finished	0.1s	34,593	-
lihat_periode	0	2045	2041	0	0	0	0	0	Finished	0.2s	9,976	-
in_periode	0	0	4	4	0	0	0	0	Finished	0.0s	444	-
Group by	0	2041	1965	0	0	0	0	0	Finished	0.2s	9,537	-
out_pelanggaran	0	1965	1965	0	1965	0	0	0	Finished	0.5s	3,750	-
lihat_kelas	0	2070	2041	0	0	0	0	0	Finished	0.2s	9,904	-
in_kelas	0	0	29	29	0	0	0	0	Finished	0.0s	1,611	-

Pada tabel 4.14 ditunjukkan log dari proses ETL fact\_tatib, pada *stepname* in\_pelanggaran terdapat keterangan 0 pada *copynr*, artinya jumlah langkah (*step*) yang tersalin sebanyak 0, pada *Written* terdapat nilai 2041, artinya pada *database* OLTP yang menjadi inputan sistem berhasil tertulis sebanyak 2041 baris, pada keterangan *Input* terdapat nilai 2041, artinya jumlah baris yang terbaca dari *database* OLTP sebanyak 2041 baris. Untuk menuliskan data ke *database* pada *stepname* out\_pelanggaran proses ETL ini membutuhkan waktu 0,5 detik dengan kecepatan (*speed*) 3.750 rows/second, jumlah baris yang terbaca dari input sistem (*read*) sebanyak 1965 baris, jumlah baris yang tertulis di keluaran sistem (*written*) sebanyak 1965 baris dan jumlah baris yang tertulis di *database* (*output*) pada tabel fact\_tatib sebanyak 1965 baris.

Hasil proses ETL fact\_tatib yang dilakukan pada *Kettle* ditunjukkan pada gambar 4.20, sedangkan gambar 4.21 menunjukkan proses ETL fact\_tatib pada *Kettle* yang telah dimasukkan kedalam tabel fact\_tatib pada MySQL.

Rows of step: out\_pelanggaran (2040 rows)

#	nis	KD_KELAS	kd_tatib	id_periode	jml_poin
1	9968	0100	28	2	8
2	10019	0100	32	2	15
3	10035	0100	29	2	5
4	10121	0100	29	2	5
5	10125	0100	28	2	8
6	10142	0100	27	2	10
7	10146	0100	28	2	8

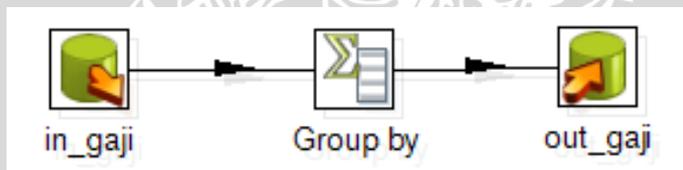
**Gambar 4.20** Hasil Proses ETL fact\_tatib pada *Kettle*

+ Options		nis	kd_tatib	id_periode	jml_poin	kd_kelas
<input type="checkbox"/>		9968	28	2	8	0100
<input type="checkbox"/>		10019	32	2	15	0100
<input type="checkbox"/>		10035	29	2	5	0100
<input type="checkbox"/>		10121	29	2	5	0100
<input type="checkbox"/>		10125	28	2	8	0100
<input type="checkbox"/>		10142	27	2	10	0100
<input type="checkbox"/>		10146	28	2	8	0100

Gambar 4.21 Hasil Proses ETL fact\_tatib pada MySQL

#### 4.2.1.8 Proses ETL fact\_penghasilan

Data penghasilan orang tua yang bersumber pada *database* MySQL akan di transformasi melalui proses ETL. Data penghasilan orang tua tersebut akan dimasukkan pada tabel fact\_penghasilan. Skema ETL fact\_penghasilan ditunjukkan pada gambar 4.22.



Gambar 4.22 Skema ETL fact\_penghasilan

Untuk penjelasan dari setiap simbol pada skema ETL fact\_penghasilan ditunjukkan pada tabel 4.15 dan untuk Log dari proses ETL fact\_nilai ditunjukkan oleh tabel 4.16.

Tabel 4.15 Keterangan Skema ETL fact\_penghasilan

Simbol	Keterangan
in_gaji	Berfungsi untuk mengambil data penghasilan orang tua dari <i>database</i> OLTP.
Group by	Berfungsi untuk mengelompokkan data penghasilan berdasarkan nis, dan nama_siswa.

 out_gaji	Berfungsi untuk memasukkan data hasil transformasi ke dalam tabel fact_penghasilan pada <i>datawarehouse</i> .
---	--

**Tabel 4.16** Log dari Proses ETL fact\_tatib

Stepname	Copy nr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
in_gaji	0	0	1463	1463	0	0	0	0	Finished	0.0s	45,719	-
Group by	0	1463	1463	0	0	0	0	0	Finished	0.0s	44,333	-
out_gaji	0	1463	1463	0	1463	0	0	0	Finished	0.3s	5,670	-

Pada tabel 4.16 ditunjukkan log dari proses ETL fact\_penghasilan, pada *stepname* in\_gaji terdapat keterangan 0 pada *copynr*, artinya jumlah langkah (*step*) yang tersalin sebanyak 0, pada *Written* terdapat nilai 1463, artinya pada *database* OLTP yang menjadi inputan sistem berhasil tertulis sebanyak 1463 baris, pada keterangan *Input* terdapat nilai 1463, artinya jumlah baris yang terbaca dari *database* OLTP sebanyak 1463 baris. Untuk menuliskan data ke *database* pada *stepname* out\_gaji proses ETL ini membutuhkan waktu 0,3 detik dengan kecepatan (*speed*) 5.670 rows/second, jumlah baris yang terbaca dari input sistem (*read*) sebanyak 1463 baris, jumlah baris yang tertulis di keluaran sistem (*written*) sebanyak 1463 baris dan jumlah baris yang tertulis di *database* (*output*) pada tabel fact\_penghasilan sebanyak 1463 baris.

Hasil proses ETL fact\_penghasilan yang dilakukan pada *Kettle* ditunjukkan pada gambar 4.23, sedangkan gambar 4.24 menunjukkan proses ETL fact\_penghasilan pada *Kettle* yang telah dimasukkan kedalam tabel fact\_penghasilan pada MySQL.

Rows of step: out\_gaji (1463 rows)

#	NIS	Nama_Siswa	Penghasilan_Ortu
1	10000	Arizal Ismoyo Wijanarko	6526000
2	10001	Aldwin Ryan Zakaria	5762000
3	10002	Annissa Kusumawardhani	7917000
4	10003	Annisa Rahmadiyah	5449000
5	10004	Dana Helmi Anggara	1147000

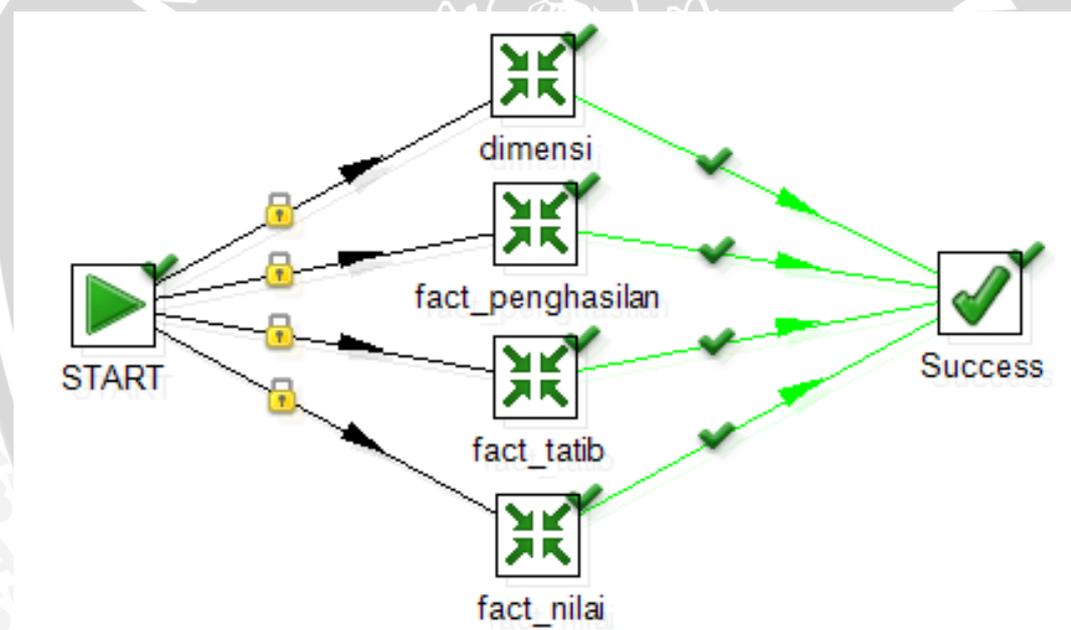
**Gambar 4.23** Hasil Proses ETL fact\_penghasilan pada *Kettle*

+ Options			NIS	Nama_Siswa	Penghasilan_Ortu
<input type="checkbox"/>			10000	Arizal Ismoyo Wijanarko	6526000
<input type="checkbox"/>			10001	Aldwin Ryan Zakaria	5762000
<input type="checkbox"/>			10002	Annissa Kusumawardhani	7917000
<input type="checkbox"/>			10003	Annisa Rahmadiah	5449000
<input type="checkbox"/>			10004	Dana Helmi Anocara	1147000

**Gambar 4.24** Hasil Proses ETL fact\_penghasilan pada MySQL

#### 4.2.1.9 Proses ETL job\_beasiswa

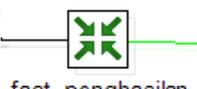
Proses ini merupakan proses menggabungkan semua transformasi menjadi satu job yang akan dijalankan pada Pentaho Data Integration (*Kettle*). Skema dari job\_ETL ditunjukkan oleh gambar 4.25.



**Gambar 4.25** Skema ETL fact\_penghasilan

Untuk penjelasan dari setiap simbol pada skema ETL job\_beasiswa ditunjukkan pada tabel 4.17 .

**Tabel 4.17** Keterangan Skema ETL job\_basiswa

Simbol	Keterangan
	Berfungsi untuk memulai beberapa file transformasi.
	Berfungsi untuk menjalankan transformasi untuk semua dimensi pada <i>datawarehouse</i> .
	Berfungsi untuk menjalankan transformasi untuk fact_penghasilan pada <i>datawarehouse</i> .
	Berfungsi untuk menjalankan transformasi untuk fact_tatib pada <i>datawarehouse</i> .
	Berfungsi untuk menjalankan transformasi untuk fact_nilai pada <i>datawarehouse</i> .
	Berfungsi untuk menjalankan untuk mengakhiri transformasi pada <i>datawarehouse</i> .

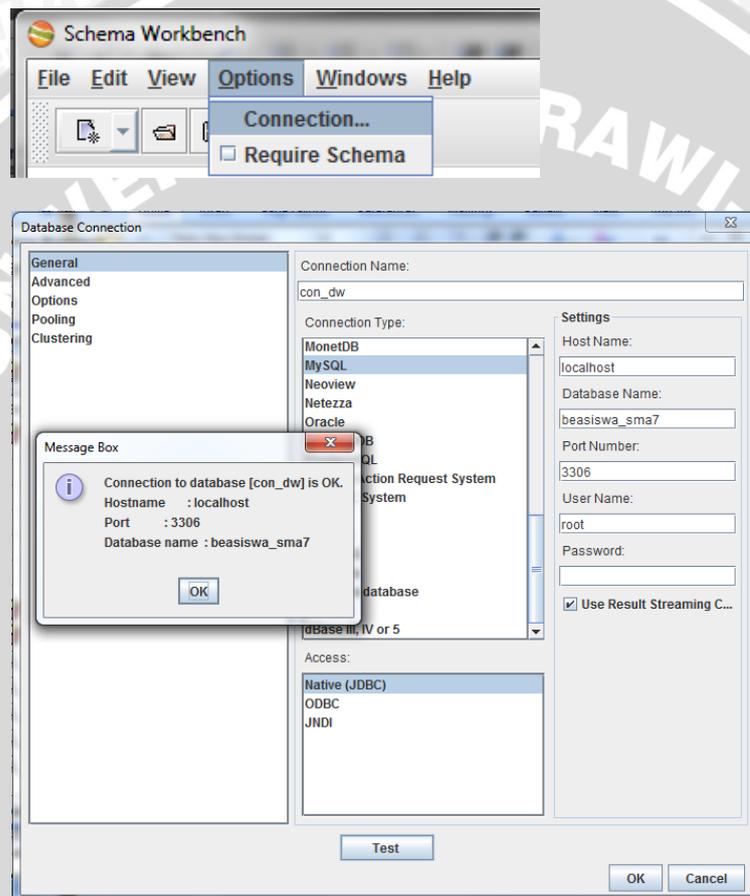
Pada gambar 4.26 menunjukkan semua proses ETL job\_basiswa yang dilakukan oleh *Kettle* berjalan dengan sukses.

Job / Job Entry	Comment	Result	Reason	Filename	Nr
job_basiswa					
Job: job_basiswa	Start of job execution		start		
START	Start of job execution		start		
START	Job execution finished	Success			0
dimensi	Start of job execution		Followed unconditional link	D:\KULIAH\_SKRIPSI\job_basiswa...	
dimensi	Job execution finished	Success		D:\KULIAH\_SKRIPSI\job_basiswa...	1
Success	Start of job execution		Followed link after success		
Success	Job execution finished	Success			1
fact_penghasilan	Start of job execution		Followed unconditional link	D:\KULIAH\_SKRIPSI\job_basiswa...	
fact_penghasilan	Job execution finished	Success		D:\KULIAH\_SKRIPSI\job_basiswa...	1
Success	Start of job execution		Followed link after success		
Success	Job execution finished	Success			1
fact_tatib	Start of job execution		Followed unconditional link	D:\KULIAH\_SKRIPSI\job_basiswa...	
fact_tatib	Job execution finished	Success		D:\KULIAH\_SKRIPSI\job_basiswa...	1
Success	Start of job execution		Followed link after success		
Success	Job execution finished	Success			1
fact_nilai	Start of job execution		Followed unconditional link	D:\KULIAH\_SKRIPSI\job_basiswa...	
fact_nilai	Job execution finished	Success		D:\KULIAH\_SKRIPSI\job_basiswa...	1
Success	Start of job execution		Followed link after success		
Success	Job execution finished	Success			1
Job: job_basiswa	Job execution finished	Success	finished		1

**Gambar 4.26** Hasil Proses ETL job\_basiswa pada *Kettle*

#### 4.2.2 Implementasi Struktur Kubus Data (*Cube*) OLAP

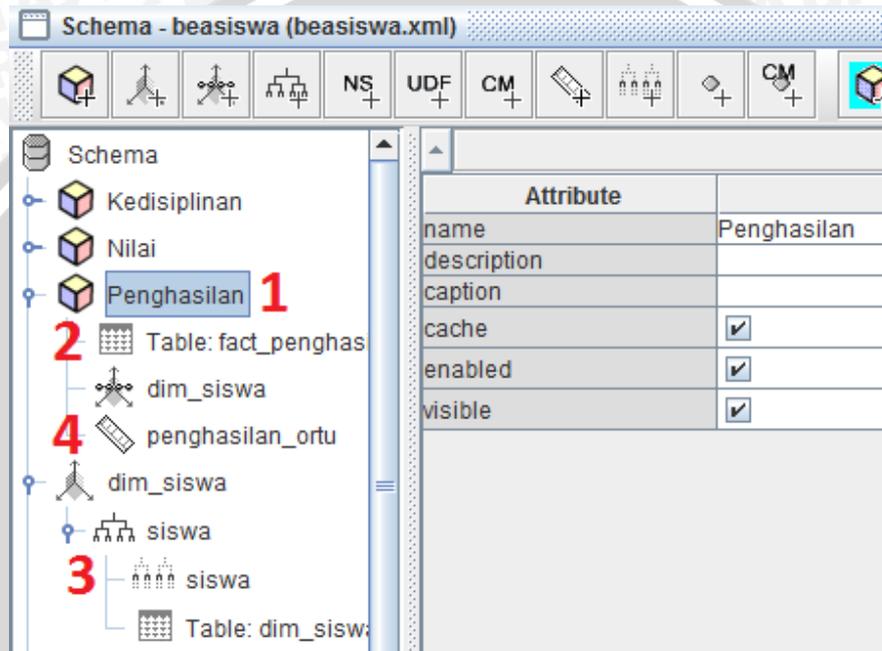
Implementasi struktur kubus data (*cube*) OLAP dalam penelitian ini menggunakan 3 kubus data yaitu kubus data penghasilan, kubus data nilai dan kubus data kedisiplinan. Untuk menghubungkan skema *datawarehouse* yang terdapat pada *database* MySQL dengan OLAP *Server* (Mondrian) dibutuhkan beberapa konfigurasi yang ditunjukkan pada gambar 4.27.



**Gambar 4.27** Konfigurasi koneksi *database* MySQL pada Mondrian.

Pada gambar 4.24 ditunjukkan konfigurasi untuk menghubungkan *database* MySQL dengan OLAP *server* (Mondrian). Langkah pertama adalah memilih menu *Options*, kemudian *Connection*. Pada *form database connection* pilih MySQL pada *connection type*, lalu isikan nama *database datawarehouse* pada *database name*, serta *user name* dan *password* untuk *database* kita. Pada *message box* ditunjukkan status OK yang berarti koneksi yang dibuat sudah berhasil.

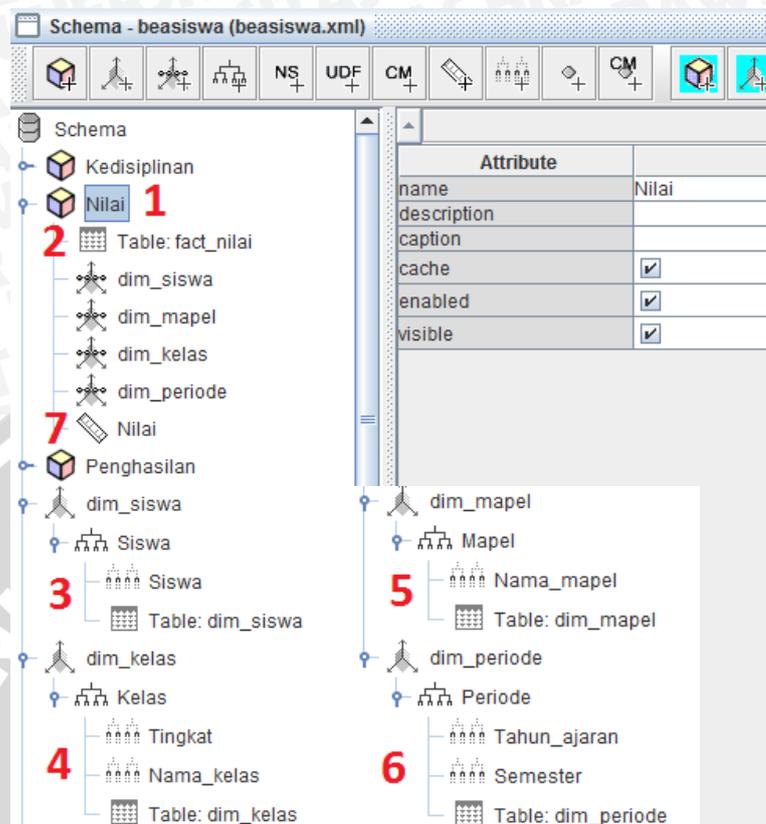
Setelah koneksi dengan *database* MySQL berhasil dilakukan, selanjutnya adalah membuat hirarki data yang berfungsi untuk melakukan proses OLAP yaitu *drill down*, dan *roll up*. Pada gambar 4.28 menunjukkan struktur dari kubus data Penghasilan dan pada gambar 4.29 menunjukkan struktur dari kubus data Nilai, sedangkan untuk struktur dari kubus data Kedisiplinan ditunjukkan oleh gambar 4.30.



**Gambar 4.28** Struktur Kubus Data Penghasilan

Berikut ini penjelasan untuk struktur kubus data Penghasilan :

1. Nama dari kubus data adalah Penghasilan.
2. Tabel yang digunakan adalah tabel *fact\_penghasilan*.
3. Menentukan suatu hirarki data. Dari sudut pandang dimensi siswa, dibuat satu hirarki data yaitu siswa. Sehingga data dari dimensi siswa dapat di *drill down* ke masing-masing siswa.
4. Pada kubus data Penghasilan ini ditentukan *measure* (ukuran yang akan digunakan dalam proses agregasi data) yaitu “*penghasilan\_ortu*”. Dengan ini akan dapat dilihat *penghasilan\_ortu* sesuai dengan dimensi yang ditentukan.



**Gambar 4.29** Struktur Kubus Data Nilai

Berikut ini penjelasan untuk struktur kubus data Nilai :

1. Nama dari kubus data adalah Nilai.
2. Tabel yang digunakan adalah tabel fact\_nilai
3. Menentukan suatu hirarki data. Dari sudut pandang dimensi siswa, dibuat satu hirarki data yaitu siswa. Sehingga data dari dimensi siswa dapat di *drill down* ke masing-masing siswa.
4. Dari sudut pandang dimensi kelas, dibuat dua hirarki data yaitu Tingkat dan Nama\_kelas. Hirarki paling tinggi adalah Tingkat, sehingga data dari dimensi kelas dapat di *drill down* ke Tingkat, kemudian juga dapat di *drill down* lagi ke Nama\_kelas.
5. Dari sudut pandang dimensi mata pelajaran, dibuat satu hirarki data yaitu Nama\_mapel. Sehingga data dari dimensi mata pelajaran dapat di *drill down* ke masing-masing nama mata pelajaran.
6. Dari sudut pandang dimensi periode, dibuat dua hirarki data yaitu Tahun\_ajaran dan Semester. Hirarki paling tinggi adalah Tahun Ajaran,



kelas dapat di *drill down* ke Tingkat, kemudian juga dapat di *drill down* lagi ke Nama\_kelas.

5. Dari sudut pandang dimensi tata tertib, dibuat satu hirarki data yaitu Nama\_tatib. Sehingga data dari dimensi tata tertib dapat di *drill down* ke masing-masing nama tata tertib.
6. Dari sudut pandang dimensi periode, dibuat dua hirarki data yaitu Tahun\_ajaran dan Semester. Hirarki paling tinggi adalah Tahun Ajaran, sehingga data dari dimensi periode dapat di *drill down* ke Tahun\_ajaran, kemudian juga dapat di *drill down* lagi ke Semester.
7. Pada kubus data Kedisiplinan ini ditentukan *measure* (ukuran yang akan digunakan dalam proses agregasi data) yaitu “total\_poin”. Sehingga akan dapat dilihat total\_poin sesuai dengan dimensi yang ditentukan.

### 4.2.3 Implementasi Program Rekomendasi Beasiswa

#### 4.2.3.1 Menentukan Batasan Parameter

Proses ini digunakan untuk menentukan batasan dari setiap parameter yang digunakan dalam proses rekomendasi, dan setiap parameter dibagi menjadi tiga himpunan. Kemudian batasan parameter tersebut akan disimpan ke dalam *array* untuk diproses ke fungsi selanjutnya. Proses menentukan batasan parameter ditunjukkan oleh tabel 4.18.

**Tabel 4.18** Source Code Menentukan Batasan Parameter

Baris	parameter.php -> function main_process()
1	<code>\$threshold = array();</code>
2	
3	<code>\$bound = array('lower' =&gt; 0, 'middle' =&gt; 60, 'upper' =&gt;</code>
4	<code>70);</code>
5	<code>\$threshold[1] = \$bound;</code>
6	
7	<code>\$bound = array('lower' =&gt; 60, 'middle' =&gt; 70, 'upper' =&gt;</code>
8	<code>85);</code>
9	<code>\$threshold[2] = \$bound;</code>
10	
11	<code>\$bound = array('lower' =&gt; 80, 'upper' =&gt; 100);</code>
12	<code>\$threshold[3] = \$bound;</code>
13	
14	<code>\$bound = array('lower' =&gt; 0, 'middle' =&gt; 2000000, 'upper'</code>
15	<code>=&gt; 4000000);</code>
16	<code>\$threshold[4] = \$bound;</code>

```

17
18 $bound = array('lower' => 2000000, 'middle' => 4000000,
19 'upper' => 6000000);
20 $threshold[5] = $bound;
21
22 $bound = array('lower' => 5000000, 'upper' => 8000000);
23 $threshold[6] = $bound;
24
25 $bound = array('lower' => 0, 'middle' => 20, 'upper' =>
26 50);
27 $threshold[7] = $bound;
28
29 $bound = array('lower' => 20, 'middle' => 50, 'upper' =>
30 80);
31 $threshold[8] = $bound;
32
33 $bound = array('lower' => 60, 'upper' => 120);
34 $threshold[9] = $bound;
35
36 $this->session->set_userdata('threshold', $threshold);
37 $this->load->view('templatel', $data);

```

#### 4.2.3.2 Mengambil Nilai pada Semua Parameter untuk Setiap Siswa

Proses ini digunakan untuk mengambil nilai dari semua parameter untuk setiap siswa pada kelas dan tahun ajaran tertentu yang telah dipilih dalam proses rekomendasi. Setelah didapatkan rata-rata nilai, penghasilan orang tua, dan poin pelanggaran untuk setiap siswa pada tahun ajaran dan kelas tertentu, selanjutnya akan diproses dalam fungsi `hitung_derajat`, `hitung_firestrength`, dan `hitung_selected` seperti yang ditunjukkan pada baris ke-34 hingga baris ke-37 pada tabel 4.19

**Tabel 4.19** *Source Code* Mengambil Nilai Setiap Parameter dari Semua Siswa

Baris	parameter.php -> function query_process(\$parameter_kelas = 0, \$parameter_tahun = "")
1	<code>\$query_nilai = \$this-&gt;Fuzzy_model-&gt;get_nilai_candidate();</code>
2	<code>\$list_nilai = array();</code>
3	<code>foreach(\$query_nilai-&gt;result() as \$row)</code>
4	<code>{</code>
5	<code>    \$list_nilai[\$row-&gt;nis] = \$row-&gt;rerata;</code>
6	<code>}</code>
7	
8	<code>\$query_penghasilan = \$this-&gt;Fuzzy_model-&gt;</code>
9	<code>get_penghasilan_candidate();</code>
10	<code>\$list_penghasilan = array();</code>
11	<code>foreach(\$query_penghasilan-&gt;result() as \$row)</code>
12	<code>{</code>

```

13     $list_penghasilan[$row->NIS] = $row-
14 >Penghasilan_Ortu;
15     }
16
17     $list_nis = $this->Fuzzy_model->get_nis_candidate();
18     $list_disiplin = array();
19     foreach($list_nis->result() as $row)
20     {
21         $nis = $row->nis;
22         $query_disiplin = $this->Fuzzy_model->
23             get_disiplin_candidate($nis);
24         if($query_disiplin->num_rows() > 0)
25         {
26             $list_disiplin[$nis] = $query_disiplin->row()->
27                 total;
28         }
29         else
30         {
31             $list_disiplin[$nis] = 0;
32         }
33     }
34
35     $derajat = $this->hitung_derajat($list_nis, $list_nilai,
36     $list_penghasilan, $list_disiplin);
37     $this->hitung_firestrength($list_nis, $derajat);
38     $this->hitung_selected();

```

Pada tabel 4.19, untuk mengambil nilai pada semua parameter diperlukan adanya suatu *query* yang diambil dari `fuzzy_model.php` yang ditunjukkan oleh tabel 4.20. Pada baris ke-1, ke-9, dan ke-22 terdapat sebuah perintah yang memanggil *query* pada `fuzzy_model.php`.

**Tabel 4.20** *Source Code Query Mengambil Nilai Setiap Parameter*

Baris	fuzzy_model.php
1	<code>function get_nilai_candidate()</code>
2	<code>{</code>
3	<code>    \$sql = "select fact_nilai.nis, sum(fact_nilai.Nilai) as</code>
4	<code>        total, avg(fact_nilai.Nilai) as rerata</code>
5	<code>from fact_nilai where fact_nilai.id_periode in</code>
6	<code>    (select * from view_periode) and fact_nilai.KD_KELAS</code>
7	<code>in</code>
8	<code>    (select * from view_tingkat) group by</code>
9	<code>    fact_nilai.nis;";</code>
10	<code>return \$this-&gt;db-&gt;query(\$sql);</code>
11	<code>}</code>
12	
13	<code>function get_penghasilan_candidate()</code>
14	<code>{</code>
15	<code>    \$sql = "select fact_penghasilan.NIS,</code>
16	<code>        fact_penghasilan.Penghasilan_Ortu</code>
17	<code>from fact_penghasilan where fact_penghasilan.nis in</code>

```

18         (select * from view_nis);" ;
19     return $this->db->query($sql);
20 }
21
22 function get_disiplin_candidate($nis)
23 {
24     $sql = "select fact_tatib.nis, sum(fact_tatib.jml_poin)
25           as total
26           from fact_tatib where fact_tatib.nis = $nis group by
27           fact_tatib.nis);" ;
28     return $this->db->query($sql);
29 }

```

### 4.2.3.3 Menentukan Derajat Keanggotaan

Proses ini digunakan untuk menentukan derajat keanggotaan setiap parameter yang akan berfungsi untuk menghitung nilai *firestrength* dalam proses rekomendasi. Untuk inisialisasi derajat keanggotaan pada setiap himpunan yang dimiliki masing-masing parameter ditunjukkan oleh tabel 4.21.

**Tabel 4.21** Source Code Inisialisasi Derajat Keanggotaan

Baris	parameter.php -> function hitung_derajat(\$list_nis, \$list_nilai, \$list_penghasilan, \$list_disiplin)
1	\$derajat = array();
2	\$threshold = \$this->session->userdata('threshold');
3	foreach(\$list_nis->result() as \$row)
4	{
5	\$nis = \$row->nis;
6	\$level = array();
7	\$level[1] = \$this->low(\$list_nilai[\$nis],
8	\$threshold[1]);
9	\$level[2] = \$this->med(\$list_nilai[\$nis],
10	\$threshold[2]);
11	\$level[3] = \$this->high(\$list_nilai[\$nis],
12	\$threshold[3]);
13	\$level[4] = \$this->low(\$list_penghasilan[\$nis],
14	\$threshold[4]);
15	\$level[5] = \$this->med(\$list_penghasilan[\$nis],
16	\$threshold[5]);
17	\$level[6] = \$this->high(\$list_penghasilan[\$nis],
18	\$threshold[6]);
19	\$level[7] = \$this->low(\$list_disiplin[\$nis],
20	\$threshold[7]);
21	\$level[8] = \$this->med(\$list_disiplin[\$nis],
22	\$threshold[8]);
23	\$level[9] = \$this->high(\$list_disiplin[\$nis],
24	\$threshold[9]);
25	\$derajat[\$nis] = \$level;
	}
	return \$derajat;

Sedangkan untuk aturan atau rumus derajat keanggotaan pada setiap parameter ditunjukkan oleh tabel 4.22. Setiap parameter dibagi menjadi tiga himpunan yaitu kurang, cukup, dan baik. Kemudian setiap himpunan ini ditentukan rumus derajat keanggotaannya berdasarkan batasan yang telah ditentukan pada proses awal.

**Tabel 4.22** *Source Code* Menentukan Rumus Derajat Keanggotaan

Baris	
1	function low(\$nilai, \$bound)
2	{
3	\$fuzzy = 0;
4	\$lower = \$bound['lower'];
5	\$middle = \$bound['middle'];
6	\$upper = \$bound['upper'];
7	
8	if(\$nilai <= \$middle)
9	{
10	\$fuzzy = 1;
11	}
12	else if(\$nilai <= \$upper && \$nilai > \$middle)
13	{
14	\$fuzzy = (\$upper - \$nilai) / (\$upper - \$middle);
15	}
16	else if(\$nilai > \$upper)
17	{
18	\$fuzzy = 0;
19	}
20	return \$fuzzy;
21	}
22	
23	function med(\$nilai, \$bound)
24	{
25	\$fuzzy = 0;
26	\$lower = \$bound['lower'];
27	\$middle = \$bound['middle'];
28	\$upper = \$bound['upper'];
29	if(\$nilai <= \$upper && \$nilai > \$middle)
30	{
31	\$fuzzy = (\$upper - \$nilai) / (\$upper - \$middle);
32	}
33	else if(\$nilai <= \$middle && \$nilai > \$lower)
34	{
35	\$fuzzy = (\$nilai - \$lower) / (\$middle - \$lower);
36	}
37	else if(\$nilai <= \$lower    \$nilai > \$upper)
38	{
39	\$fuzzy = 0;
40	}
41	return \$fuzzy;
42	}
43	

```

44 function high($nilai, $bound)
45 {
46     $fuzzy = 0;
47     $lower = $bound['lower'];
48     $upper = $bound['upper'];
49     if($nilai > $upper)
50     {
51         $fuzzy = 1;
52     }
53     else if($nilai <= $upper && $nilai > $lower)
54     {
55         $fuzzy = ($nilai - $lower) / ($upper - $lower);
56     }
57     else if($nilai <= $lower)
58     {
59         $fuzzy = 0;
60     }
61     return $fuzzy;
62 }

```

#### 4.2.3.4 Menghitung Firestrength

Proses ini digunakan untuk menghitung *firestrength* untuk setiap siswa pada kelas dan tahun ajaran yang telah ditentukan, besarnya nilai *firestrength* pada setiap siswa akan mempengaruhi besarnya nilai rekomendasi setiap siswa. Proses untuk menghitung *firestrength* ditunjukkan oleh tabel 4.23.

**Tabel 4.23** Source Code Menghitung Firestrength

Baris	parameter.php -> function hitung_firestrength(\$list_nis, \$derajat)
1	\$parameter_nilai = \$this->session-> userdata
2	('parameter_nilai');
3	\$parameter_gaji = \$this->session-> userdata
4	('parameter_gaji');
5	\$parameter_disiplin = \$this->session-> userdata
6	('parameter_disiplin');
7	
8	foreach(\$list_nis->result() as \$row)
9	{
10	\$nis = \$row->nis;
11	\$parameter = array();
12	\$level = \$derajat[\$nis];
13	\$n = 0;
14	if(\$parameter_nilai != 0)
15	{
16	\$parameter[\$n] = \$level[\$parameter_nilai];
17	\$n++;
18	}
19	if(\$parameter_gaji != 0)
20	{

```

21     $parameter[$n] = $level[$parameter_gaji];
22     $n++;
23 }
24 if($parameter_disiplin != 0)
25 {
26     $parameter[$n] = $level[$parameter_disiplin];
27     $n++;
28 }
29
30 $min = 1;
31 for($i = 0; $i < $n; $i++)
32 {
33     if($parameter[$i] < $min)
34     {
35         $min = $parameter[$i];
36     }
37 }
38 $firestrength = array('nis' => $nis, 'firestrength' =>
39 $min);
40 $this->Fuzzy_model->add_firestrength($firestrength);
41 $data_derajat = array('nis' => $nis, 'nilai_kurang' =>
42     $level[1], 'nilai_cukup' => $level[2],
43     'nilai_baik' => $level[3], 'gaji_kurang' =>
44     $level[4], 'gaji_cukup' => $level[5],
45     'gaji_kaya' => $level[6],
46     'disiplin_kurang'=> $level[9],
47     'disiplin_cukup'=> $level[8],
48     'disiplin_baik'=> $level[7]);
    $this->Fuzzy_model->add_derajat($data_derajat);
}

```

Nilai dari setiap parameter setiap siswa yang telah diambil kemudian digunakan dalam proses menghitung *firestrength*. Proses ini dilakukan berdasarkan parameter-parameter apa saja yang telah dipilih dalam sistem. Selanjutnya dilakukan proses perbandingan dari setiap parameter yang terpilih dan diambil nilai terkecil seperti yang ditunjukkan pada baris ke-27 hingga baris ke-33. Nilai terkecil itulah yang kemudian menjadi nilai *firestrength* dari setiap siswa. Dan nilai *firestrength* semua siswa disimpan dalam tabel *firestrength*. Selanjutnya semua derajat keanggotaan setiap parameter untuk semua siswa akan disimpan dalam tabel *derajat*.

#### 4.2.3.5 Mendapatkan Calon Penerima Beasiswa

Proses ini digunakan untuk mendapatkan calon siswa yang berhak menerima beasiswa berdasarkan kelas dan tahun ajaran tertentu serta jumlah siswa yang telah ditentukan pada sistem. Nilai *firestrength* setiap siswa akan disimpan

dalam tabel\_firestrength, kemudian tabel tersebut diurutkan berdasarkan *firestrength* terbesar dan kemudian diambil *n* data teratas untuk mendapatkan jumlah calon penerima beasiswa. Proses tersebut ditunjukkan oleh tabel 4.24.

**Tabel 4.24** *Source Code* Mendapatkan Calon Penerima Beasiswa

Baris	parameter.php -> function hitung_selected()
1	<code>\$kuota = \$this-&gt;session-&gt;userdata('kuota');</code>
2	<code>\$parameter_gaji = \$this-&gt;session-&gt;</code>
3	<code>userdata('parameter_gaji');</code>
4	<code>\$sorted_firestrength = array();</code>
5	<code>\$n = 0;</code>
6	<code>\$query_firestrength = \$this-&gt;Fuzzy_model-&gt;get_uptozero();</code>
7	
8	<code>foreach(\$query_firestrength-&gt;result() as \$row)</code>
9	<code>{</code>
10	<code>    \$sorted_firestrength[\$n] = array('nis' =&gt; \$row-&gt;nis,</code>
11	<code>        'firestrength' =&gt; \$row-</code>
12	<code>        -&gt;firestrength);</code>
13	<code>    \$n++;</code>
14	<code>}</code>
15	
16	<code>for(\$i = 0; \$i &lt; \$kuota; \$i++)</code>
17	<code>{</code>
18	<code>    \$selected = array('nis' =&gt;</code>
19	<code>        \$sorted_firestrength[\$i]['nis'],</code>
20	<code>        'firestrength' =&gt;</code>
21	<code>        \$sorted_firestrength[\$i]['firestrength']);</code>
22	<code>    \$this-&gt;Fuzzy_model-&gt;add_selected(\$selected);</code>
	<code>}</code>

#### 4.2.3.6 Menghitung Presentase Perbandingan Hasil

Proses ini digunakan untuk menghitung Presentase perbandingan hasil rekomendasi beasiswa oleh sistem yang telah dibandingkan dengan data aktual penerima beasiswa SMA Negeri 7 Malang. Proses perhitungan ditunjukkan oleh tabel 4.25.

**Tabel 4.25** *Source Code* Menghitung Presentase Perbandingan Hasil

Baris	proses_akurasi.php -> function hitung_akurasi()
1	<code>\$match = 0;\$jumlah = \$this-&gt;session-&gt;userdata('kuota');</code>
2	<code>foreach(\$dataexcel as \$row)</code>
3	<code>{</code>
4	<code>    \$nis = \$row['nis'];</code>
5	<code>    \$query_siswa = \$this-&gt;Fuzzy_model-</code>
6	<code>    -&gt;get_siswa bynim(\$nis);</code>

```

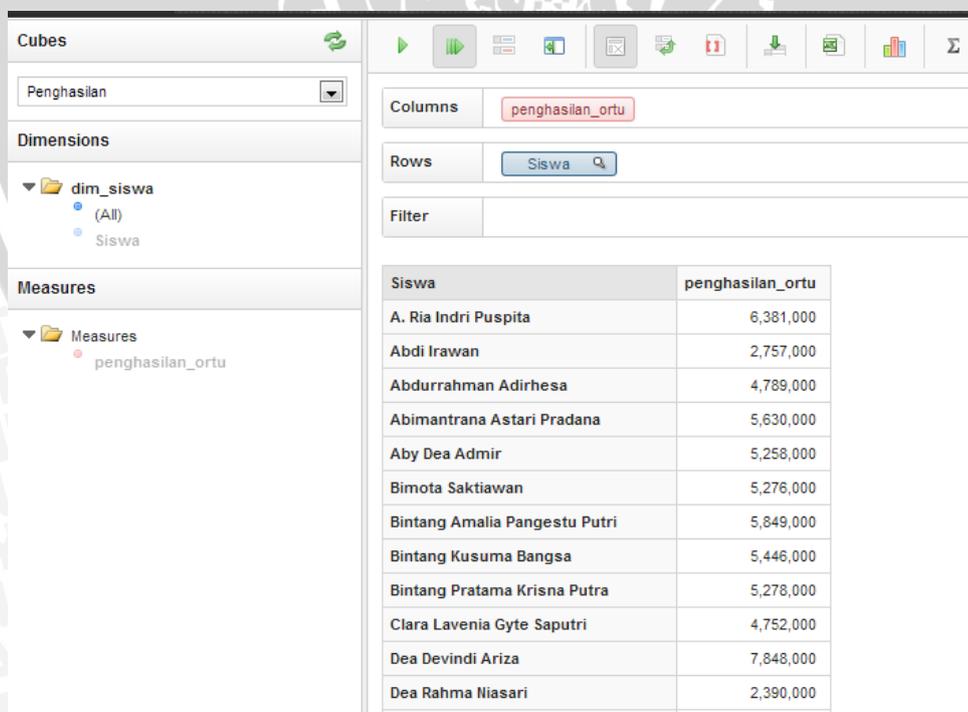
7     if($query_siswa->num_rows() > 0)
8     {
9         $match++;
10    }
11    }
12    $akurasi = ($match / $jumlah) * 100;
13    $data['jml_cocok'] = "Terdapat sebanyak ".$match." data
14    yang sama dari ".$skuota." data yang
15    dibandingkan";
16    $data['tampil_akurasi'] = "SEHINGGA => Presentase Kecocokan
17    sistem dengan data Aktual dari kelas ".$skelas." dan tahun
18    ajaran ".$stahun_ajaran." adalah sebesar ".round($akurasi,
19    2, PHP_ROUND_HALF_EVEN).' %';

```

### 4.3 Implementasi Antarmuka

#### 4.3.1 Implementasi Antarmuka Rekap Laporan

Berdasarkan perancangan antar muka pada bab 3 maka dihasilkan *form* rekap laporan Penghasilan Orang Tua, *form* rekap laporan Nilai, dan *form* rekap laporan Kedisiplinan. *Form* laporan untuk penghasilan orang tua ditunjukkan oleh gambar 4.31.



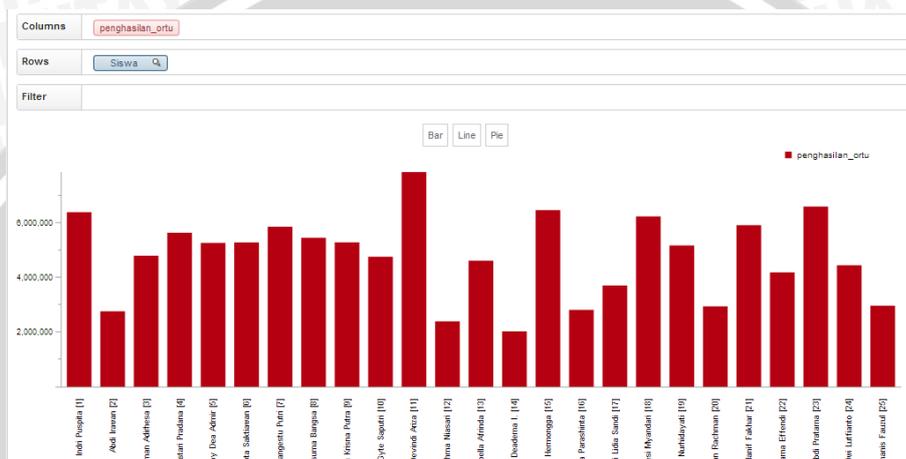
The screenshot shows a data visualization interface with the following components:

- Cubes:** A dropdown menu set to 'Penghasilan'.
- Dimensions:** A folder 'dim\_siswa' containing '(All)' and 'Siswa'.
- Measures:** A folder 'Measures' containing 'penghasilan\_ortu'.
- Columns:** A dropdown menu set to 'penghasilan\_ortu'.
- Rows:** A dropdown menu set to 'Siswa'.
- Filter:** An empty filter box.
- Table:** A table with two columns: 'Siswa' and 'penghasilan\_ortu'. The table contains 13 rows of student names and their corresponding values.

Siswa	penghasilan_ortu
A. Ria Indri Puspita	6,381,000
Abdi Irawan	2,757,000
Abdurrahman Adirhesa	4,789,000
Abimantrana Astari Pradana	5,630,000
Aby Dea Admir	5,258,000
Bimota Saktiawan	5,276,000
Bintang Amalia Pangestu Putri	5,849,000
Bintang Kusuma Bangsa	5,446,000
Bintang Pratama Krisna Putra	5,278,000
Clara Lavenia Gyte Saputri	4,752,000
Dea Devindi Ariza	7,848,000
Dea Rahma Niasari	2,390,000

Gambar 4.31 Rekap Laporan Penghasilan Orang Tua

Pada gambar 4.31 ditampilkan rekap laporan Penghasilan orang tua dilihat dari sudut pandang siswa. Selain itu rekap laporan juga dapat ditampilkan dengan menggunakan fasilitas *Filter* data pada aplikasi, hal ini digunakan untuk menampilkan laporan berdasarkan siswa yang dipilih, dan laporan yang telah dilakukan proses *Filter* data juga dapat disajikan dalam bentuk grafik batang seperti yang ditunjukkan pada gambar 4.32.

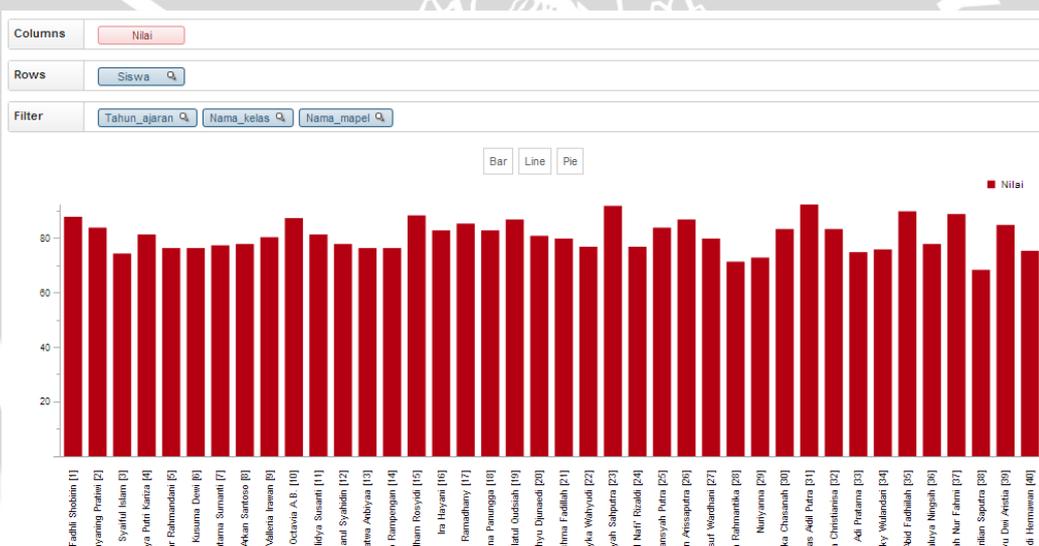


**Gambar 4.32** Grafik Batang Rekap Laporan Penghasilan Orang Tua

Begitu pula dengan rekap laporan nilai yang juga dapat dilihat dari berbagai sudut pandang dan juga dapat dilakukan proses *filter* data berdasarkan apa yang ingin disajikan. Pada gambar 4.33 menunjukkan hasil laporan nilai berdasarkan sudut pandang tahun ajaran, nama kelas, mata pelajaran, dan siswa. Kemudian dilakukan proses *filter* data yaitu pada tahun ajaran 2010/2011, pada kelas X-1, dan pada mata pelajaran Matematika. Sedangkan untuk laporan dalam bentuk grafik batang ditunjukkan oleh gambar 4.34.

Tahun_ajaran	Nama_kelas	Nama_mapel	Siswa	Nilai
2010/2011	X-1	Matematika	Achmad Fadhli Shobirin	88
			Andini Cahyaning Pratiwi	84
			Anfa Syaiful Islam	74.5
			Anindya Putri Kariza	81.5
			Anissa Yanuar Rahmandani	76.5
			Anita Kusuma Dewi	76.5
			Anthony Yoga Pratama Sumanti	77.5
			Atras Arkan Santoso	78
			Baby Rachel Valleria Irawan	80.5
			Chotimatus Kartika Octavia A.B.	87.5
			Cici Maulidva Susanti	81.5

Gambar 4.33 Rekap Laporan Nilai



Gambar 4.34 Grafik Batang Rekap Laporan Nilai

Untuk hasil laporan kedisiplinan (pelanggaran tata tertib) berdasarkan sudut pandang tahun ajaran, dan nama kelas ditunjukkan oleh gambar 4.35. Kemudian untuk menampilkan perbandingan jumlah poin pelanggaran setiap kelas pada tahun ajaran 2011/2012, maka dilakukan proses *filter* data yaitu hanya pada tahun ajaran 2011/2012, dan laporan dalam bentuk grafik batang ditunjukkan oleh gambar 4.36.

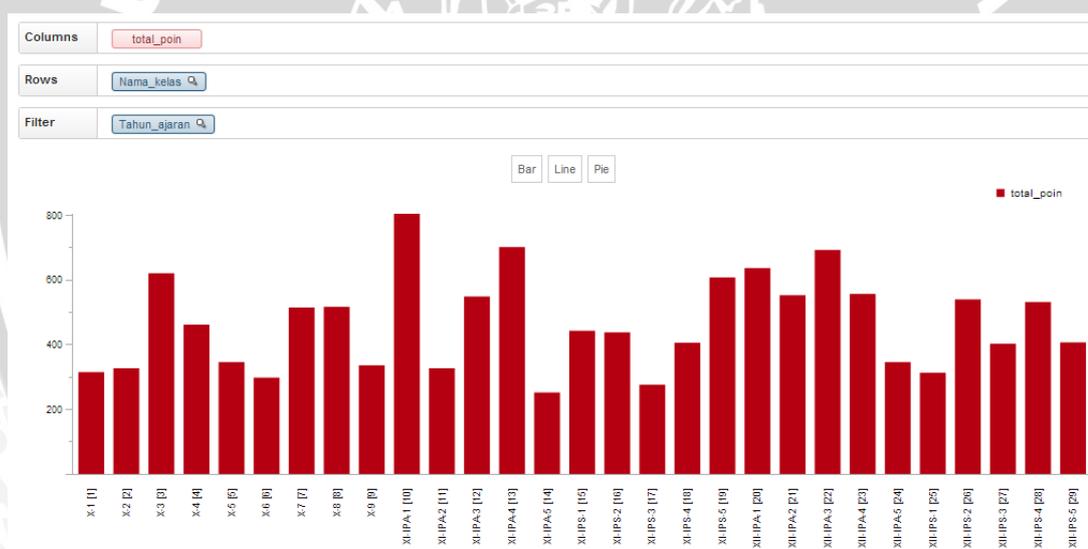


The screenshot shows a data cube interface with the following components:

- Dimensions:**
  - dim\_kelas: (All), Tingkat, Nama\_kelas
  - dim\_periode: (All), Tahun\_ajaran, Semester
  - dim\_siswa
  - dim\_tatib
- Measures:**
  - total\_poin
- Columns:** total\_poin
- Rows:** Tahun\_ajaran, Nama\_kelas
- Table Data:**

Tahun_ajaran	Nama_kelas	total_poin
2011/2012	X-1	315
	X-2	327
	X-3	621
	X-4	462
	X-5	346
	X-6	298
	X-7	515
	X-8	517
	X-9	336
XI-IPA-1	805	
XI-IPA-2	327	
XI-IPA-3	549	

Gambar 4.35 Rekap Laporan Kedisiplinan (Pelanggaran Tata Tertib)



Gambar 4.36 Grafik Batang Rekap Laporan Kedisiplinan (Pelanggaran Tata Tertib)

### 4.3.2 Implementasi Antarmuka Rekomendasi Beasiswa

#### 4.3.2.1 Halaman Login

Sistem pendukung keputusan rekomendasi beasiswa ini mempunyai dua jenis *user* yaitu *super\_admin* yang memiliki semua hak akses pada sistem, *user* jenis ini hanya digunakan oleh peneliti, karena terdapat sebuah halaman pengujian yang hanya digunakan dalam penelitian ini. Sedangkan jenis *user* kedua adalah *admin\_sekolah*, yaitu *user* yang memiliki semua hak akses kecuali halaman pengujian. Halaman Login ditunjukkan oleh gambar 4.37.



Gambar 4.37 Halaman Login

#### 4.3.2.2 Halaman Beranda

Halaman beranda merupakan halaman sambutan selamat datang dan sedikit deskripsi tentang sistem rekomendasi beasiswa. Halaman Beranda ditunjukkan oleh gambar 4.38.



**Gambar 4.38** Halaman Beranda

#### 4.3.2.3 Halaman Rekomendasi

Halaman ini merupakan halaman utama yang digunakan dalam proses rekomendasi beasiswa. Pengguna harus memilih parameter-parameter yang akan digunakan sebagai acuan dalam proses rekomendasi beasiswa. Selain itu, pengguna juga harus memilih kelas dan tahun ajaran, serta mengisi jumlah siswa calon penerima beasiswa. Untuk patokan normal parameter yang digunakan dalam proses rekomendasi adalah “Nilai Baik” untuk parameter nilai, “Kurang Mampu” untuk parameter penghasilan orang tua, dan “Disiplin” untuk parameter kedisiplinan. Halaman Rekomendasi ditunjukkan oleh gambar 4.39.

Selanjutnya sistem akan menampilkan derajat keanggotaan untuk semua siswa pada kelas dan tahun ajaran yang telah dipilih, seperti yang ditunjukkan oleh gambar 4.40. Sedangkan untuk hasil rekomendasi beasiswa dengan jumlah siswa akan ditampilkan kedalam bentuk tabel seperti yang ditunjukkan oleh gambar 4.41. Selain itu tabel hasil rekomendasi beasiswa juga dapat diunduh kedalam format “.xls” (*Microsoft Excel 1997-2003*).



Gambar 4.39 Halaman Rekomendasi

Derajat Keanggotaan Semua Siswa X Tahun Ajaran 2010/2011

1 2 3 > Last >

NO	NIS	NAMA	KELAS	NILAI KURANG	NILAI CUKUP	NILAI BAIK	PENGHASILAN KURANG	PENGHASILAN MAMPU	PENGHASILAN KAYA	DISIPLIN	CUKUP DISIPLIN	KURANG DISIPLIN
1	9875	A'isy Putri Akbar As'ad	X-7	0.000	0.089	0.183	0.000	0.000	0.571	0.167	0.333	0.000
2	9864	Aby Dea Admir	X-9	0.000	0.149	0.138	0.000	0.371	0.086	0.000	0.000	1.000
3	9968	Achmad Fadhli Shobirin	X-1	0.000	0.080	0.190	1.000	0.000	0.000	0.000	0.867	0.000
4	9555	Achmad Soleh Aditya	X-7	0.000	0.102	0.173	0.000	0.185	0.210	0.000	0.000	1.000
5	9879	Adam Ciputra	X-4	0.000	0.084	0.187	0.000	0.000	0.337	0.000	0.000	1.000
6	9877	Adam Syah Rizal	X-9	0.000	0.178	0.117	0.000	0.241	0.173	0.000	0.000	1.000
7	9873	Ade Fardhansyah	X-5	0.000	0.156	0.133	0.000	0.000	0.928	0.000	0.000	1.000
8	9860	Adelia Lailatul Fitriyah	X-7	0.000	0.160	0.130	0.000	0.000	0.905	0.000	0.000	1.000
9	9862	Adhik Rendy Putra Anggara	X-7	0.000	0.178	0.117	0.000	0.000	0.503	0.000	0.000	1.000
10	9911	Adilah Irzwanika	X-4	0.000	0.302	0.023	0.000	0.000	0.401	0.000	0.000	1.000
11	9896	Adinda Caesar P	X-9	0.000	0.244	0.067	0.000	0.023	0.318	0.000	0.000	1.000
12	9859	Adinda Rizki Zuhail	X-8	0.000	0.216	0.088	0.000	0.442	0.039	0.000	0.000	1.000
13	9892	Adisti Niken Sawitri Anugrahani	X-8	0.000	0.082	0.188	0.000	0.000	0.618	0.167	0.333	0.000
14	9854	Aditya Aris Pradana	X-6	0.000	0.173	0.120	0.000	0.000	0.459	0.000	0.000	1.000
15	9639	Aditya Kristianto	X-2	0.000	0.191	0.107	0.000	0.000	0.585	1.000	0.000	0.000
16	9675	Aditya Nugraha	X-3	0.000	0.118	0.162	1.000	0.000	0.000	0.000	0.267	0.733
17	9856	Adriawan Alif	X-3	0.000	0.073	0.185	0.000	0.000	0.378	0.000	0.667	0.333

Gambar 4.40 Derajat Keanggotaan Siswa

### Hasil Rekomendasi Beasiswa Kelas X Tahun Ajaran 2010/2011

1 2 3 >

NO	NIS	NAMA	KELAS	FIRESTRENGTH	NILAI REKOMENDASI	
1	10272	Mahaka Rizal Faizzuna A.F	X-4	0.270	2.79 %	<a href="#">▶ detail</a>
2	10128	Denti Lintang Lajuarida	X-3	0.253	2.61 %	<a href="#">▶ detail</a>
3	9913	Avryan Harvyandha	X-7	0.253	2.61 %	<a href="#">▶ detail</a>
4	10111	Galih Anggriawan	X-5	0.248	2.56 %	<a href="#">▶ detail</a>
5	9921	Anisa Permatasari	X-4	0.240	2.48 %	<a href="#">▶ detail</a>
6	10043	Dhimaz Ajeng Larasati	X-7	0.232	2.4 %	<a href="#">▶ detail</a>
7	10065	Dhea Revana Rossanti	X-8	0.230	2.38 %	<a href="#">▶ detail</a>
8	9919	Alvi Nailus Sa'adah	X-4	0.228	2.36 %	<a href="#">▶ detail</a>
9	10178	Ilham Rosyidi	X-1	0.223	2.3 %	<a href="#">▶ detail</a>
10	10209	Fernanda Tabetha Claudia Wongkar	X-3	0.222	2.29 %	<a href="#">▶ detail</a>
11	10107	Happy Tsani	X-2	0.218	2.25 %	<a href="#">▶ detail</a>
12	9976	Annisa Kusumawardani	X-6	0.215	2.22 %	<a href="#">▶ detail</a>
13	10228	Imam Ekopambudi	X-4	0.215	2.22 %	<a href="#">▶ detail</a>
14	10372	Hernada Firdaus Eka Saputri	X-9	0.208	2.15 %	<a href="#">▶ detail</a>
15	10094	Diah Puspita Ayu	X-9	0.205	2.12 %	<a href="#">▶ detail</a>
16	10119	Dipta Setyo Rama	X-9	0.203	2.1 %	<a href="#">▶ detail</a>
17	10213	Fajar Jiwandono	X-9	0.203	2.1 %	<a href="#">▶ detail</a>
18	10218	Sami Irfina Unaiyahroya	X-6	0.202	2.09 %	<a href="#">▶ detail</a>
19	10037	Fahreza Fajar Ramadhan	X-5	0.198	2.05 %	<a href="#">▶ detail</a>
20	9947	Dina Anggun Saraswati	X-4	0.197	2.04 %	<a href="#">▶ detail</a>

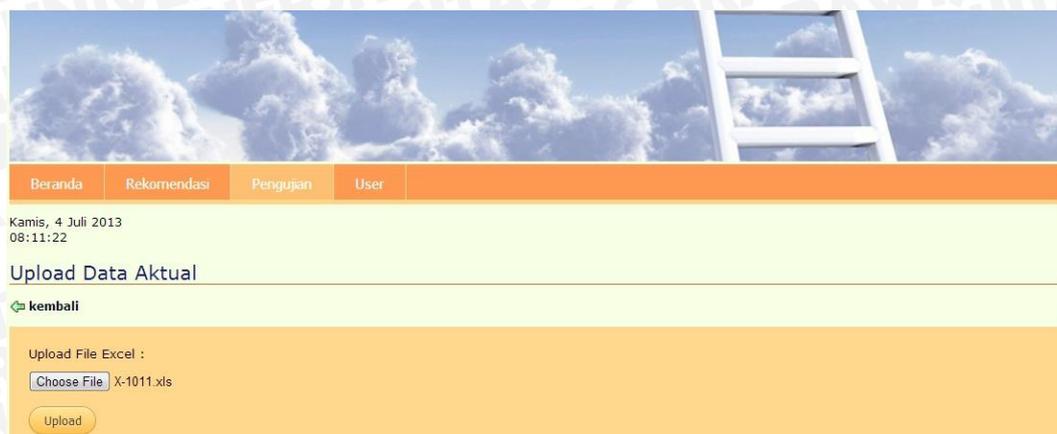
[← kembali](#)

[Download Excel](#)

**Gambar 4.41** Tabel Hasil Rekomendasi

#### 4.3.2.4 Halaman Pengujian

Halaman ini merupakan halaman yang berfungsi untuk proses pengujian perbandingan hasil rekomendasi sistem terhadap data aktual calon penerima beasiswa pada SMAN 7 Malang. Proses pengujian ini dilakukan dengan mengunggah data aktual berupa file excel ke dalam sistem, kemudian dilakukan proses perhitungan besarnya presentase kecocokan hasil rekomendasi beasiswa pada kelas dan tahun ajaran yang telah dipilih sebelumnya. Dan untuk parameter yang dipilih adalah “Nilai Baik” untuk parameter nilai, “Kurang Mampu” untuk parameter penghasilan orang tua, dan “Disiplin” untuk parameter kedisiplinan. Halaman untuk *upload* file excel pada sistem ditunjukkan oleh gambar 4.42. Sedangkan untuk hasil pengujian ditunjukkan oleh gambar 4.43.



**Gambar 4.42** Halaman *Upload File Excel*

20	10103	Dwiki Ardianto	X-6
21	9913	Avryan Harvyandha	X-7
22	9992	Citra Ayu Desinta	X-7
23	10043	Dhimaz Ajeng Larasati	X-7
24	10074	Faiz Alwan Baskoro Resnale	X-7
25	9953	Ariyuana Eka Rahmawati	X-8
26	9953	Ariyuana Eka Rahmawati	X-8
27	10049	Dhea Ramadhan Dinarta	X-8
28	10094	Diah Puspita Ayu	X-9
29	10288	Febriana Ika Syafitri	X-9
30	10372	Hernada Firdaus Eka Saputri	X-9

Terdapat sebanyak 30 data yang sama dari 50 data yang dibandingkan

SEHINGGA => Presentase Kecocokan sistem dengan data Aktual dari kelas X dan tahun ajaran 2010/2011 adalah sebesar 60 %

**Gambar 4.43** Halaman Hasil Pengujian Perbandingan

#### 4.3.2.5 Halaman *User*

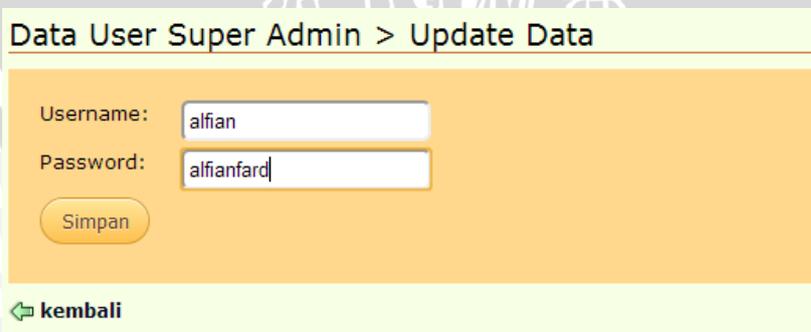
Halaman ini merupakan halaman yang berfungsi untuk mengelola pengguna sistem. Pada halaman ini terdapat fasilitas tambah data *user*, *update* data *user*, dan hapus data *user*. Halaman *User* ditunjukkan oleh gambar 4.44. Sedangkan untuk halaman tambah data *user* ditunjukkan oleh gambar 4.45. Dan untuk halaman *update* data *user* ditunjukkan oleh gambar 4.46.



Gambar 4.44 Halaman User



Gambar 4.45 Halaman Tambah Data User



Gambar 4.46 Halaman Update Data User

## BAB V

### PENGUJIAN DAN ANALISIS

#### 5.1 Implementasi Pengujian

Terdapat empat tahapan pengujian pada skripsi ini, yang pertama adalah pengujian perbandingan sistem yang dilakukan dengan membandingkan data hasil rekomendasi beasiswa dari sistem terhadap data konvensional (aktual) rekomendasi beasiswa dari SMAN 7 Malang. Kemudian terdapat pengujian integrasi (*integration testing*), yaitu pengujian terhadap waktu respon yang diperlukan untuk proses ETL pada data pada datawarehouse. Ketiga adalah pengujian *Balck Box*, yaitu pengujian yang dilakukan terhadap fungsional sistem yang telah dibuat. Dan yang terakhir adalah pengujian *User Acceptance*, yaitu dengan menggunakan metode kuisioner yang akan diberikan kepada pengguna sistem mengenai performa dari sistem yang telah dibuat.

##### 5.1.1 Pengujian Perbandingan Sistem terhadap Data Aktual

Pengujian ini dilakukan oleh sistem dengan cara membandingkan hasil rekomendasi sistem yang menggunakan metode *fuzzy database* terhadap data konvensional (aktual). Sebelum melakukan pengujian ini, dilakukan proses rekomendasi terlebih dahulu dengan menentukan parameter yang mempengaruhi proses rekomendasi yaitu “Nilai Baik” untuk parameter nilai, “Kurang Mampu” untuk parameter penghasilan orang tua, dan “Disiplin” untuk parameter kedisiplinan. Selanjutnya ditentukan kelas dan tahun ajaran siswa yang ingin direkomendasikan sebagai calon penerima beasiswa. Pengujian perbandingan terhadap sistem ini dilakukan dengan mengupload data aktual dalam bentuk microsoft excel, kemudian data hasil upload tersebut akan ditampilkan pada sistem. Selanjutnya, sistem akan melakukan proses untuk mencari data siswa yang sama (cocok) antara hasil rekomendasi sitem dengan data aktual, dan data yang sama tersebut akan ditampilkan pada sistem. Kemudian akan dihitung berapa besar presentase kecocokan antara hasil rekomendasi sistem terhadap data

aktual. Implementasi pengujian akurasi untuk setiap kelas dan tahun ajaran ditunjukkan oleh gambar 5.1, gambar 5.2, gambar 5.3, gambar 5.4, gambar 5.5, dan gambar 5.6.

20	10103	Dwiki Ardianto	X-6
21	9913	Avryan Harvyandha	X-7
22	9992	Citra Ayu Desinta	X-7
23	10043	Dhimaz Ajeng Larasati	X-7
24	10074	Faiz Alwan Baskoro Resnale	X-7
25	9953	Ariyuana Eka Rahmawati	X-8
26	9953	Ariyuana Eka Rahmawati	X-8
27	10049	Dhea Ramadhan Dinarta	X-8
28	10094	Diah Puspita Ayu	X-9
29	10288	Febriana Ika Syafitri	X-9
30	10372	Hernada Firdaus Eka Saputri	X-9

Terdapat sebanyak 30 data yang sama dari 50 data yang dibandingkan

SEHINGGA => Presentase Kecocokan sistem dengan data Aktual dari kelas X dan tahun ajaran 2010/2011 adalah sebesar 60 %

**Gambar 5.1** Hasil Uji Perbandingan Kelas X dan Tahun Ajaran 2010-2011

20	10443	Dicky Putra Prasetya	XI-IPS-2
21	10474	Dwi Lidiana Wati	XI-IPS-3
22	10193	Heidy Anasya Ali Vianti	XI-IPS-3
23	10472	Rizaldy Aji Muzakky	XI-IPS-3
24	10240	Lita Aslama	XI-IPS-3
25	10453	Nurwidya Restu Putriyani	XI-IPS-4
26	10455	Ratry Frestika Putri	XI-IPS-4
27	10039	Dhani Mahendra D.P.	XI-IPS-4
28	10583	Rizal Dwirama Martdiandi Lubis	XI-IPS-4
29	9904	Vita Andryani	XI-IPS-5
30	10212	Fajar Jihan Firdauzi	XI-IPS-5
31	10139	Ekanabilah Amalina	XI-IPS-5

Terdapat sebanyak 31 data yang sama dari 50 data yang dibandingkan

SEHINGGA => Presentase Kecocokan sistem dengan data Aktual dari kelas XI dan tahun ajaran 2010/2011 adalah sebesar 62 %

**Gambar 5.2** Hasil Uji Perbandingan Kelas XI dan Tahun Ajaran 2010-2011

14	9578	Abraham Gunawan Wicaksana	XII-IPS-2
15	9598	Adam Hakim Ardiansyah	XII-IPS-2
16	9850	Nike Kristanti	XII-IPS-3
17	9505	Nivo Haneda	XII-IPS-3
18	9570	Sandya Wahyu Nurhidayat	XII-IPS-3
19	9677	Fanisa Paramitha Auliasari	XII-IPS-4
20	9845	Tito Danang Wahyu Perdana	XII-IPS-4
21	9852	Yona Dwi Novita Sari	XII-IPS-4
22	9682	Farah Fairuz Susanti	XII-IPS-4
23	9844	Nadya Rizky Amanda	XII-IPS-5
24	9514	Rinaldy Sukoco Putro	XII-IPS-5

Terdapat sebanyak 24 data yang sama dari 50 data yang dibandingkan

SEHINGGA => Presentase Kecocokan sistem dengan data Aktual dari kelas XII dan tahun ajaran 2010/2011 adalah sebesar 48 %

**Gambar 5.3** Hasil Uji Perbandingan Kelas XII dan Tahun Ajaran 2010-2011

19	10839	Eren Indra Paripurna	X-6
20	10766	Andri Adhi Wijaya	X-7
21	10779	Yogi Pratama	X-7
22	10911	Winda Widya Mandasari	X-7
23	10623	M. Aris Trianto	X-8
24	10719	Ade Cahyono	X-8
25	10651	Resty Riag Agustia	X-9
26	10739	Intan Savitri	X-9
27	10739	Intan Savitri	X-9
28	10931	Moch. Aqireza Hermawan	X-9
29	10966	Yulia Arifiani	X-9

Terdapat sebanyak 29 data yang sama dari 50 data yang dibandingkan

SEHINGGA => Presentase Kecocokan sistem dengan data Aktual dari kelas X dan tahun ajaran 2011/2012 adalah sebesar 58 %

**Gambar 5.4** Hasil Uji Perbandingan Kelas X dan Tahun Ajaran 2011-2012

14	9909	Arief Ridwan Ma'ruf	XI-IPS-2
15	10160	Frita Alif Rahmanti	XI-IPS-2
16	9998	Candra Dita Mumpuni	XI-IPS-2
17	10376	Rahma Tri Agustini	XI-IPS-3
18	10162	Fadhila Kamelia	XI-IPS-3
19	10186	Januar Wahyu Ramadhany	XI-IPS-4
20	9869	Alamsyah Satriatama	XI-IPS-4
21	10015	Chusnul Liyah	XI-IPS-5
22	10128	Denti Lintang Lajuarida	XI-IPS-5
23	9777	Afni Nurachma Putri	XI-IPS-5
24	9945	Della Novita Sari	XI-IPS-5

Terdapat sebanyak 24 data yang sama dari 50 data yang dibandingkan

SEHINGGA => Presentase Kecocokan sistem dengan data Aktual dari kelas XI dan tahun ajaran 2011/2012 adalah sebesar 48 %

**Gambar 5.5** Hasil Uji Perbandingan Kelas XI dan Tahun Ajaran 2011-2012

17	10193	Heidy Anasya Ali Vianti	XII-IPS-3
18	10437	Novita Wulansari	XII-IPS-3
19	10240	Lita Aslama	XII-IPS-3
20	10269	Mega Febri Rafikasari	XII-IPS-3
21	10455	Ratry Frestika Putri	XII-IPS-4
22	10482	Rexy Anggara Priga	XII-IPS-4
23	9944	Aulia Nisa Firdaus	XII-IPS-4
24	9872	Uvi Datus Sufria	XII-IPS-5
25	10212	Fajar Jihan Firdauzi	XII-IPS-5
26	9936	Salsabila Widdhie A.	XII-IPS-5
27	9983	Sela Yuni Puspita	XII-IPS-5

Terdapat sebanyak 27 data yang sama dari 50 data yang dibandingkan

SEHINGGA => Presentase Kecocokan sistem dengan data Aktual dari kelas XII dan tahun ajaran 2011/2012 adalah sebesar 54 %

**Gambar 5.6** Hasil Uji Perbandingan Kelas XI dan Tahun Ajaran 2011-2012

### 5.1.2 *Integration Testing*

*Integration testing* bertujuan untuk mengetahui akurasi waktu dari skema ETL yang telah dibuat. Data yang digunakan pada pengujian ini adalah *database* asli yang didapat dari data akademik siswa SMAN 7 Malang. Pada pengujian ini skema ETL akan di *run* sebanyak 7 kali dengan jumlah data yang berbeda yaitu 10, 100, 1000, 2000, 5000, 10000, dan 20000 data. Pada pengujian ini akan diketahui perbedaan pengaruh jumlah data terhadap kecepatan waktu yang diperlukan pada proses ETL yang telah dibuat. Skema ETL yang digunakan dalam pengujian ini adalah *dim\_siswa*, *fact\_nilai*, dan *fact\_tatib*, karena ketiga tabel tersebut memiliki jumlah data yang banyak dan untuk tabel *fact\_nilai* serta *fact\_tatib* merupakan tabel yang dibangun dari beberapa tabel sehingga memerlukan proses yang lebih banyak dari tabel lainnya pada *datawarehouse*. Hasil dari *integration testing* yang dilakukan terhadap skema ETL yang telah dibuat ditunjukkan oleh tabel 5.1.

**Tabel 5.1** Hasil *Integration Testing* skema ETL

Tabel	Jumlah data						
	10	100	1000	2000	5000	10000	20000
dim_siswa	0,1 s	0,13 s	0,3 s	-	-	-	-
fact_nilai	0,2 s	0,3 s	0,5 s	0,7 s	1,5 s	1,7 s	3,7 s
fact_tatib	0,2 s	0,22 s	0,4 s	0,5 s	-	-	-

### 5.1.3 Pengujian *Black Box*

Pengujian *Blac Box* ini merupakan pengujian fungsional terhadap sistem yang telah dibuat. Pengujian ini dilakukan untuk mengetahui apakah sistem yang dibuat sudah dapat digunakan secara baik atau tidak. Pengujian *black box* pada sistem dilakukan pada setiap menu atau *service* yang terdapat dalam sistem, selanjutnya dari setiap menu atau *service* tersebut dibuatlah skenario pengujian yang berfungsi untuk melakukan perintah pada sistem tersebut. Hasil dari pengujian *Black Box* dapat dilihat pada lampiran.

### 5.1.4 Pengujian *User Acceptance*

Pengujian *User Acceptance* ini dilakukan dengan memberikan agket kepada pihak SMA Negeri 7 Malang (pihak akademik) untuk menilai performa dari sistem yang telah dibuat berdasarkan pernyataan yang terdapat pada lembar angket tersebut. Hasil dari pengujian ini dapat dilihat pada lampiran.

## 5.2 Analisa Hasil Pengujian

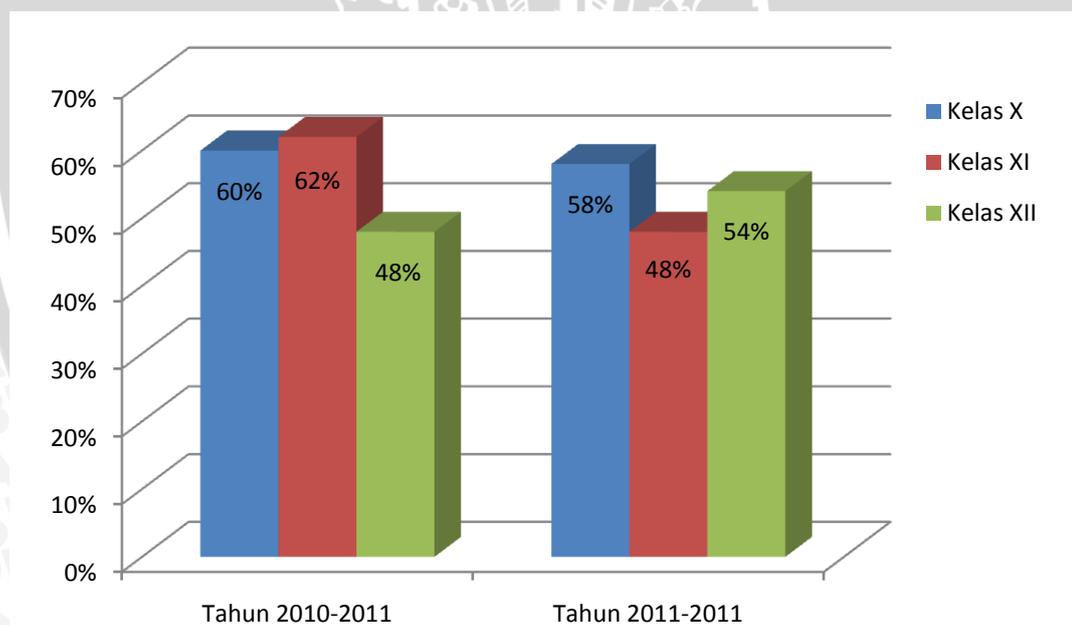
### 5.2.1 Analisa Hasil Perbandingan Sistem terhadap Data Aktual

Pada implementasi pengujian perbandingan terlihat bahwa hasil rekomendasi sistem dan data aktual memiliki perbedaan yang relatif cukup besar pada kelas dan tahun ajaran tertentu. Hal ini dapat dilihat pada tabel 5.2 yang menunjukkan hasil perhitungan pada setiap kelas dan tahun ajaran.

**Tabel 5.2** Tabel Hasil Perhitungan Uji Perbandingan

Tahun Ajaran	Kelas	Perhitungan	Presentase Kecocokan
2010-2011	X	$\frac{30}{50} \times 100\%$ ..... persamaan 2.6	60 %
	XI	$\frac{31}{50} \times 100\%$ ..... persamaan 2.6	62 %
	XII	$\frac{24}{50} \times 100\%$ ..... persamaan 2.6	48 %
2011-2012	X	$\frac{29}{50} \times 100\%$ ..... persamaan 2.6	58 %
	XI	$\frac{24}{50} \times 100\%$ ..... persamaan 2.6	48 %
	XII	$\frac{27}{50} \times 100\%$ ..... persamaan 2.6	54 %

Dari tabel 5.5, hasil pengujian perbandingan dapat dibuat sebuah grafik batang seperti yang ditunjukkan pada gambar 5.7.

**Gambar 5.7** Grafik Batang Presentase Kecocokan Hasil Rekomendasi Sistem terhadap Data Aktual

Berdasarkan gambar 5.7 terlihat bahwa nilai kecocokan yang berbeda-beda pada setiap kelas dan tahun ajaran, yaitu pada tahun ajaran 2010-2011 kelas X sebesar 60 %, pada kelas XI sebesar 62 %, dan pada kelas XII sebesar 48 %.

Sedangkan pada tahun ajaran 2011-2012 kelas X sebesar 58 %, pada kelas XI sebesar 48 %, dan pada kelas XII sebesar 54 %. Perbedaan hasil rekomendasi dari sistem dengan data aktual memang relatif cukup besar, hal ini dikarenakan hasil rekomendasi pada data aktual lebih bersifat subyektif, artinya hanya mengacu atau menekankan pada satu atau dua parameter saja, misalnya hasil rekomendasi pada data aktual hanya menggunakan parameter nilai saja atau penghasilan orang tua saja sebagai acuannya. Sedangkan pada sistem, semua parameter menjadi acuan dalam proses rekomendasi beasiswa.

Perbedaan presentase kecocokan antar kelas yang sama pada tahun ajaran yang berbeda. Hal ini disebabkan oleh beberapa faktor, salah satunya adalah faktor perbedaan poin pelanggaran (tingkat kedisiplinan) siswa atau bisa juga faktor rata-rata nilai siswa pada kelas dan tahun ajaran tertentu. Karena pada kelas tertentu akan ditempati oleh siswa yang berbeda untuk setiap tahun ajaran. Dan siswa pada kelas tersebut juga akan memiliki rata-rata nilai, poin pelanggaran, serta penghasilan orang tua yang berbeda pula. Sehingga akan mengakibatkan terjadinya perbedaan presentase hasil perbandingan pada setiap kelas dan tahun ajaran tertentu.

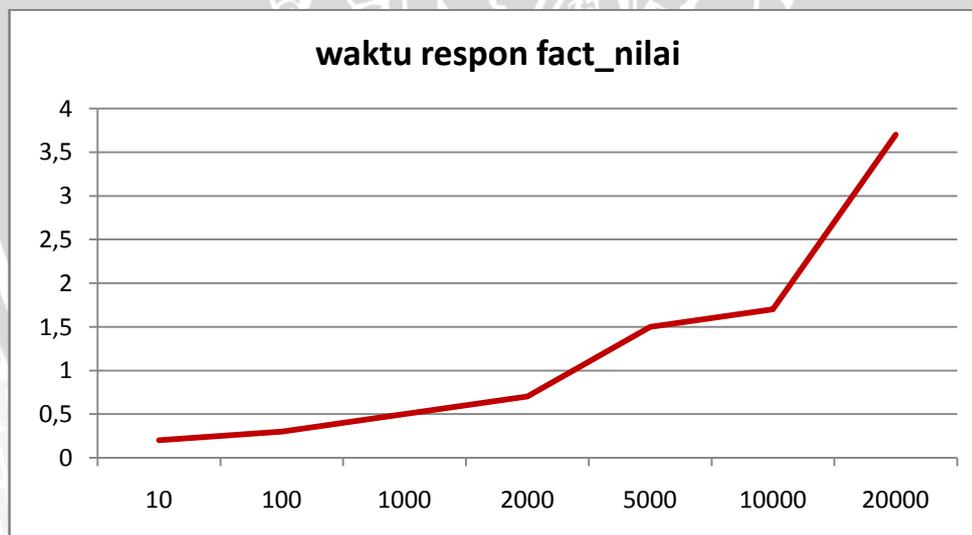
### 5.2.2 Analisa Hasil *Integration Testing*

Dari hasil implementasi *integration testing*, dapat dibuat sebuah grafik untuk setiap tabel terhadap lamanya waktu respon proses ETL tabel tersebut. Grafik waktu respon tabel `dim_siswa` ditunjukkan oleh gambar 5.8, sedangkan waktu respon tabel `fact_nilai` ditunjukkan oleh gambar 5.9, dan untuk waktu respon tabel `fact_tatib` ditunjukkan oleh gambar 5.10.



**Gambar 5.8** Grafik Waktu Respon Tabel dim\_siswa

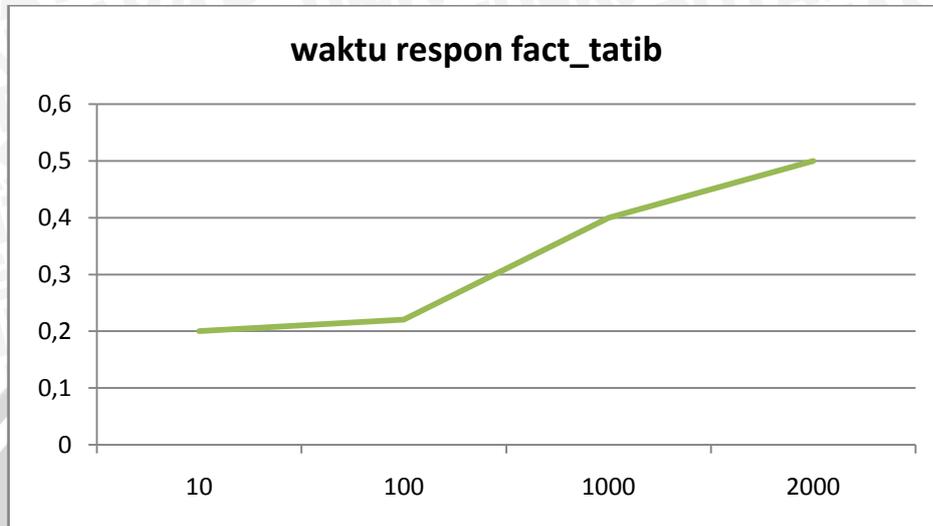
Dari gambar 5.8 dapat dilihat bahwa jumlah data mempengaruhi waktu proses eksekusi (waktu respon) skema ETL. Semakin banyak data, maka semakin lama pula waktu respon skema ETL nya. Untuk jumlah data dengan kelipatan 10, waktu yang diperlukan untuk eksekusi tidak terlalu berbeda, tetapi jika kelipatan 100 data, waktu respon mengalami kenaikan yang cukup signifikan.



**Gambar 5.9** Grafik Waktu Respon Tabel fact\_nilai

Sama halnya dengan tabel fact\_nilai, semakin banyak data, maka semakin lama pula waktu respon skema ETL nya. Untuk jumlah data dengan kelipatan 10, waktu yang diperlukan untuk eksekusi tidak terlalu berbeda, tetapi jika kelipatan 100 data, waktu respon mengalami kenaikan yang cukup signifikan.

Sedangkan untuk kelipatan 1000 data dan 10000 data, waktu respon mengalami kenaikan yang relatif besar pada proses skema ETL.



**Gambar 5.10** Grafik Waktu Respon Tabel fact\_tatib

Untuk tabel fact\_tatib, jumlah data juga mempengaruhi besarnya waktu respon terhadap skema ETL. Semakin banyak data, maka semakin lama pula waktu respon skema ETL tabel fact\_tatib.

Selain itu juga terdapat perbedaan waktu respon untuk setiap tabel pada masing-masing kelipatan jumlah data. Misalkan untuk kelipatan 1000 data, waktu respon yang dibutuhkan tabel dim\_siswa pada skema ETL sebesar 0,3 s, sedangkan pada tabel fact\_nilai sebesar 0,5 s, dan pada tabel fact\_tatob sebesar 0,4 s. Hal ini disebabkan oleh perbedaan banyaknya skema ETL yang diproses dalam membentuk dan menghasilkan tabel dimensi dan fakta pada *datawarehouse*. Semakin banyak dan rumit skema ETL yang digunakan dalam membentuk tabel dimensi dan tabel fakta, maka semakin lama pula waktu respon yang dibutuhkan dalam proses ETL. Pada tabel fact\_nilai memiliki skema ETL yang paling banyak, sehingga proses ETL nya memerlukan waktu yang paling lama (dalam setiap kelipatan jumlah data) daripada semua tabel yang ada pada *datawarehouse*.

### 5.2.3 Analisa Hasil Pengujian *Black Box*

Setelah melakukan pengujian *Black Box*, yaitu dimana sistem telah diuji secara fungsional dengan melakukan skenario pengujian berdasarkan semua menu atau *service* yang terdapat dalam sistem. Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa secara garis besar sistem telah dapat digunakan dan dioperasikan dengan baik, walaupun terdapat beberapa skenario pengujian yang gagal dan tidak dapat dilakukan oleh sistem. Skenario yang gagal dilakukan oleh sistem adalah :

1. Sistem tidak bisa menangani sintaks *SQL Injections* jika sintaks tersebut diterapkan pada saat melakukan Log In, sintaks tersebut diisikan pada *text field* username dan password.
2. Sistem tidak dapat memberikan solusi jika jumlah penerima pada proses rekomendasi diisikan bilangan negatif. Sistem akan tetap melakukan proses rekomendasi, tetapi sistem tidak dapat menampilkan daftar siswa yang berhak menerima beasiswa.
3. Sistem tidak menampilkan informasi password yang dimiliki setiap *user* dalam bentuk simbol, sehingga jika seorang *user* melihat menu kelola *user*, maka akan dapat melihat username dan password dari *user* lain.

### 5.2.4 Analisa Hasil Pengujian *User Acceptance*

Setelah melakukan pengujian *User Acceptance*, yaitu dimana sistem telah diuji secara langsung oleh pengguna (pihak SMAN 7 Malang) dengan menggunakannya secara fungsional dan melihat hasil keputusan yang dihasilkan. Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa secara garis besar sistem telah dapat diterima dan dapat memenuhi kebutuhan pengguna.

## BAB VI PENUTUP

### 6.1 Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

1. Teknologi OLAP dapat diterapkan dengan melakukan proses ETL pada sumber data berupa *database* dari data akademik siswa SMA Negeri 7 Malang. Hasil dari proses ETL akan dimasukkan kedalam sebuah *datawarehouse* yang nantinya akan dipanggil pada *front end user* untuk menganalisa data akademik siswa yang terkait dengan proses rekomendasi beasiswa serta analisa data akademik siswa juga dapat dilihat dari berbagai sudut pandang dimensi. Waktu respon yang dihasilkan dari proses ETL sangat dipengaruhi oleh jumlah data yang diproses pada *database* akademik siswa SMAN 7 Malang. Selain itu, waktu respon juga sangat dipengaruhi oleh banyaknya proses ETL dalam membentuk setiap tabel pada skema *datawarehouse*.
2. Metode *Fuzzy Database* dapat diimplementasikan untuk mendukung keputusan rekomendasi beasiswa pada SMA Negeri 7 Malang, dalam bentuk aplikasi yang berbasis website. Langkah-langkah utama yang diterapkan dalam aplikasi adalah :
  - a. Penentuan parameter yang diperlukan untuk proses rekomendasi yang diambil dari *database* sumber, yaitu penghasilan orang tua, nilai siswa, dan kedisiplinan siswa (poin pelanggaran tata tertib siswa).
  - b. Proses rekomendasi beasiswa dengan *fuzzy database* ini memiliki tiga proses utama yaitu menentukan batasan setiap parameter, menentukan derajat keanggotaan setiap parameter, dan menghitung *firestrength* setiap siswa berdasarkan parameter yang dipilih.

3. Pada hasil pengujian akurasi yaitu membandingkan data aktual dengan hasil rekomendasi sistem menggunakan *fuzzy database* diperoleh nilai akurasi yang berbeda pada setiap kelas dan tahun ajaran tertentu, yaitu pada tahun ajaran 2010-2011 kelas X sebesar 60 %, pada kelas XI sebesar 62 %, dan pada kelas XII sebesar 48 %. Sedangkan pada tahun ajaran 2011-2012 kelas X sebesar 58 %, pada kelas XI sebesar 48 %, dan pada kelas XII sebesar 54 %. Perbedaan hasil rekomendasi dari sistem dengan data aktual memang relatif cukup besar, hal ini dikarenakan hasil rekomendasi pada data aktual lebih bersifat subyektif, artinya hanya mengacu atau menekankan pada satu atau dua parameter saja, misalnya hasil rekomendasi pada data aktual hanya menggunakan parameter nilai saja atau penghasilan orang tua saja sebagai acuannya. Sedangkan pada sistem, semua parameter menjadi acuan dalam proses rekomendasi beasiswa. Sedangkan untuk kelas pada tahun ajaran yang berbeda juga memiliki perbedaan nilai akurasi kecocokannya, hal ini disebabkan setiap kelas pada tahun ajaran yang berbeda memiliki siswa yang berbeda pula, sehingga secara otomatis nilai siswa, penghasilan orang tua, dan nilai kedisiplinan juga berbeda.
4. Berdasarkan hasil pengujian *black box*, terdapat skenario pengujian yang gagal dan tidak bisa dijalankan oleh sistem, antara lain :
  - a. Sistem tidak bisa menangani sintaks *SQL Injections* jika sintaks tersebut diterapkan pada saat melakukan Log In, sintaks tersebut diisikan pada *text field* username dan password.
  - b. Sistem tidak dapat memberikan solusi jika jumlah penerima pada proses rekomendasi diisikan bilangan negatif. Sistem akan tetap melakukan proses rekomendasi, tetapi sistem tidak dapat menampilkan daftar siswa yang berhak menerima beasiswa.
  - c. Sistem tidak menampilkan informasi password yang dimiliki setiap *user* dalam bentuk simbol, sehingga jika seorang *user* melihat menu kelola *user*, maka akan dapat melihat username dan password dari *user* lain.

5. Berdasarkan hasil pengujian *user acceptance*, dapat disimpulkan bahwa secara garis besar sistem telah dapat diterima dan dapat memenuhi kebutuhan pengguna.

## 6.2 Saran

Saran yang dapat diberikan berdasarkan hasil yang didapat dari penelitian ini yaitu :

1. Diperlukan suatu adanya optimalisasi pada *database* OLTP dan *database datawarehouse* untuk menghasilkan waktu respon yang lebih baik pada proses ETL, sehingga proses rekap laporan dapat dilakukan lebih cepat dalam jumlah data yang banyak.
2. Dapat ditambahkan dengan suatu metode yang memberikan bobot pada setiap parameter seperti metode TOPSIS, MADM, ANP, atau SAW, sehingga hasil perhitungan akhir yang mempengaruhi besarnya nilai rekomendasi beasiswa dapat bersifat lebih obyektif (merata pada semua parameter) dan agar hasil rekomendasi dapat lebih optimal.
3. Diperlukan adanya perbaikan dalam sistem, sehingga semua skenario pengujian yang terdapat dalam setiap menu atau *service* pada sistem dapat dilakukan dengan baik oleh sistem.

## DAFTAR PUSTAKA

- [CON-02] Connoly, T. M. And Begg C.E. 2002. *Database Systems : A Practical Approach In Design, Implementation, And Management, 3th Edition*. Addison-Wesley. USA.
- [CON-05] Connolly, Thomas dan Carolyn Begg. 2005. *Database Systems : A Practical Approach to Design, Implementation, And Management, 4th Edition*. Addison Wesley Publishing Company Inc. California.
- [BOU-09] Bouman R, Doungen J. 2009. *Pentaho Solution : Business Intelligence and Data Warehousing with Pentaho and MySQL*. Indianapolis : Wiley Publishing, Inc.
- [CRI-05] Cristiono, Denny. 2005. *Aplikasi Pendukung Keputusan Dengan Menggunakan Logika Fuzzy (studi kasus Pemilihan Handphone Berdasarkan Kebutuhan Konsumen)*. Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana. Salatiga.
- [GOL-09] Golfarelli, Matteo, dan Rizzi, Stefano. 2009. *Datawarehouse Design (Modern Principles and Methodologies)*. Mc Graw Hill. New York.
- [HAN-01] Han, Jiawei dan Kamber Micheline. 2001. *Data Mining : Concepts and techniques*. Morgan Kaufmann. San Fransisco.
- [HAN-06] Han, Jiawei dan Kamber Micheline. 2006. *Data Mining : Concept and Techniques Second Edition*. Morgan Kaufmann Publishers. San Fransisco.
- [HEL-01] Hellmann, M. 2001. *Fuzzy Logic Introduction*. Laboratoire Antennes Radar Telecom. F.R.E CNRS 2272. Equipe Radar Polarimetrie.
- [HOF-05] Hoffer, Jeffrey A., et.al. 2005. *“Modern Systems Analysis and Design”*, 4th ed. Pearson Prentice Hall. New Jersey.
- Inmon, W.H. 2002. *Building the Datawarehouse Third Edition*. Willey Publishing, Inc. Indianapolis.

- [KIM-04] Kimball, Ralph. 2004. *The Datawarehouse ETL Toolkit*. Wiley Publishing. New York.
- [KOS-02] Kosasi, Sandy. 2002. *Sistem Penunjang Keputusan (Decision Support System)*. Pontianak: Sekolah Tinggi Manajemen Informatika dan Komputer.
- [KUS-04] Kusumadewi, Sri dan Hari Purnomo. 2004. *Aplikasi Logika Fuzzy untuk Mendukung Keputusan*. Graha Ilmu. Yogyakarta.
- [LAN-02] Lane, Paul. 2002. *Oracle 9i Data Warehousing Guide Release 1 (9.0.1)*. Oracle Corporation. California, USA.
- [MAL-00] Mallach, Efrem G. 2000. *Decision support and datawarehouse systems*. McGraw Hill. New York.
- [NAD-03] Nadlir, Syariful dan Oon Amroni. 2003. *Teknologi Sistem Fuzzy, Jurnal Komputer dan Informatika 4*. FTI, Universitas Tarumanagara. Jakarta.
- [PON-01] Ponniah, Paulraj. 2001. *Data Warehousing Fundamentals*. John Wiley & Son. New York .
- [PRE-02] Pressman, Roger S, 2002, *Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku I)*, Andi, Yogyakarta.
- [SUB-02] Subakti, Irfan. 2002. *Sistem Pendukung Keputusan (Decision Support System)*. Surabaya: Jurusan Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh November.
- [TUR-05] Turban, E, Aronson J. E, dan Liang Ting-Peng. 2005. *Decision Support System and Intellegent System (Sistem Pendukung Keputusan dan Sistem Cerdas)*. Jilid I. Edisi 7. Terjemahan Dwi Prabantini. Yogyakarta: Andi.
- [TRU-02] Trujillo, Juan dan Sergio Lujan Mora. 2003. *A UML Based Approach for Modeling ETL Processes inDatawarehouse*. Springer Verlag. Chicago.

- [WIL-00] William, C. Amo. 2000. *Microsoft SQL Server OLAP Developer's Guide*. IDG Books Worldwide. Foster City, CA.
- [WAY-00] Wayne, S. Freeze. 2000. *Unlocking OLAP with Microsoft SQL Server and Excel 2000*. IDG Books World-wide. Foster City, CA.
- [AN0-10] Anonymous, 2010. ETL datawarehouse  
<http://www.gudangmateri.com/2010/04/etl-data-warehouse.html> diakses pada tanggal 4 Juni 2012.

