

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penentuan Subyek

Pada tahun 2014 pihak universitas memiliki program kerja untuk mengembangkan suatu sistem pangkalan data. Pangkalan data dibuat untuk menunjang pengisian borang akreditasi khususnya standar 3 mengenai mahasiswa dan lulusan. Pada standar 3 terdapat dua jenis subjek yang dapat dipilih yaitu mahasiswa dan lulusan. Berdasarkan hal tersebutlah subjek pangkalan data ditentukan, yaitu mahasiswa.

4.2. Hasil Analisis Kebutuhan dan Analisis Ketersediaan Data

Sebelum melaksanakan pembuatan pangkalan data, tahapan yang dilakukan adalah analisis kebutuhan. Pada penelitian ini analisis kebutuhan berasal dari dua sumber, yaitu dokumen borang akreditasi dan *stakeholder* dari berbagai unit kerja universitas.

Pada dokumen borang akreditasi terdapat tujuh standar yang merupakan unsur-unsur dalam penilaian akreditasi. Dari tujuh standar tersebut hanya salah satu yang menjadi fokus pengamatan dalam penelitian ini, yaitu borang akreditasi standar 3 tentang mahasiswa dan lulusan dengan subjek mahasiswa.

Selain kebutuhan yang berasal dari dokumen borang akreditasi juga terdapat kebutuhan tambahan dari *stakeholder* yang masih terkait dengan mahasiswa. Kebutuhan ini didapatkan melalui cara berdiskusi bersama *stakeholder* dari

beberapa unit kerja. Dari hasil diskusi ini didapatkan beberapa kebutuhan tambahan yang hasilnya telah digabungkan bersama dengan kebutuhan dokumen borang akreditasi. Hasil Analisis kebutuhan akan menjadi acuan untuk apa pangkalan data ini dibuat dan menemukan data seperti apa saja yang diperlukan dalam pembuatan pangkalan data.

Tahapan kedua dalam penelitian ini adalah analisis ketersediaan data. analisis ketersediaan data bermaksud untuk mengetahui dan memastikan apakah data yang terkait dengan kebutuhan tersedia pada basis data. Dalam penelitian ini terdapat tiga basis data yang digunakan sebagai sumber, yaitu SIMAK, Mastertabel dan Pembayaran. Dari ketiga sumber basis data inilah terdapat sumber data yang dipergunakan dalam pembuatan pangkalan data.

Basis data SIMAK memiliki beberapa tabel yang berguna untuk keperluan operasional akademik seperti data mahasiswa, mata kuliah mahasiswa maupun nilai mata kuliah dari seluruh mahasiswa. Mastertabel merupakan basis data yang memiliki tabel master atau tabel-tabel independen yang tidak memiliki ketergantungan terhadap tabel lainnya seperti tabel mata kuliah, tabel nilai, maupun tabel program studi

Hasil dari analisis kebutuhan dan hasil analisis ketersediaan data telah diolah ke dalam satu tabel. Pada tabel tersebut akan dipaparkan jenis informasi yang dibutuhkan serta keterangan ketersediaan data pada basis data dan sumber data yang dibutuhkan untuk memenuhi kebutuhan. Adapun tabel yang dimaksud dapat dilihat pada Tabel 6.

Tabel 6. Hasil Analisis Kebutuhan dan Analisis Ketersediaan Data

Sumber Kebutuhan	Kategori	Jenis Informasi	Ketersediaan Data	Sumber Data
Borang Akreditasi Program Studi Sarjana(2008)	Data Mahasiswa	Jumlah Mahasiswa Baru Regular bukan Transfer lima tahun terakhir	Tersedia	SIMAK
		Jumlah Mahasiswa Baru Regular Transfer lima tahun terakhir	Tersedia	SIMAK
		Jumlah Mahasiswa Baru NonRegular bukan Transfer lima tahun terakhir	Tersedia	SIMAK
		Jumlah Mahasiswa Baru NonRegular Transfer lima tahun terakhir	Tersedia	SIMAK
		Jumlah Total Mahasiswa Regular bukan Transfer lima tahun terakhir	Tersedia	SIMAK & pembayaran
		Jumlah Total Mahasiswa Regular Transfer lima tahun terakhir	Tersedia	SIMAK & pembayaran
		Jumlah Total Mahasiswa NonRegular bukan Transfer lima tahun terakhir	Tersedia	SIMAK & pembayaran
		Jumlah Total Mahasiswa NonRegular Transfer lima tahun terakhir	Tersedia	SIMAK & pembayaran
	Data Pencapaian Prestasi	Data Prestasi Akademik dan Non-Akademik	Belum tersedia	-
	Data Layanan Mahasiswa	Bimbingan dan Konseling	Belum tersedia	-
		Minat dan Bakat (Ekstra Kurikuler)	Belum tersedia	-
		Pembinaan Soft Skill	Belum tersedia	-
		Beasiswa	Belum tersedia	-
Kesehatan		Belum tersedia	-	

(Lanjutan) Tabel 6. Hasil Analisis Kebutuhan dan Analisis Ketersediaan Data

Sumber Kebutuhan	Kategori	Jenis Informasi	Ketersediaan Data	Sumber Data
<i>Stakeholder</i>	Data Mahasiswa	Profil Mahasiswa	Tersedia	SIMAK
		Data IP Per Semester	Tersedia	SIMAK
		Data Aktifitas Organisasi/UKM	Belum tersedia	-
		Data Presensi Mahasiswa	Belum tersedia	-
		Data Distribusi Nilai Per Mata Kuliah	Tersedia	SIMAK
		Data Tracer Study Mahasiswa lima tahun terakhir	Belum tersedia	-
		Data student exchange	Belum tersedia	-
		Data Character Building Mahasiswa (Kualitatif)	Belum tersedia	-
	Data Status Mahasiswa	Profil Mahasiswa Aktif Kuliah	Tersedia	SIMAK & pembayaran
		Profil Mahasiswa Cuti Kuliah	Belum tersedia	-
		Profil Mahasiswa Kena Sanksi	Belum tersedia	-
		Profil Mahasiswa Pindah Kuliah	Belum tersedia	-
		Profil Mahasiswa Meninggal	Belum tersedia	-

(Lanjutan) Tabel 6. Hasil Analisis Kebutuhan dan Analisis Ketersediaan Data

Sumber Kebutuhan	Kategori	Jenis Informasi	Ketersediaan Data	Sumber Data
<i>Stakeholder</i>	Data Pencapaian Prestasi	Data Keikutsertaan Lomba & Prestasinya	Belum tersedia	-
		Data Paper & Call Paper Mahasiswa tingkat Nasional dan Internasional	Belum tersedia	-
	Data Layanan Mahasiswa	Data Jenis Beasiswa	Belum tersedia	-
		Data Mahasiswa Penerima Beasiswa	Belum tersedia	-

4.3. Hasil Perancangan *Star Schema*

Perancangan *star schema* mengacu dari hasil matriks bus. Matriks bus mendefinisikan proses bisnis yang berasal dari hasil analisis kebutuhan. Adapun matriks bus yang dihasilkan ditunjukkan pada Tabel 7.

Tabel 7. Matriks bus

Proses bisnis	<i>Dimension</i>											
	mahasiswa	Status_mahasiswa	Status_terdaftar	Faculty	Department	Course	Nilai_huruf	Term	Thajaran	Status_transfer	Class_program	angkatan
Perkuliahan	×	×	×	×	×	×	×	×	×	×	×	×

Star schema merupakan tabel yang terdiri dari satu tabel *fact* dan beberapa tabel *dimension* yang memiliki *surrogate key* sebagai *primary key*. Baris pada tabel 7 menjadi tabel *fact* dan kolom menjadi *dimension*.

Tabel *fact* terdiri dari beberapa *foreign key* dan kolom yang memiliki nilai *measure*. Dalam pembuatan pangkalan data ini hanya terdapat satu tabel *fact_perkuliahan* yang memiliki nilai *measure* seperti SKS, bobot nilai, IP dan IPK.

Tabel *dimension* memiliki *surrogate key* yang bersifat *auto increment* dan dimulai dari angka nol. Tabel *dimension* mendukung penjelasan isi data pada tabel *fact*. Adapun tabel *dimension* yang dirancang adalah *dim_faculty*, *dim_mahasiswa*,

dim_department, dim_course, dim_nilai_huruf, dim_term, dim_thajaran, dim_angkatan, dim_status_mahasiswa, dim_status_transfer, dim_class_program, dan dim_status_terdaftar.

Tabel dim_mahasiswa merupakan tabel *dimension* yang menyimpan data profil mahasiswa. Profil mahasiswa yang tersedia pada tabel ini meliputi nama mahasiswa, nomor mahasiswa, nomor telepon, email, jenis kelamin, agama, tempat lahir, dan tanggal lahir. Tabel ini memiliki *primary key* pada kolom student_key yang menjadi *foreign key* pada tabel fact_perkuliahan. Adapun struktur tabel dim_mahasiswa dapat dilihat pada Tabel 8.

Tabel 8. Struktur tabel dim_mahasiswa

Nama Kolom	Keterangan
Student_key	Int, <i>Primary Key</i> , <i>Surrogate Key</i>
Studentid	Varchar(15)
Fullname	Varchar(50)
Placeofbirth	Varchar(50)
Dateofbirth	Datetime
Pembayar	Varchar(15)
Telp	Varchar(30)
Email	Varchar(60)
Nama_gender	Varchar(9)
Nama_agama	Varchar(7)

Tabel dim_faculty merupakan tabel yang menyimpan data mengenai berbagai fakultas beserta keterangannya. Keterangan yang ada meliputi kode fakultas, nama fakultas dan kode singkatan fakultas. *Primary key* pada tabel ini berada pada kolom faculty_key. Adapun struktur tabel ditunjukkan oleh Tabel 9.

Tabel 9. Struktur tabel dim_faculty

Nama Kolom	Keterangan
Faculty_key	Smallint, <i>Primary Key</i> , <i>Surrogate Key</i>
Faculty_id	Tinyint
Name_of_faculty	Varchar(17)
Faculty_id_huruf	Varchar(4)

Tabel dim_department merupakan tabel yang memiliki data berbagai *department* atau program studi. Tabel ini menjelaskan tentang nama program studi, kode program studi dan kode singkatan program studi. Kolom yang menjadi *primary key* pada tabel ini adalah *department_key*. Struktur tabel tabel dim_department dapat dilihat pada Tabel 10.

Tabel 10. Struktur tabel dim_department

Nama Kolom	Keterangan
Department_key	Smallint, <i>Primary Key</i> , <i>Surrogate Key</i>
Department_id	Smallint
Name_of_department	Varchar(38)
Department_id_huruf	Varchar(4)

Tabel dim_term merupakan tabel yang berisi data mengenai jenis-jenis semester, seperti semester gasal, genap, dan pendek. Tabel ini memiliki term_key sebagai *primary key*. Adapun tabel dim_term dapat dilihat pada Tabel 11.

Tabel 11. Struktur tabel dim_term

Nama Kolom	Keterangan
Term_key	Smallint, <i>Primary Key</i> , <i>Surrogate Key</i>
Termid	Tinyint
Termname	Varchar(15)

Tabel *dim_thajaran* merupakan tabel yang menerangkan tentang tahun ajaran atau juga dikenal dengan tahun akademik. *Primary key* pada tabel ini adalah *thajaran_key*. Adapun struktur tabel *dim_thajaran* dapat dilihat pada Tabel 12.

Tabel 12. Struktur tabel *dim_thajaran*

Nama Kolom	Keterangan
<i>Thajaran_key</i>	Smallint, <i>Primary Key</i> , <i>Surrogate Key</i>
<i>Thajaranid</i>	Smallint
<i>Startdate</i>	Datetime
<i>Enddate</i>	Datetime
<i>Description</i>	Varchar(10)

Tabel *dim_angkatan* merupakan tabel yang menyimpan data terkait angkatan meliputi nama angkatan dan tanggal awal mulai terhitungnya mahasiswa dalam suatu angkatan. *Primary key* pada tabel ini adalah *Angkatan_key*. Adapun struktur tabel *dim_angkatan* dapat dilihat pada Tabel 13.

Tabel 13. Struktur tabel *dim_angkatan*

Nama Kolom	Keterangan
<i>Angkatan_key</i>	Smallint, <i>Primary Key</i> , <i>Surrogate Key</i>
<i>Angkatan_id</i>	Smallint
<i>Startdate_of_acceptance</i>	Datetime
<i>Enddate_of_acceptance</i>	Datetime
<i>Name_of_angkatan</i>	Varchar(9)

Tabel *dim_status_transfer* merupakan tabel yang berisi data mengenai jenis status transfer mahasiswa seperti non transfer dan mahasiswa dari D3 atau S1. Selain jenis-jenis status transfer pada tabel ini juga berisi singkatan dari tiap jenis status transfer. Struktur tabel *dim_status_transfer* ditunjukkan pada Tabel 14.

Tabel 14. Struktur tabel dim_status_transfer

Nama Kolom	Keterangan
Status_transfer_key	Smallint, <i>Primary Key, Surrogate Key</i>
Status_transfer_id	Tinyint
Nama_status_transfer	Varchar(27)
Singkatan	Varchar(10)

Tabel dim_status_mahasiswa merupakan tabel yang memiliki data mengenai jenis-jenis status mahasiswa. Tabel ini merupakan tabel tambahan yang dibuat untuk mendukung pembuatan pangkalan data dan bukan merupakan tabel yang berasal dari basis data universitas. Struktur tabel ditunjukkan pada Tabel 15.

Tabel 15. Struktur tabel dim_status_mahasiswa

Nama Kolom	Keterangan
Status_mahasiswa_key	Smallint, <i>Primary Key, Surrogate Key</i>
Status_mahasiswa_id	Tinyint
Status_mahasiswa	Varchar(5)

Tabel dim_class_program merupakan tabel yang menyimpan data mengenai jenis-jenis program kelas seperti regular dan internasional. Struktur tabel ditunjukkan pada Tabel 16.

Tabel 16. Struktur tabel dim_class_program

Nama Kolom	Keterangan
Classprog_key	Smallint, <i>Primary Key, Surrogate Key</i>
Classprog_id	Tinyint
Name_of_class_program	Varchar(21)
Description	Varchar(1)

Tabel `dim_status_tranfer` merupakan tabel yang memiliki data mengenai jenis-jenis status terdaftar mahasiswa seperti mahasiswa baru, mahasiswa pindahan dari universitas lain, dan *student exchange*. *Primary key* pada tabel ini adalah `status_terdaftar_key`. Struktur tabel ditunjukkan pada Tabel 17.

Tabel 17. Struktur tabel `dim_status_terdaftar`

Nama Kolom	Keterangan
<code>Status_terdaftar_key</code>	Smallint, <i>Primary Key</i> , <i>Surrogate Key</i>
<code>Status_terdaftar_id</code>	Tinyint
<code>Name_of_status</code>	Varchar(38)
<code>Description</code>	Varchar(2)

Tabel `dim_course` merupakan tabel yang memiliki data daftar mata kuliah yang ada di universitas. Dalam rancangan ini terdapat `faculty_id` dan `department_id` yang merupakan kolom dari tabel lain. Keberadaan kedua kolom dikarenakan ada kondisi dimana terdapat nilai `course_id` sama dan hanya dibedakan oleh `faculty_id` dan `department_id`. Struktur tabel ditunjukkan pada Tabel 18.

Tabel 18. Struktur tabel `dim_course`

Nama Kolom	Keterangan
<code>Course_key</code>	Int, <i>Primary Key</i> , <i>Surrogate Key</i>
<code>Course_id</code>	Varchar(12)
<code>Faculty_id</code>	Tinyint
<code>Department_id</code>	Smallint
<code>Name_of_course</code>	Varchar(82)

Tabel `dim_nilai_huruf` merupakan tabel yang menyimpan data jenis-jenis nilai beserta bobot dari setiap nilai. Setiap program studi memiliki aturan masing-

masing dalam penentuan nilai. Hal ini menyebabkan tabel mengandung kolom `faculty_id` dan `department_id`. Struktur tabel ditunjukkan pada Tabel 19.

Tabel 19. Struktur tabel `dim_nilai_huruf`

Nama Kolom	Keterangan
<code>Nilai_huruf_key</code>	<code>Int, Primary Key, Surrogate Key</code>
<code>Nilai_huruf</code>	<code>Varchar(2)</code>
<code>Faculty_id</code>	<code>Tinyint</code>
<code>Department_id</code>	<code>Smallint</code>
<code>Bobot_nilai</code>	<code>Decimal(5,2)</code>

Tabel `fact_perkuliahan` merupakan tabel yang berisi beberapa *foreign key* dan kolom yang memiliki nilai-nilai berupa angka. *Foreign key* pada tabel ini merupakan *primary key* pada tabel *dimension*. Pada tabel ini terdapat beberapa kolom yang memiliki nilai *measure*, yaitu `SKS`, `bobot_nilai`, `IP` dan `IPK`. Adapun struktur tabel `fact_perkuliahan` ditampilkan pada Tabel 20.

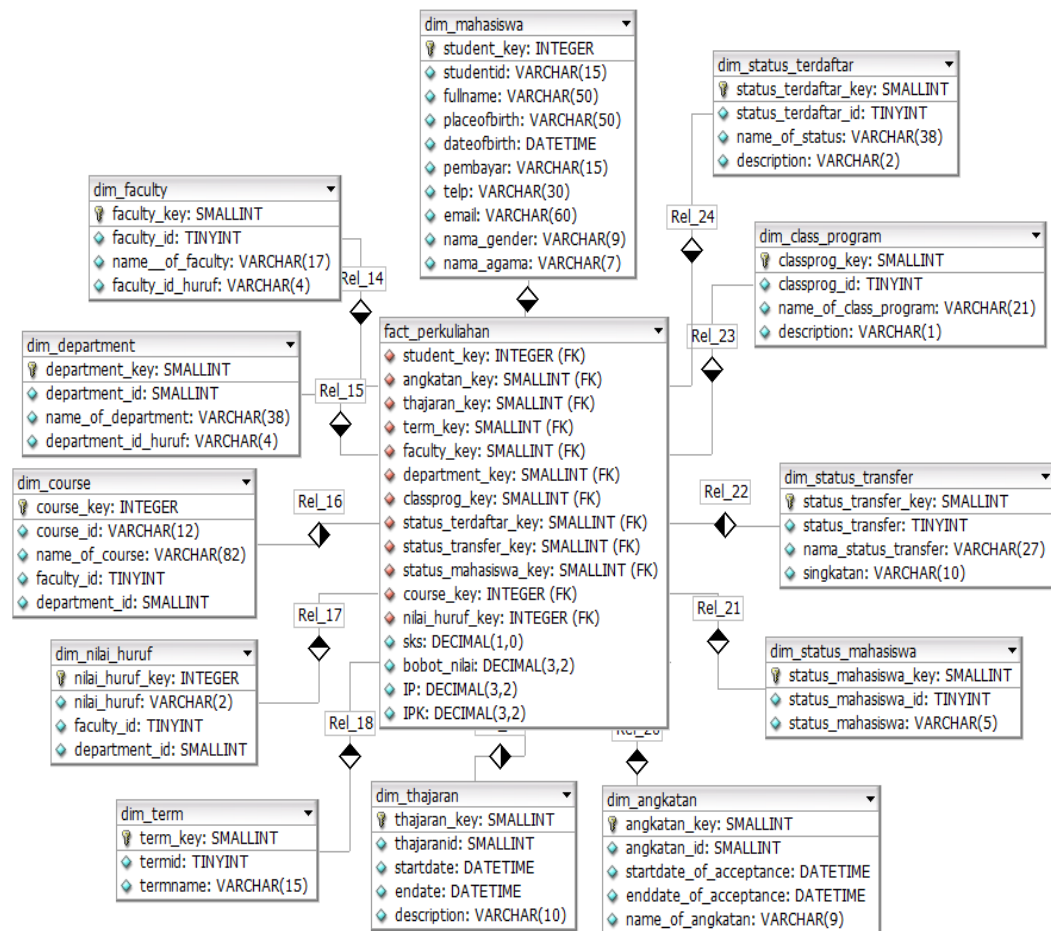
Tabel 20. Struktur tabel `fact_perkuliahan`

Nama Kolom	Keterangan
<code>Student_key</code>	<code>Int, Foreign Key</code>
<code>Angkatan_key</code>	<code>Smallint, Foreign Key</code>
<code>Thajaran_key</code>	<code>Smallint, Foreign Key</code>
<code>Term_key</code>	<code>Smallint, Foreign Key</code>
<code>Faculty_key</code>	<code>Smallint, Foreign Key</code>
<code>Department_key</code>	<code>Smallint, Foreign Key</code>
<code>Classprog_key</code>	<code>Smallint, Foreign Key</code>
<code>Status_terdaftar_key</code>	<code>Smallint, Foreign Key</code>
<code>Status_transfer_key</code>	<code>Smallint, Foreign Key</code>
<code>Status_mahasiswa_key</code>	<code>Smallint, Foreign Key</code>
<code>Course_key</code>	<code>Int, Foreign Key</code>
<code>Nilai_huruf_key</code>	<code>Int, Foreign Key</code>

(Lanjutan) Tabel 20. Struktur tabel fact_perkuliah

Nama Kolom	Keterangan
Sks	Decimal(1,0)
Bobot_nilai	Decimal(3,2)
IP	Decimal(3,2)
IPK	Decimal(3,2)

Berdasarkan penjelasan struktur tabel dari masing-masing tabel *dimension* dan tabel *fact* maka dapat dilihat rancangan *star schema* secara keseluruhan beserta relasi antar tabel pada Gambar 11.



Gambar 11. Hasil rancangan star schema

4.4. Hasil Pembuatan Pangkalan Data

4.4.1. Pengambilan data dari sumber

Sumber data yang digunakan pada pembuatan pangkalan data ini berasal dari tiga sumber basis data yaitu, SIMAK, Mastertabel dan Pembayaran. Adapun data yang telah diambil dari basis data dapat dilihat pada Tabel 21.

Tabel 21. Sumber data

Sumber data	Tabel	Keterangan
SIMAK	STUDENT_COURSE_KRS	Berisi data mengenai mata kuliah yang diambil mahasiswa.
	MAHASISWA	Berisi tentang profil mahasiswa
	COURSE	Berisi macam-macam mata kuliah
	NILAI_HURUF	Berisi berbagai nilai beserta bobotnya dari setiap department
	TRANSKRIP	Berisi data nilai mahasiswa per mata kuliah
	STATUS_TRANSFER	Berisi data jenis-jenis status transfer mahasiswa
Mastertabel	FACULTY	Berisi data fakultas
	THAJARAN	Berisi data tahun ajaran
	class_program	Berisi data kelas program
	status_terdaftar	Berisi data jenis-jenis status terdaftar mahasiswa
	DEPARTMENT	Berisi tentang penjelasan department
	angkatan	Berisi data daftar angkatan.
	TERM	Berisi data daftar semester
	AGAMA	Berisi data berbagai jenis agama
Pembayaran	STUDENT_PAYMENT	Berisi data mengenai pembayaran mahasiswa pada setiap semester

4.4.2. Hasil Stage

Dalam arsitektur ODS + DDS, penyimpanan data pertama yang harus dilalui adalah *stage*. *Stage* bertujuan menyimpan data dari sumber agar data yang digunakan dalam pembuatan pangkalan data merupakan data yang konsisten. Data yang diambil kedalam *stage* merupakan data yang sesuai pada Tabel 21. Semua data yang ada pada *stage* berjumlah sama dengan yang ada pada sumber. Hal ini dikarenakan pada tahap ini tidak ada proses *cleansing* ataupun *transforming*. Adapun hasil dari proses pengambilan data ke dalam penyimpanan data *stage* ditampilkan pada Tabel 22.

Tabel 22. Hasil sumber data ke stage

Sumber data	Stage	Jumlah Data
STUDENT_COURSE_KRS	buffer_student_course_krs	2128744
MAHASISWA	buffer_mahasiswa	52732
COURSE	buffer_course	5725
NILAI_HURUF	buffer_nilai_huruf	240
TRANSKRIP	buffer_transkrip	1439846
STATUS_TRANSFER	buffer_status_transfer	5
FACULTY	buffer_faculty	11
THAJARAN	buffer_thajaran	16
class_program	buffer_class_program	3
status_terdaftar	buffer_status_terdaftar	7
DEPARTMENT	buffer_department	45
angkatan	buffer_angkatan	30
TERM	buffer_term	6
AGAMA	buffer_agama	5
STUDENT_PAYMENT	buffer_student_payment	1434472

4.4.3. Hasil Proses ETL ODS

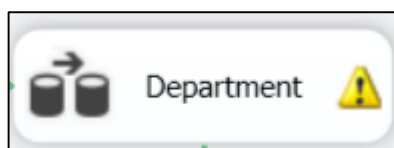
Seluruh tabel yang ada pada *stage* dimuat dalam penyimpanan data berikutnya, yaitu ODS. ODS merupakan penyimpanan data yang bersifat operasional dan data yang ada pada ODS dapat dimanipulasi oleh *end user*. Pemuatan data dari *stage* menuju ODS dengan menggunakan proses ETL. Dalam proses ETL data diekstrak, dibersihkan dari data kotor, ditransformasikan, ditambahkan informasi jika diperlukan dan setelah semua proses itu selesai data kemudian dimuat ke dalam ODS. Data operasional yang dimuat ke dalam ODS dimulai dari tahun ajaran 2008 hingga 2014. Hal ini dikarenakan pada tahun 2007 ke 2008 pihak universitas mengadakan perubahan penggunaan sistem informasi sehingga data dari tahun 2007 ke bawah berada pada sistem yang berbeda dan hanya sebagian yang dimuat ke sistem informasi sekarang. Selain itu banyak data yang mengandung *null*.

Penerapan proses ETL dilakukan dengan menggunakan salah satu *tool* yang ada pada SSDT yaitu *Integration Services* atau juga dikenal sebagai *SQL Server Integration Service* (SSIS). SSIS memberikan kemudahan untuk melakukan transformasi dan integrasi data dari berbagai sumber data. Dalam SSIS terdapat suatu komponen utama yaitu *package*. Di dalam *package* dapat berisi kumpulan *task* yang dapat dieksekusi berdasarkan urutan yang diinginkan. Di dalam *package* juga terdapat proses kontrol yang dapat mengatur alur dari *task* yaitu *control flow*. Adapun beberapa *task* yang ada pada *control flow* diantaranya adalah *data flow* dan

execute SQL. *Data flow* menerapkan bagaimana data diambil dari satu penyimpanan data ke penyimpanan data lainnya. *Execute SQL* merupakan *task* yang digunakan untuk menuliskan dan menjalankan sintak *query*. Sebagai contoh akan diuraikan bagaimana penerapan proses ETL menggunakan SSIS.

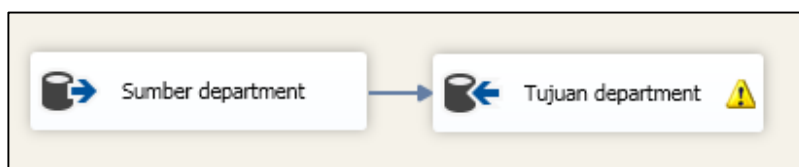
a. Tabel ods.department

Tahap pertama dalam proses kontrol adalah *control flow*. Seperti yang terlihat pada Gambar 12 bahwa terdapat suatu *control flow* yang memiliki *task* yaitu *data flow task* yang bernama department.



Gambar 12. *Control flow* ods.department

Pada *data flow task* department terdapat dua komponen yang dapat dilihat pada Gambar 13. Sumber department merupakan proses ekstrak data yang berasal dari tabel yang bernama *buffer_department* dan dapat dilihat pada Gambar 14. Tujuan department adalah proses dimana data yang telah diekstrak akan dimuat ke dalam tabel tujuan yaitu *ods.department* seperti pada Gambar 15.



Gambar 13. *Data flow* ods.department

Data access mode:
 Table or view

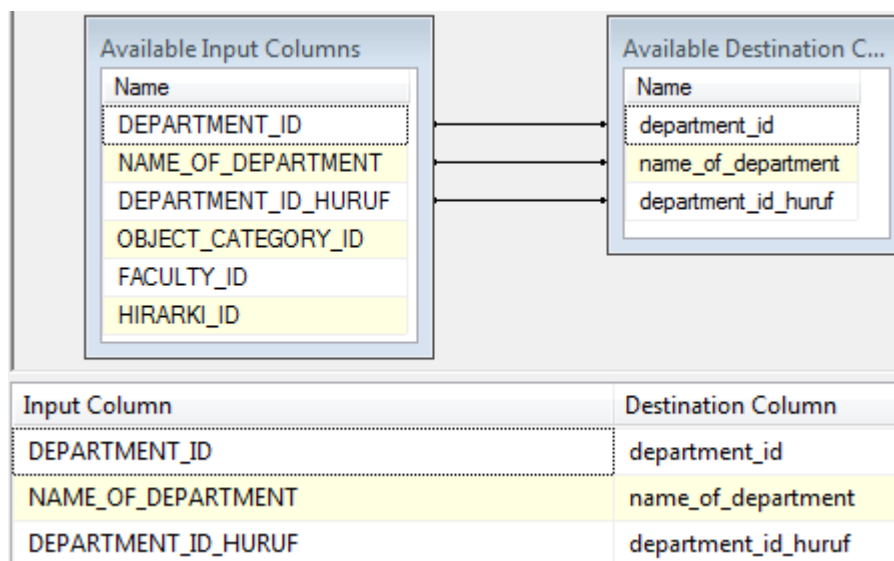
Name of the table or the view:
 [dbo].[buffer_department]

Gambar 14. Sumber department

Data access mode:
 Table or view - fast load

Name of the table or the view:
 [dbo].[ods.department]

Gambar 15. Tujuan department



Gambar 16. Mapping kolom tabel department

Gambar 16 merupakan tampilan proses *mapping* kolom apa saja yang digunakan pada tabel tujuan ods.department. Pada gambar terdapat input column dan destination column. Input column merupakan kolom-kolom yang tersedia pada tabel buffer_department dan destination column merupakan kolom pada tabel ods.department yang disediakan untuk menampung data sumber. Pada tabel

buffer_department tersedia enam kolom namun hanya tiga kolom yang digunakan di tabel ods.department. Pemilihan tiga kolom ini berdasarkan kebutuhan informasi dan kebutuhan untuk relasi tabel. Adapun hasil yang ada dari penerapan ETL dapat dilihat pada Gambar 17.

department_id	name_of_department	department_id_huruf
11	Teknik Sipil	TS
12	Teknik Elektro	TE
13	Teknik Mesin	TM
14	Teknologi Informasi	TI
15	TEKNIK ---	TEP
16	TEKNIK ---	TMP
21	Agroteknologi	BP
22	Agrobisnis	SE
31	Pendidikan Dokter	KU

Gambar 17. Hasil sebagian ods.department

b. Tabel ods.status_mahasiswa

Tabel ods.status_mahasiswa tidak berasal dari sumber sistem melainkan dibuat sebagai tabel tambahan. Tabel ini memiliki tiga nilai yang menyatakan status mahasiswa yaitu aktif, baru dan bayar. Status yang bernilai aktif dan baru dapat digunakan untuk kebutuhan informasi seperti jumlah mahasiswa baru dan jumlah total mahasiswa. Status bayar untuk menemukan mahasiswa yang telah membayar namun tidak mengambil mata kuliah. Pengisian data pada tabel ods.status_mahasiswa menggunakan *execute SQL task* yang didalamnya sudah di tuliskan *query* yang memberikan perintah untuk melakukan pengisian data. Adapun

execute SQL task dan *query* yang dimaksud dapat dilihat pada Gambar 18 dan Gambar 19.



Gambar 18. *Execute SQL task* status_mahasiswa

```
INSERT INTO [ods.status_mahasiswa] VALUES(1, 'Baru');
INSERT INTO [ods.status_mahasiswa] VALUES(2, 'Aktif');
INSERT INTO [ods.status_mahasiswa] VALUES(3, 'Bayar');
```

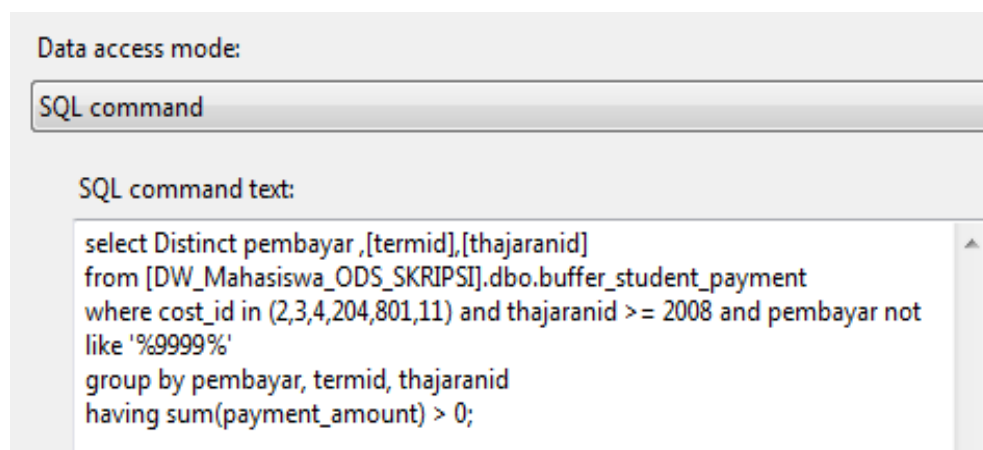
Gambar 19. *Query* menambahkan data

c. Tabel ods.student_payment

Data flow tabel ods.student_payment terdapat dua *task* seperti yang ditunjukkan pada Gambar 20. Sumber payment memiliki fungsi untuk mengekstrak data dari tabel buffer_student_payment. Data yang diambil tidak secara keseluruhan melainkan disaring berdasarkan beberapa ketentuan. Adapun pemilihan data berdasarkan beberapa ketentuan dapat dilihat pada Gambar 21.



Gambar 20. *Data flow* ods.student_payment



Gambar 21. *Query* pengambilan data dari `buffer_student_payment`.

Seperti yang terlihat pada gambar ada beberapa kondisi yang menjadi ketentuan, yaitu `cost_id`, `thajaranid`, dan `pembayar`. Penentuan nilai dari `cost_id` merupakan ketentuan dari pihak universitas. Ketentuan kolom `pembayar` tidak mengandung nilai '%9999%' dikarenakan data ini merupakan data percobaan sistem informasi oleh pihak universitas. Jumlah data dari `buffer_student_payment` sebanyak 1434472, setelah dilakukan proses pemilihan data berdasarkan ketentuan tersebut didapatkan jumlah data sebanyak 150386. Proses ETL hanya mengambil tiga kolom yaitu `pembayar`, `thajaranid`, dan `termid`. Pemilihan tiga kolom ini didasari dari kebutuhan informasi yaitu mahasiswa aktif setiap semester. Proses ekstrak data dari tabel `buffer_student_payment` selanjutnya dimuat ke dalam tabel `ods.student_payment` seperti yang ditampilkan pada Gambar 22. Adapun proses *mapping* kolom dan hasil data dapat dilihat pada Gambar 23 dan Gambar 24.

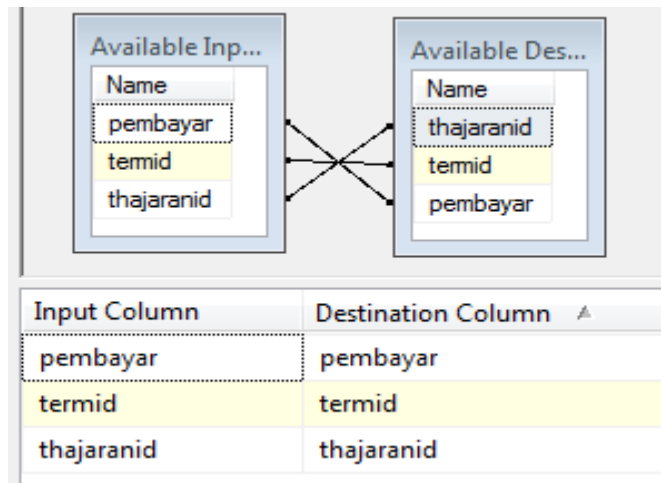
Data access mode:

Table or view - fast load

Name of the table or the view:

[dbo].[ods.student_payment]

Gambar 22. Tujuan tabel ods.student_payment



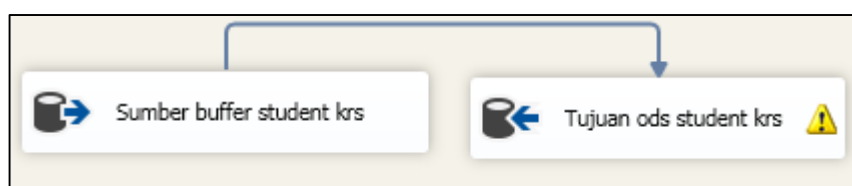
Gambar 23. Mapping kolom tabel status_mahasiswa

thajaranid	termid	pembayar
2008	1	0600110004
2008	2	0600110004
2009	1	0600110004
2009	2	0600110004
2010	1	0600110004
2008	1	0600110008
2008	2	0600110008
2009	1	0600110008
2009	2	0600110008
2010	1	0600110008
2008	1	0600110012
2008	2	0600110012

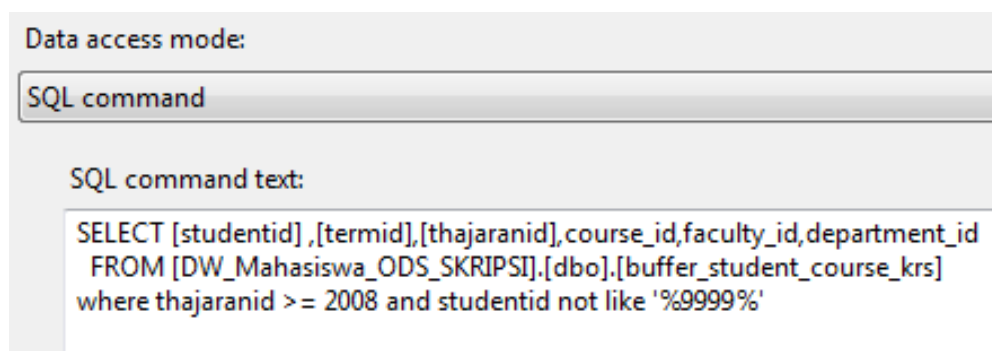
Gambar 24. Hasil sebagian tabel ods.student_payment

d. Tabel ods.student_course_krs

Data flow tabel ods.student_course krs dapat dilihat pada Gambar 25. Pada bagian sumber buffer student krs data diekstrak dan diambil dari tabel buffer_student_course_krs. Pemilihan data yang telah diekstrak berdasarkan beberapa ketentuan. Adapun penerapan pengambilan data dapat dilihat pada Gambar 26.



Gambar 25. Data flow ods.student_course_krs



Gambar 26. Sumber buffer_student_course_krs

Pada Gambar 26 dapat dilihat terdapat *query* pemilihan data dari tabel buffer_student_course_krs dengan beberapa ketentuan. Jumlah data dari tabel buffer_student_course_krs sebesar 2128744. Setelah dilakukan proses pemilihan berdasarkan ketentuan jumlah data menjadi 1033974. Dari proses tersebut terdapat pemilihan enam kolom yaitu studentid, termid, thajaranid, course_id, faculty_id

dan `department_id`. Enam kolom ini digunakan untuk mendapatkan data mata kuliah yang diambil oleh mahasiswa setiap semester. Proses pengambilan data dari tabel `buffer_student_course_krs` dimuat ke dalam tabel `ods.student_course_krs` seperti yang ditunjukkan pada Gambar 27. Adapun *mapping* dan hasil sebagian dari proses ETL tabel `ods.student_course_krs` ditunjukkan pada Gambar 28 dan Gambar 29.

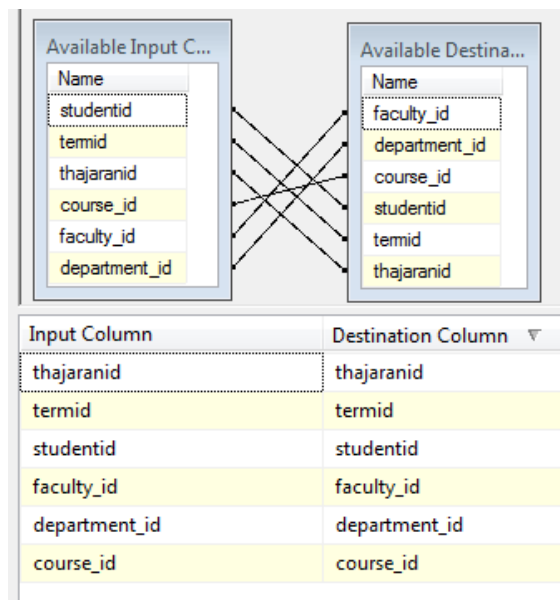
Data access mode:

Table or view - fast load

Name of the table or the view:

[dbo].[ods.student_course_krs]

Gambar 27. Tujuan `ods.student_course_krs`



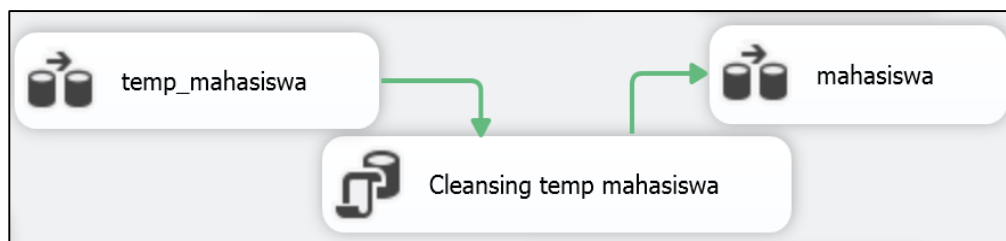
Gambar 28. *Mapping* tabel `student_course_krs`

studentid	temid	thajaranid	course_id	faculty_id	department_id
1997120053	2	2011	MKB-384	1	12
1997120053	2	2011	MKK-124	1	12
1997120053	2	2011	MPK-112	1	12
1997120053	2	2011	MPK-116	1	12
1997120053	2	2011	MPK-216	1	12
1997120053	2	2012	MKB-224	1	12
1997120053	2	2012	MKB-282	1	12
1997120053	2	2012	MKB-316	1	12
1997120053	2	2012	MKB-382	1	12
1997120053	2	2012	MKB-384	1	12
1997120053	2	2012	MKB-524	1	12
1997120053	2	2012	MKK-124	1	12

Gambar 29. Hasil sebagian tabel ods.student_course_krs

e. Tabel ods.mahasiswa

Control flow proses ETL mahasiswa dapat dilihat pada Gambar 30. Pada gambar terlihat ada tiga *task* yaitu temp_mahasiswa, cleansing temp_mahasiswa, dan mahasiswa.



Gambar 30. Control flow mahasiswa

```

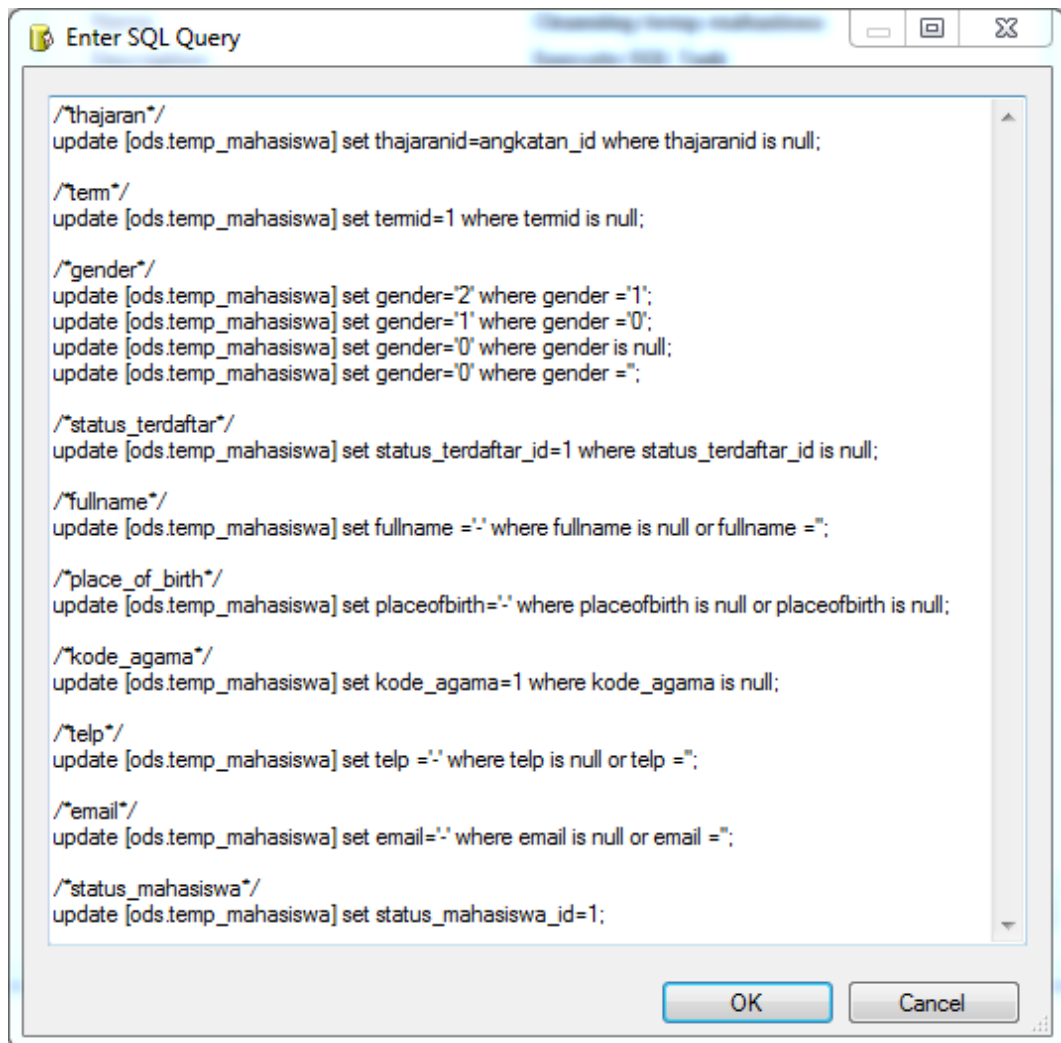
Data access mode:
SQL command
SQL command text:
SELECT *
FROM [DW_Mahasiswa_ODS_SKRIPSI].[dbo].[buffer_mahasiswa] where
STUDENTID not like '%9999%' and ANGKATAN_ID != '1982'
  
```

Gambar 31. *Query* pengambilan data tabel buffer_mahasiswa.

Temp_mahasiswa merupakan *task* yang berisi proses ekstrak data dari tabel *buffer_mahasiswa* menuju tabel *ods.temp_mahasiswa*. Tabel ini merupakan tabel yang dibuat untuk penampungan sementara data mahasiswa. Dalam proses ini pengambilan data didasari beberapa ketentuan yang ditunjukkan pada Gambar 31. Dari beberapa ketentuan tersebut jumlah data yang berasal dari *buffer_mahasiswa* sebesar 52732 menjadi 52710. Tujuan dibuatnya tabel ini untuk menjaga data pada tabel *buffer_mahasiswa* tetap konsisten dikarenakan pada tahap *cleansing* dibutuhkan manipulasi data. Proses *cleansing* data pada tabel *ods.temp_mahasiswa* dilakukan dengan menggunakan *execute SQL task* dan bertujuan untuk membersihkan data yang bernilai *null*. Beberapa kolom yang telah dibersihkan seperti *thajaranid*, *termid*, *gender*, *status_terdaftar_id*, *fullname*, *placeofbirth*, *kode_agama*, *telp*, *email*, dan *status_mahasiswa*. Kolom *status_mahasiswa* yang ada merupakan kolom tambahan yang menjelaskan status mahasiswa. Adapun *cleansing* yang telah dilakukan dapat dilihat pada Gambar 32. Adapun nilai yang digunakan untuk merubah beberapa kolom ditunjukkan pada Tabel 23.

Tabel 23. Merubah nilai *Null*

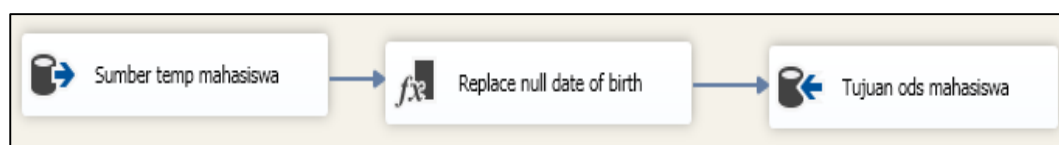
Kolom	Keterangan
Kode_agama	Diganti dengan '1' yang bernilai islam
Status_terdaftar	Diganti dengan '1' yang berarti mahasiswa baru
Thajaranid	Diganti dengan nilai dari <i>angkatan_id</i>
Termid	Diganti dengan nilai '1' yang berarti gasal



Gambar 32. Cleansing data mahasiswa

Data pada tabel `ods.temp_mahasiswa` yang telah dibersihkan kemudian menjadi sumber data dan data tersebut diekstrak serta dimuat ke dalam tabel tujuan yaitu tabel `ods.mahasiswa`. Dalam proses ETL dari tabel `ods.temp_mahasiswa` menuju tabel `ods.mahasiswa` terdapat proses *cleansing* dengan menggunakan *derived column* yang tersedia pada *data flow*. Proses *cleansing* ini data *null* pada

kolom dateofbirth dibersihkan dengan diganti nilai '1899-12-30 00:00:00.000'. Adapun proses *data flow task* dan proses *cleansing* data dilihat pada Gambar 33 dan Gambar 34.



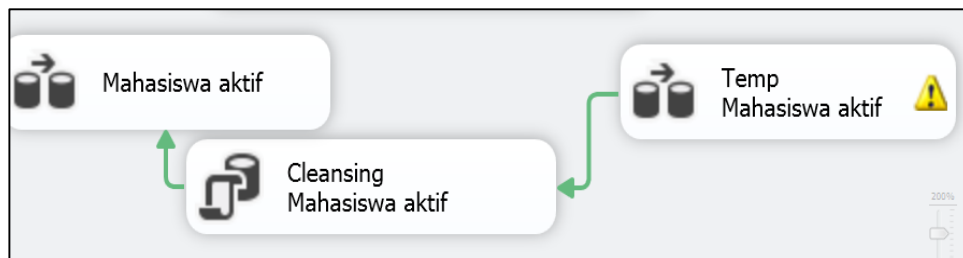
Gambar 33. *Data flow task* mahasiswa

Derived Column Name	Derived Column	Expression	Data Type
dateofbirth	Replace 'dateofbirth'	REPLACENULL(dateofbirth,0)	database timestamp [DT_...

Gambar 34. *Cleansing* data kolom dateofbirth

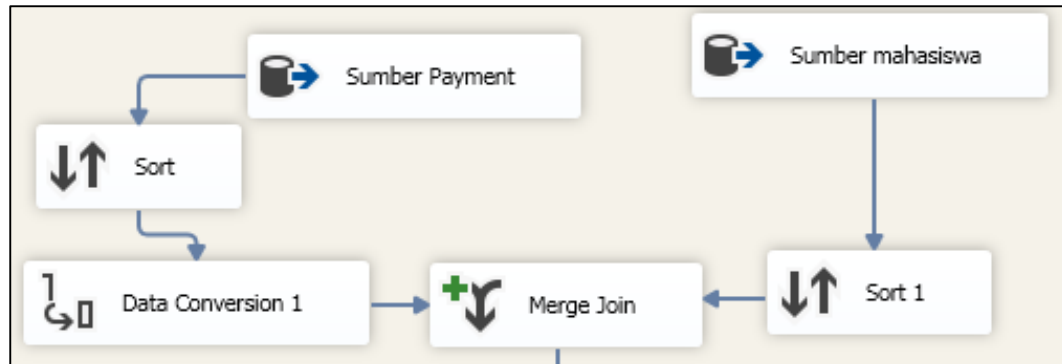
f. Tabel ods.mahasiswa_aktif

Tabel ods.mahasiswa_aktif merupakan tabel yang terbentuk dari beberapa gabungan tabel meliputi tabel ods.student_payment, tabel ods.student_course_krs, dan tabel ods.mahasiswa. sistem informasi akademik saat ini belum mempunyai data yang menjelaskan status dari mahasiswa dari setiap semester. Berdasarkan hal inilah tabel ods.mahasiswa_aktif dibuat dengan tujuan untuk mengetahui status dari setiap mahasiswa.



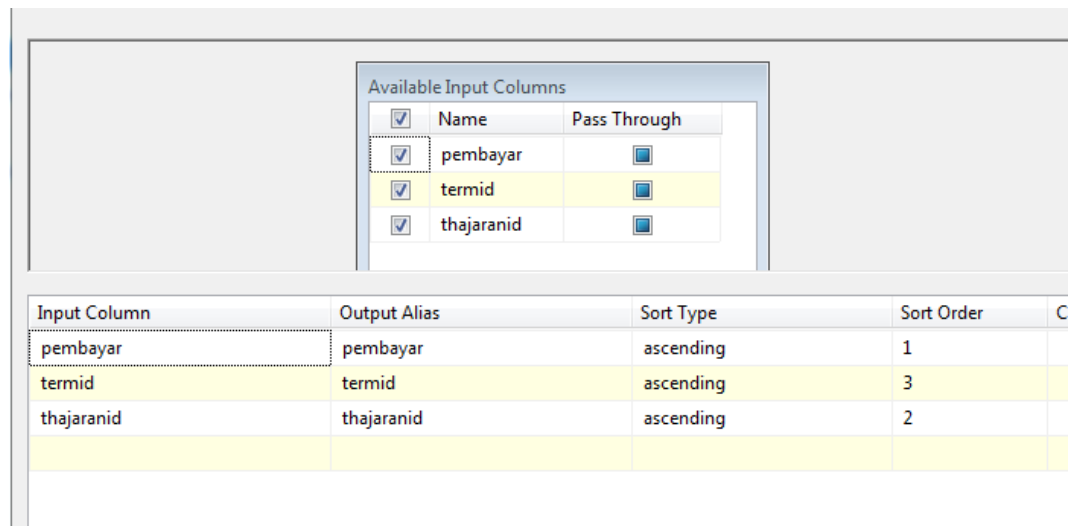
Gambar 35. *Control flow task* mahasiswa aktif

Control flow task mahasiswa aktif dapat dilihat pada Gambar 35. *Control flow* dimulai dari kanan yaitu temp mahasiswa aktif. Temp mahasiswa aktif merupakan tabel sementara untuk penampungan hasil dari penggabungan ketiga tabel yang telah disebutkan sebelumnya. Penggabungan tabel dilakukan sebanyak dua tahap, pada tahap pertama dilakukan penggabungan data antara tabel `ods.student_payment` dan tabel `ods.mahasiswa` yang dapat dilihat pada Gambar 36.



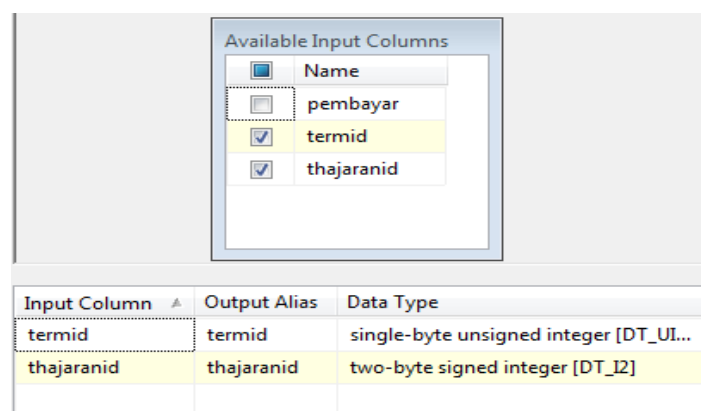
Gambar 36. `ods.student_payment` dan `ods.mahasiswa`

Pada Gambar 36 terdapat beberapa *task* seperti *sort*, *data conversion* dan *merge join*. *Sort* memiliki fungsi untuk mengurutkan data berdasarkan kolom tertentu. Adapun penerapan *Sort* ditampilkan pada Gambar 37 dengan didasari tiga kolom yaitu `pembayar`, `thajaranid`, `termid`.



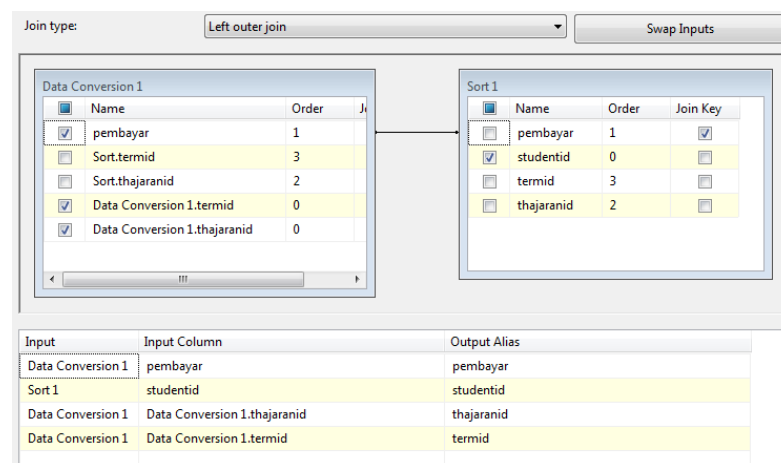
Gambar 37. *Sort* data berdasarkan kolom pembayar,thajaranid,termid

Data Conversion memiliki fungsi untuk mengubah tipe data dari suatu kolom. Penggunaan tipe data sangatlah mempengaruhi dalam penggabungan data dikarenakan ketika terdapat dua atau lebih tabel yang ingin digabungkan maka pada setiap tabel harus memiliki kolom yang bernilai dan bertipe data yang sama. Adapun penerapan *data conversion* dapat dilihat pada Gambar 38 dengan mengubah tipe data kolom thajaranid dan termid.



Gambar 38. Data conversion pengubahan tipe data

Merge join memiliki fungsi melakukan penggabungan data dengan tipe penggabungan seperti LEFT JOIN, RIGHT JOIN, dan FULL JOIN. Adapun penerapan *merge join* dapat dilihat pada Gambar 39 dengan menggunakan LEFT JOIN serta kolom pembayar sebagai penghubung antar tabel. Penggabungan tabel ods.student_payment dengan tabel ods.mahasiswa bertujuan untuk mendapatkan kolom studentid dari tabel ods.mahasiswa dan ditambahkan ke tabel ods.student_payment.

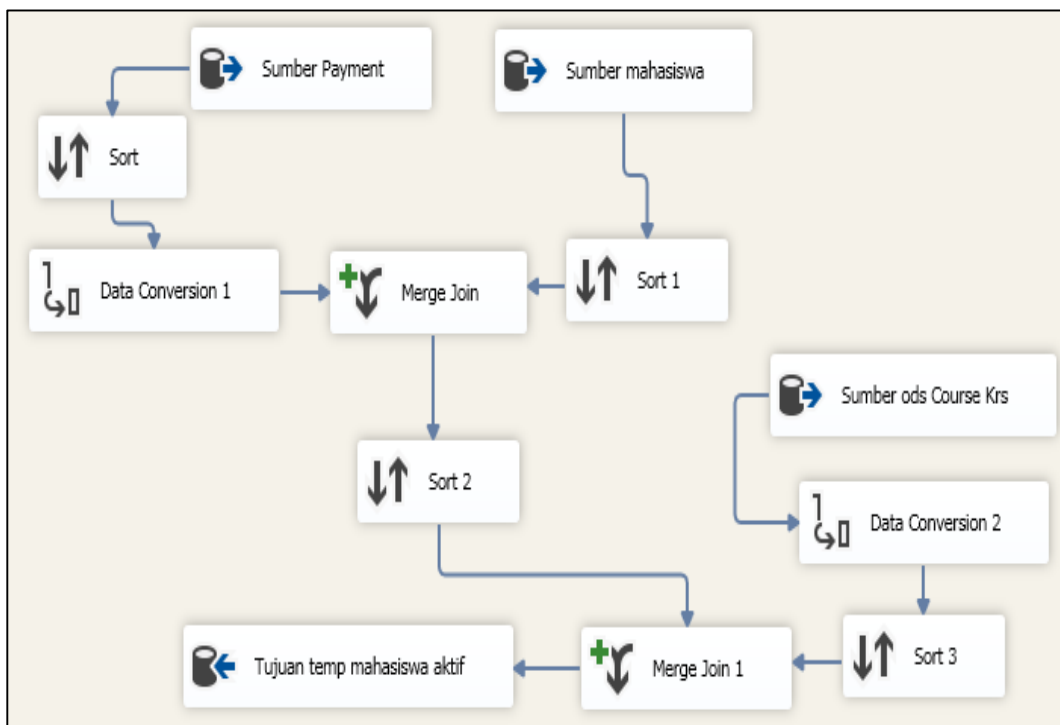


Gambar 39. *Merge join* dengan tipe penggabungan LEFT JOIN

Dari hasil penggabungan ods.student_payment dan ods.mahasiswa didapatkan jumlah data sebesar 150386 data. Dari jumlah data tersebut ditemukan 2713 data pembayar yang tidak memiliki studentid. Hasil data dari penggabungan tabel tahap pertama digabungkan dengan tabel ods.student_krs. Penggabungan ini dilakukan untuk menemukan data mahasiswa yang telah melakukan pembayaran dan yang telah mengambil mata kuliah. Penggabungan dilakukan dengan tipe penggabungan FULL JOIN. Dari hasil penggabungan ini ditemukan tiga kondisi :

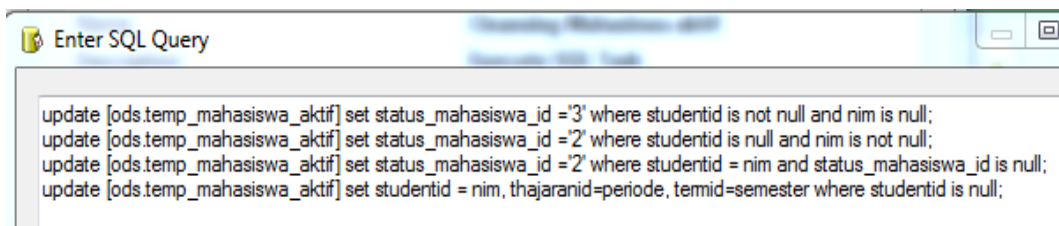
1. Mahasiswa membayar tapi tidak mengambil mata kuliah
2. Mahasiswa mengambil mata kuliah tapi tidak terdaftar di pembayaran.
3. Mahasiswa yang telah membayar dan juga mengambil mata kuliah.

Berdasarkan tiga kondisi tersebut juga dapat ditentukan status dari mahasiswa pada tiap semester. Pengisian status_mahasiswa ditampilkan pada Gambar 41. Hasil pada tahap akhir kemudian dimuat ke dalam tabel ods.temp_mahasiswa_aktif untuk berikutnya di *cleansing* terlebih dahulu sebelum dimuat ke dalam tabel ods.mahasiswa aktif. Adapun *data flow* tahap akhir penggabungan data dan keseluruhan penggabungan data ditampilkan pada Gambar 40.



Gambar 40. *Data flow task* ods.temp_mahasiswa_aktif

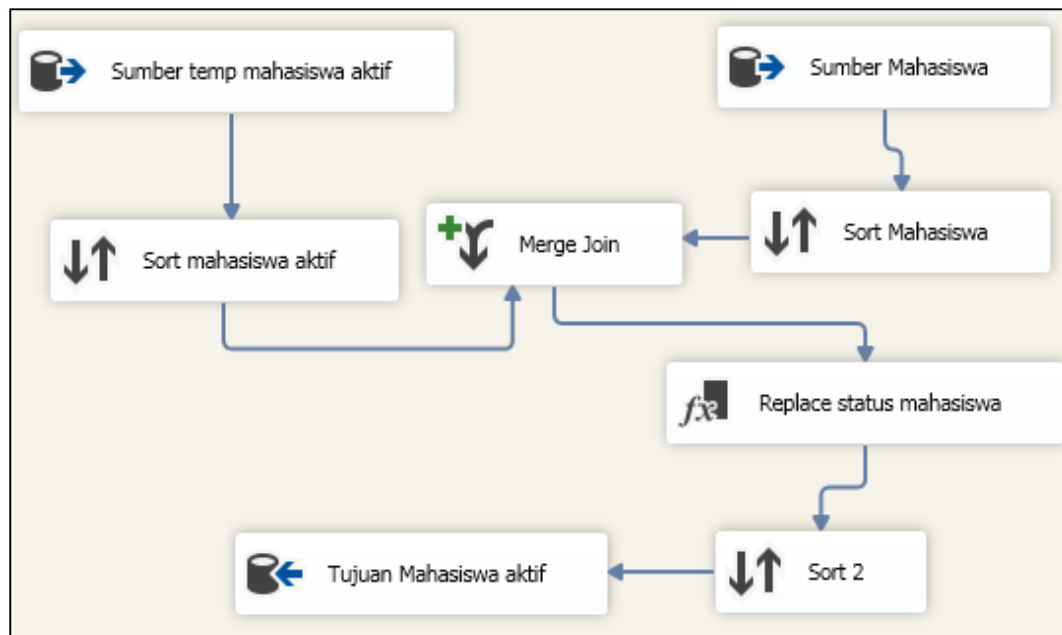
Data yang sudah dimuat ke dalam table `ods.temp_mahasiswa_aktif` selanjutnya di *cleansing* dahulu untuk memastikan data bersih. Dalam proses *cleansing* data *null* pada kolom status mahasiswa diisi dengan dua nilai berdasarkan nilai yang ada pada tabel `ods.status_mahasiswa` yaitu nilai dua, dan tiga. Nilai dua sebagai mahasiswa aktif dan nilai tiga sebagai mahasiswa yang sudah membayar sedangkan nilai satu sebagai mahasiswa baru yang akan ditambahkan pada proses ETL berikutnya. Dalam menentukan pengisian kolom status mahasiswa dengan nilai dua dan tiga tersebut dapat dilihat berdasarkan tiga kondisi yang sudah diuraikan sebelumnya. Adapun penerapan dari pengisian kolom status mahasiswa pada Gambar 41.



```
Enter SQL Query
update [ods.temp_mahasiswa_aktif] set status_mahasiswa_id = '3' where studentid is not null and nim is null;
update [ods.temp_mahasiswa_aktif] set status_mahasiswa_id = '2' where studentid is null and nim is not null;
update [ods.temp_mahasiswa_aktif] set status_mahasiswa_id = '2' where studentid = nim and status_mahasiswa_id is null;
update [ods.temp_mahasiswa_aktif] set studentid = nim, thajaranid=periode, temid=semester where studentid is null;
```

Gambar 41. Pengisian kolom status mahasiswa

Sesuai dengan Gambar 35 proses selanjutnya setelah melakukan *cleansing* adalah proses ekstrak data dari tabel `ods.temp_mahasiswa_aktif` menuju tabel `ods.mahasiswa_aktif`. Adapun proses ekstrak tersebut ditampilkan pada Gambar 42.



Gambar 42. Penggabungan mahasiswa aktif dan mahasiswa

Gambar 42 menjelaskan alur data dari tabel `ods.temp_mahasiswa_aktif` menuju tabel `ods.mahasiswa_aktif`. Dalam proses ini terdapat penggabungan tabel antara tabel `ods.temp_mahasiswa_aktif` dan tabel `ods.mahasiswa`. Penggabungan ini menggunakan LEFT JOIN yang mana tabel `ods.temp_mahasiswa_aktif` sebagai tabel disisi kiri. Kolom yang digunakan sebagai penghubung antar kedua tabel tersebut adalah `studentid`, `thajaranid`, dan `termid`. Penggabungan kedua tabel ini bertujuan untuk mendapatkan kolom `status_mahasiswa_id` dengan nilai satu. Berikut penerapan LEFT JOIN ditunjukkan pada Gambar 43. Tabel yang digabungkan kemudian di *update* terlebih dahulu untuk memasukkan nilai pada kolom `status_mahasiswa_id` di tabel `ods.mahasiswa_aktif`.

Join type: Left outer join Swap Inputs

Sort mahasiswa aktif	Name	Order	Join Ke
<input checked="" type="checkbox"/>	studentid	1	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	termid	3	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	thajaranid	2	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	status_mahasiswa_id	0	<input type="checkbox"/>

Sort Mahasiswa	Name	Order	Join Ke
<input checked="" type="checkbox"/>	studentid	1	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	status_mahasiswa_id	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	termid	3	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	thajaranid	2	<input checked="" type="checkbox"/>

Input	Input Column	Output Alias
Sort Mahasiswa	studentid	studentid (1)
Sort Mahasiswa	thajaranid	thajaranid (1)
Sort Mahasiswa	termid	termid (1)
Sort mahasiswa aktif	studentid	studentid
Sort mahasiswa aktif	termid	termid
Sort mahasiswa aktif	thajaranid	thajaranid
Sort Mahasiswa	status_mahasiswa_id	status_mahasiswa_id (1)
Sort mahasiswa aktif	status_mahasiswa_id	status_mahasiswa_id

Gambar 43. Merge join mahasiswa aktif dan mahasiswa

- Variables and Parameters
- Columns

- Mathematical Functions
- String Functions
- Date/Time Functions
- NULL Functions
- Text Constants

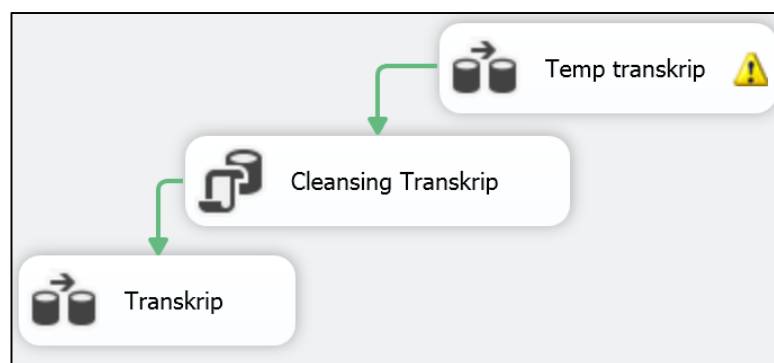
Description:

Derived Column Name	Derived Column	Expression	Data Type
status_mahasiswa_id	Replace 'status_mahasiswa...	[status_mahasiswa_id (1)]	single-byte unsigned inte...

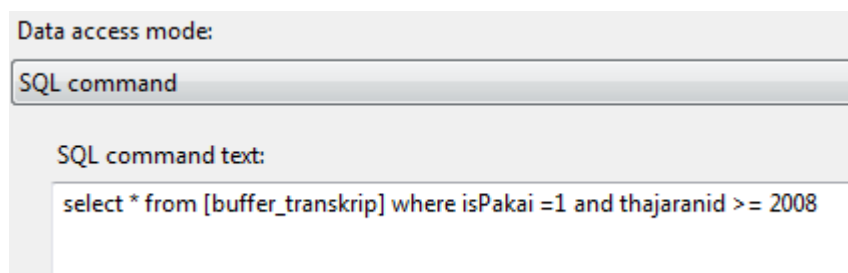
Gambar 44. Update kolom status_mahasiswa

g. Tabel ods.Transkrip

Tabel ods.transkrip memiliki nilai dari setiap mata kuliah. Tabel ini juga dapat digunakan untuk mendapatkan nilai IP Data yang dimuat pada tabel didasari beberapa ketentuan. Dalam proses ETL terdapat tiga *control flow* yang ditunjukkan pada Gambar 45. Adapun ketentuan dalam pengambilan data ditunjukkan pada Gambar 46.



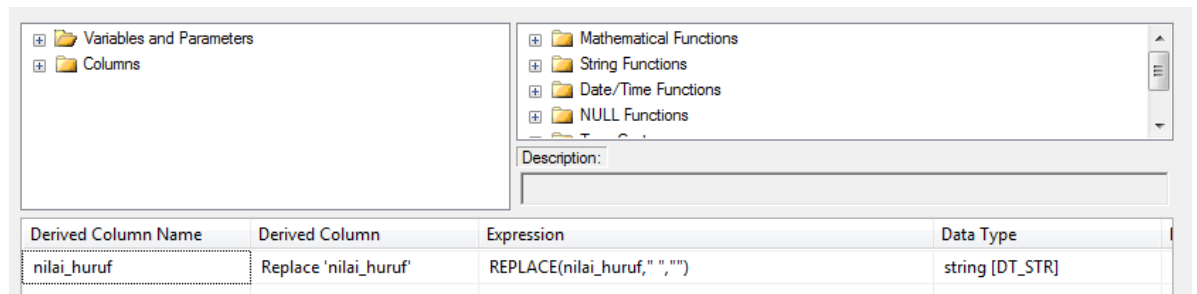
Gambar 45. *Control Flow* ods.transkrip



Gambar 46. *Query* pengambilan data dari buffer_transkrip

Data dari sumber tidak langsung dimuat ke ods.transkrip namun ditampung terlebih dahulu di tabel sementara temp.transkrip untuk di lakukan *cleansing*. Adapun *cleansing* yang dilakukan adalah menghapus spasi pada kolom nilai_huruf. Hal ini dilakukan agar menghindari konflik saat relasi tabel

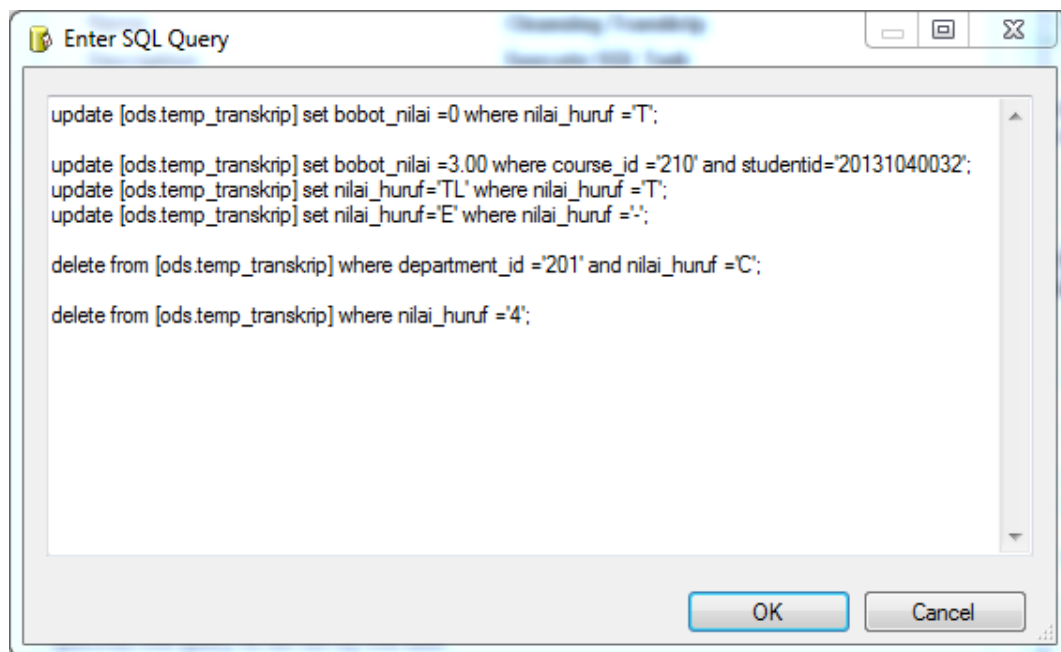
dikarenakan terdapat nilai_huruf yang tidak cocok pada master tabel ods.nilai_huruf. Adapun proses *cleansing* ditunjukkan Gambar 47. Selain melakukan *cleansing* pada kolom nilai_huruf terdapat juga *cleansing* terhadap hal lainnya yang ditampilkan pada Gambar 48.



The screenshot shows a software interface for defining a derived column. On the left, there are folders for 'Variables and Parameters' and 'Columns'. On the right, there are folders for 'Mathematical Functions', 'String Functions', 'Date/Time Functions', and 'NULL Functions'. Below these is a 'Description:' field. At the bottom, a table lists the derived column configuration:

Derived Column Name	Derived Column	Expression	Data Type
nilai_huruf	Replace 'nilai_huruf'	REPLACE(nilai_huruf, " ", "")	string [DT_STR]

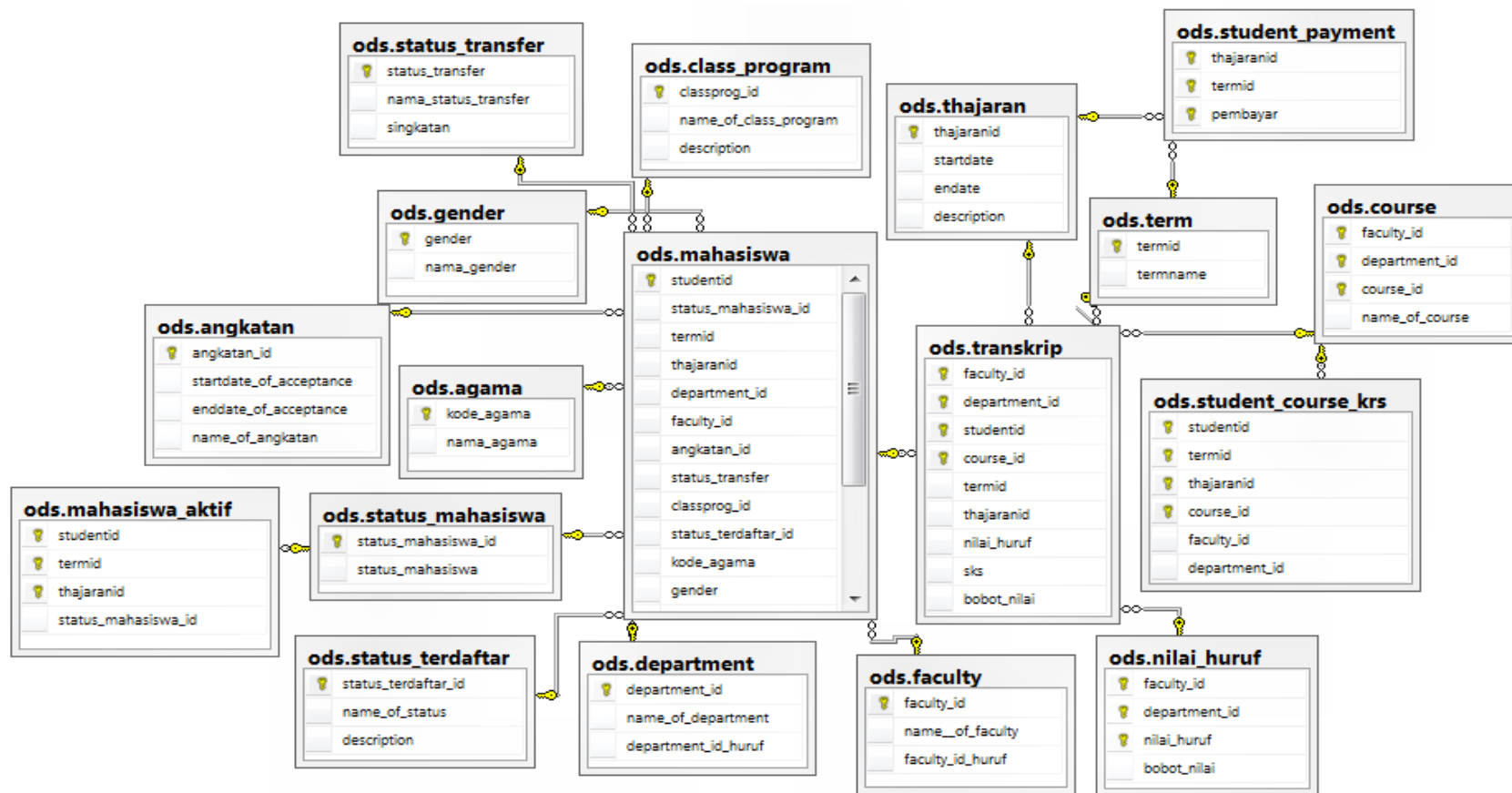
Gambar 47. *Replace* kolom nilai_huruf yang memiliki spasi



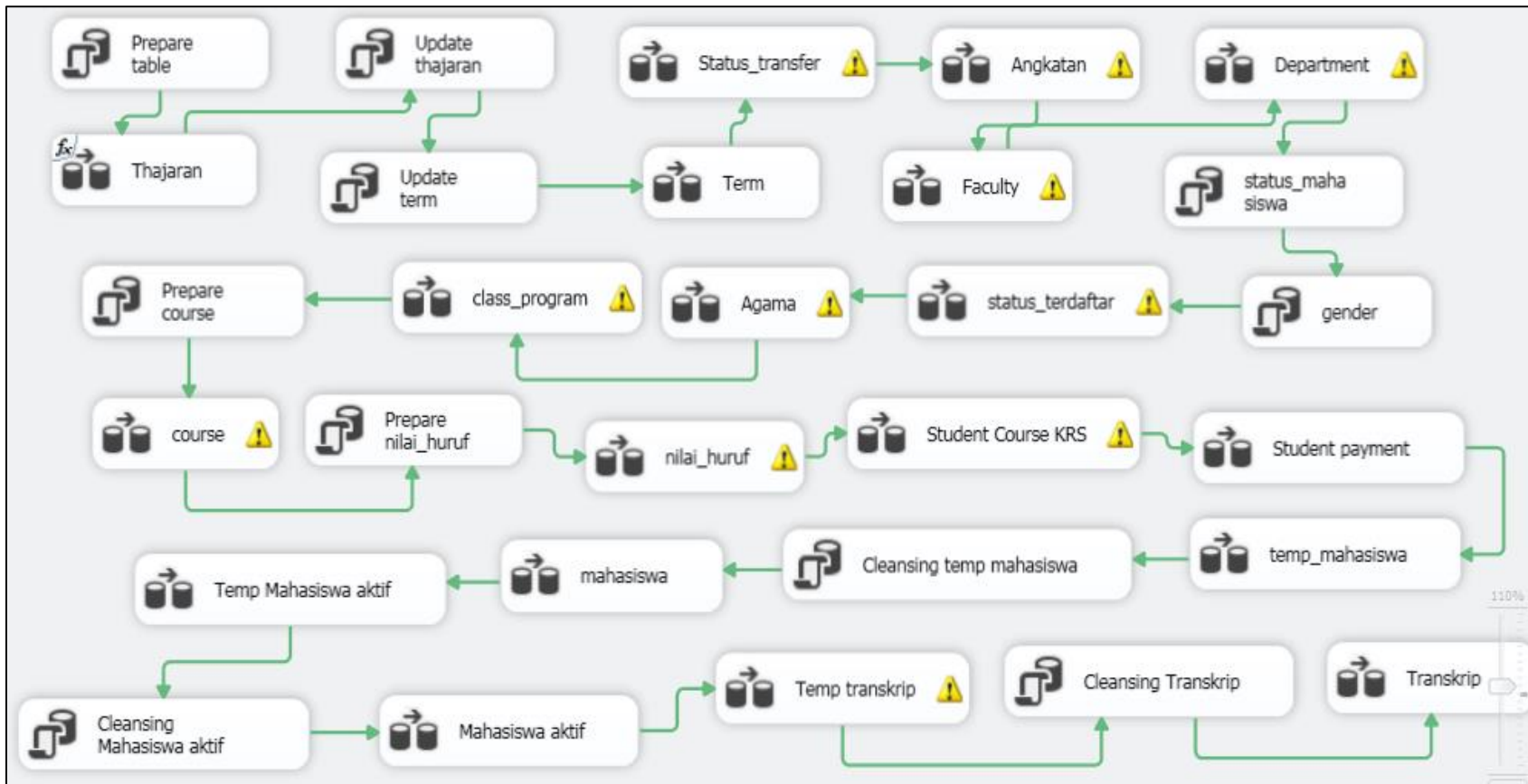
Gambar 48. *Cleansing data transkrip*

Adapun hasil dari tabel lainnya dalam proses ETL ODS telah diuraikan pada lampiran. Tabel lainnya meliputi tabel ods.angkatan, ods.faculty, ods.agama,

ods.status_mahasiswa, ods.status_terdaftar, ods.status_transfer ods.class_program, ods.nilai_huruf, ods.gender, ods.thajaran, ods.term, ods.transkrip dan ods.course. berikut gambar hasil ODS dan *control flow* secara keseluruhan yang telah dibuat dapat dilihat pada Gambar 49 dan Gambar 50.



Gambar 49. Diagram ODS



Gambar 50. Control flow ODS

4.4.4. Hasil Proses ETL DDS

Hasil dari ODS diproses kembali untuk dimuat ke dalam penyimpanan data berikutnya yaitu DDS. Dalam DDS, hasil ODS akan diubah bentuk menjadi *star schema*. *Star schema* pangkalan data ini memiliki dua belas tabel *dimension* dan satu tabel *fact*. Setiap tabel *dimension* memiliki *surrogate key* yang memiliki nilai *auto increment* dengan *seed* dimulai dari angka satu.

a. Tabel dim_department

Dim_department merupakan tabel *dimension* yang berisi berbagai data mengenai department yang ada pada universitas. Tabel ini memiliki *surrogate key* pada kolom department_key. Dalam proses ETL tabel dim_department memiliki sumber data dari tabel ods.department. Proses ETL pada dim_department menggunakan dua *task* yaitu sumber ods department dan tujuan dim department. Penerapan proses ETL pada dim_department hampir sama seperti penerapan pada ods.department. Perbedaan yang ada hanya pada penambahan kolom yang dijadikan *surrogate key* dan juga *primary key* serta memiliki nilai *auto increment*. Nilai *auto increment* dibuat saat tabel pertama kali dibuat. Adapun penggunaan *auto increment* ditampilkan pada Gambar 51.

```
CREATE TABLE [dim_department] (
  department_key SMALLINT NOT NULL IDENTITY(0,1),
  department_id SMALLINT NOT NULL,
  name_of_department VARCHAR(38) NULL,
  department_id_huruf VARCHAR(4) NULL,
  PRIMARY KEY(department_key)
);
```

Gambar 51. *Query* membuat auto increment dim_department

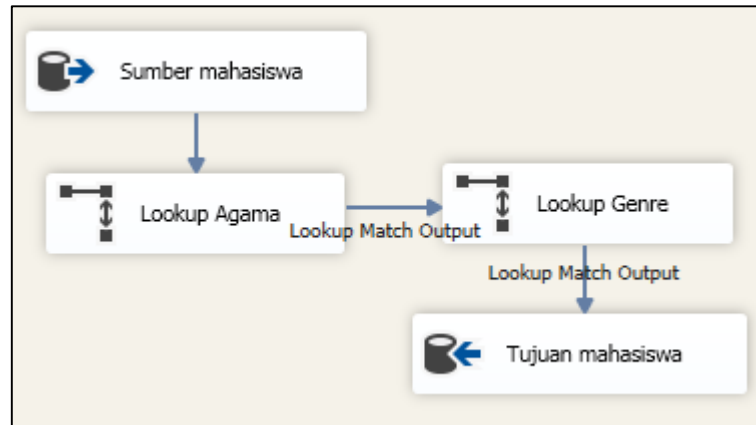
department_key	department_id	name_of_department	department_id_huruf
0	11	Teknik Sipil	TS
1	12	Teknik Elektro	TE
2	13	Teknik Mesin	TM
3	14	Teknologi Informasi	TI
4	15	TEKNIK ---	TEP
5	16	TEKNIK ---	TMP
6	21	Agroteknologi	BP
7	22	Agrobisnis	SE
8	31	Pendidikan Dokter	KU
9	32	Ilmu Keperawatan	IK
10	34	Kedokteran Gigi	KG
11	35	Farmasi	FA

Gambar 52. Hasil sebagian tabel dim_department

Adapun sebagian hasil dari tabel dim_department pada Gambar 52. Seperti yang terlihat bahwa pada kolom department_key memiliki nilai yang berurutan dan di mulai dari angka nol.

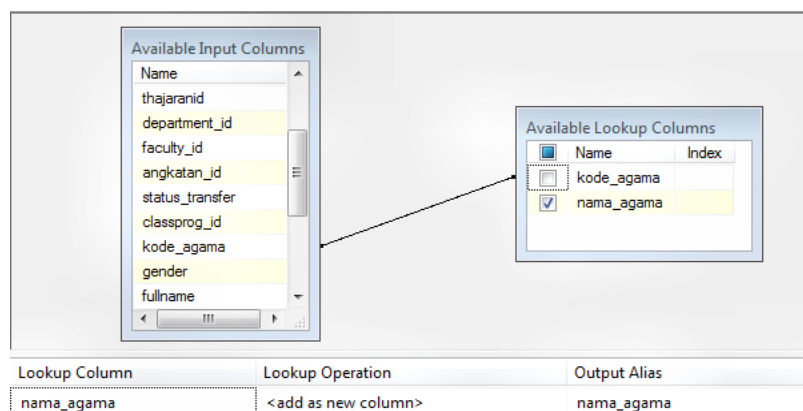
b. Tabel dim_mahasiswa

Dalam proses pembuatan tabel dim_mahasiswa terdapat penggabungan dengan dua tabel, yaitu tabel ods.agama dan tabel ods.gender. Kedua tabel tersebut memiliki kolom yang ditambahkan pada tabel dim_mahasiswa. Penggabungan dilakukan dalam dua *task* berbeda. Pertama menggabungkan dim_mahasiswa dan ods.agama. kedua menggabungkan dim_mahasiswa dengan ods.gender. Adapun hasil yang dimaksud dari penjelasan sebelumnya ditampilkan pada Gambar 53.



Gambar 53. *Data flow* dim_mahasiswa

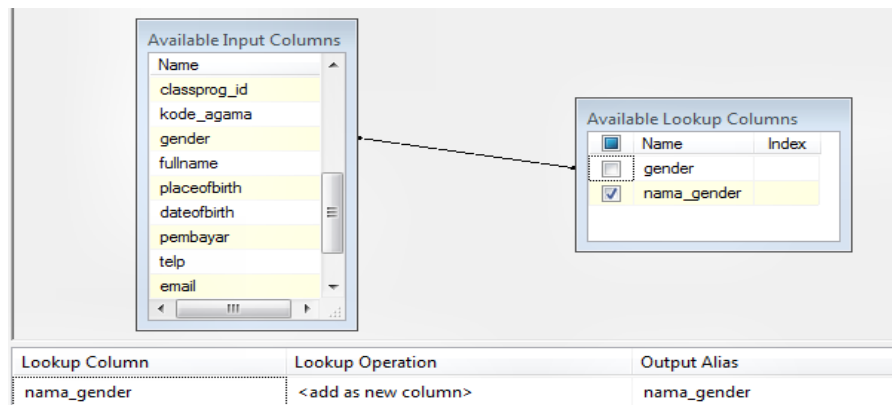
Pada Gambar 53 terlihat ada dua *task*. *Task* pertama adalah lookup agama. Lookup memiliki fungsi serupa seperti JOIN yang dapat menggabungkan dua tabel yang berbeda. Lookup agama digunakan untuk mendapatkan kolom nama_agama dari tabel ods.agama menuju tabel dim.mahasiswa. Hubungan antara tabel ods.agama dan dim_mahasiswa melalui kolom kode_agama. Adapun penerapan dari lookup agama dapat dilihat pada Gambar 54.



Gambar 54. Lookup agama

Lookup gender digunakan untuk mendapatkan kolom nama_gender dari tabel ods.gender menuju tabel dim_mahasiswa. kolom yang digunakan sebagai

penghubung kedua tabel adalah kolom gender. Berikut penerapan lookup gender ditunjukkan pada Gambar 55 dan hasil sebagian tabel dim_mahasiswa pada Gambar 68.



Gambar 55. Lookup gender

c. Tabel fact_perkuliahan

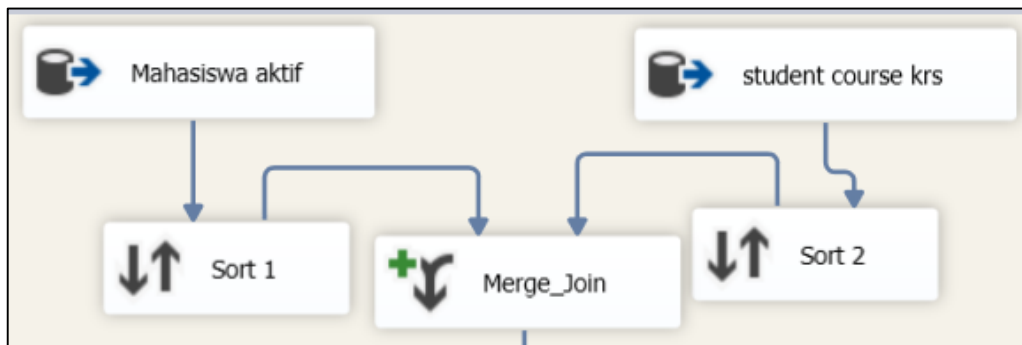
Control flow dari tabel fact_perkuliahan ditampilkan pada Gambar 56. Pada Gambar 56 terdapat dua jenis *task* yaitu temp fact perkuliahan dan fact perkuliahan.



Gambar 56. *Control flow* fact_perkuliahan

Pada bagian temp_fact_perkuliahan memiliki beberapa *task* yang bertujuan untuk melakukan penggabungan beberapa tabel. Adapun tabel yang dimaksud meliputi tabel ods.mahasiswa_aktif, tabel ods.student_course_krs, tabel ods.transkrip, dan tabel ods.mahasiswa. Dalam penggabungan ini dibagi menjadi tiga tahap, tahap pertama menggabungkan tabel ods.mahasiswa aktif dengan tabel ods.student_course_krs, kemudian pada tahap kedua, hasil dari tahap pertama

digabungkan dengan tabel ods.transkrip, dan pada tahap ketiga menggabungkan hasil tahap kedua dengan tabel ods.mahasiswa.



Gambar 57. Penggabungan tahap pertama temp_fact_perkuliah

Tahap pertama penggabungan antara tabel ods.mahasiswa aktif dan tabel ods.student_course_krs dapat dilihat pada Gambar 57. Dalam penggabungan ini kolom yang menjadi penghubung antara kedua tabel adalah studentid, thajaranid, termid. Penggabungan ini menggunakan penggabungan tipe LEFT JOIN dan tabel ods.mahasiswa_aktif yang menjadi tabel sebelah kiri. Tujuan penggabungan dari kedua tabel ini adalah untuk mendapatkan mata kuliah yang telah diambil oleh mahasiswa aktif. Adapun penerapan dari penggabungan kedua tabel tersebut ditampilkan pada Gambar 58.

Join type: Left outer join Swap Inputs

Sort 1

<input checked="" type="checkbox"/>	Name	Order	Join Key
<input checked="" type="checkbox"/>	studentid	1	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	termid	3	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	thajaranid	2	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	status_mahasiswa_id	0	<input type="checkbox"/>

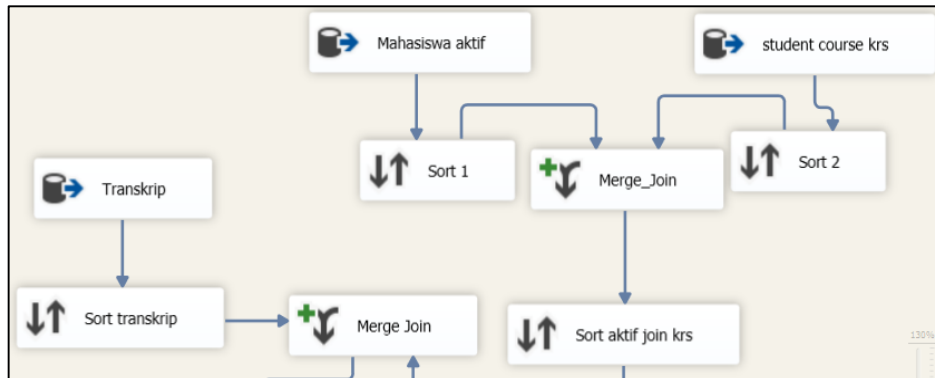
Sort 2

<input type="checkbox"/>	Name	Order	Join Key
<input type="checkbox"/>	studentid	1	<input checked="" type="checkbox"/>
<input type="checkbox"/>	termid	3	<input checked="" type="checkbox"/>
<input type="checkbox"/>	thajaranid	2	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	course_id	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	faculty_id	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	department...	0	<input type="checkbox"/>

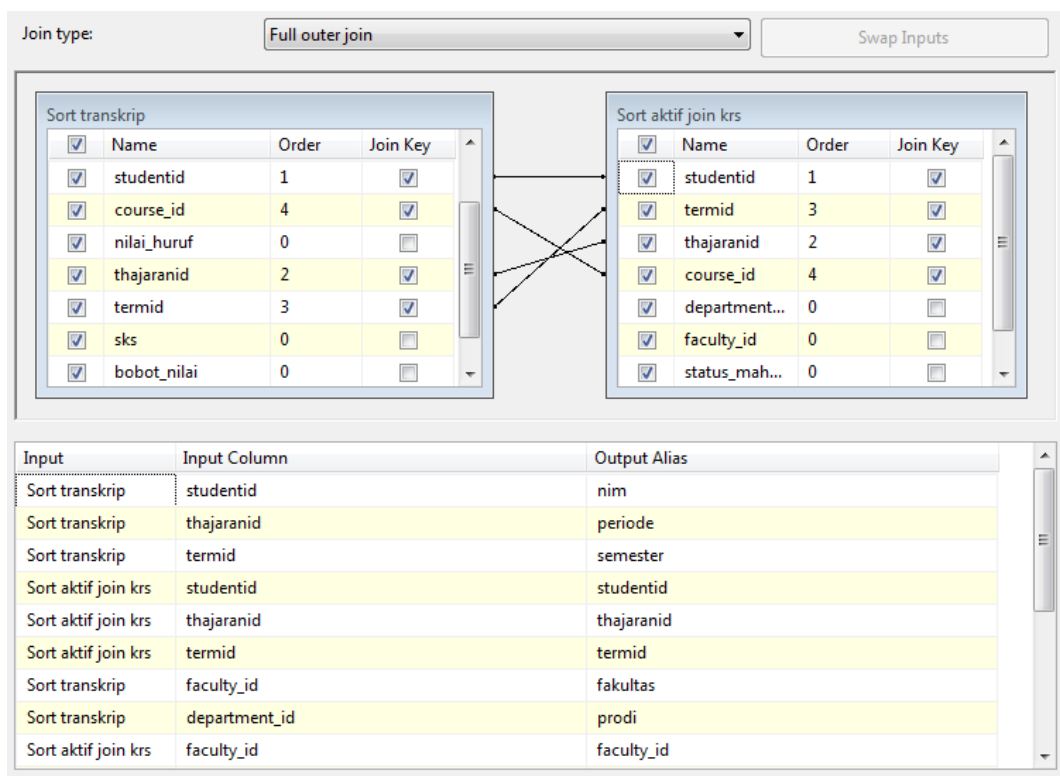
Input	Input Column	Output Alias
Sort 1	studentid	studentid
Sort 1	termid	termid
Sort 1	thajaranid	thajaranid
Sort 2	course_id	course_id
Sort 2	department_id	department_id
Sort 2	faculty_id	faculty_id
Sort 1	status_mahasiswa_id	status_mahasiswa_id

Gambar 58. LEFT JOIN tahap pertama fact_perkuliahan

Tahap kedua dari penggabungan antara hasil tahap pertama dan tabel ods.transkrip. Penggabungan antara dua tabel ini dihubungkan oleh kolom studentid, course_id, thajaranid dan termid. Tipe penggabungan yang digunakan adalah FULL JOIN. Penggabungan ini bertujuan untuk mendapatkan nilai dari mahasiswa per mata kuliah. Adapun hasil dari *data flow task* tahap kedua dan hasil penerapan penggabungan data ditunjukkan pada Gambar 59 dan Gambar 60.



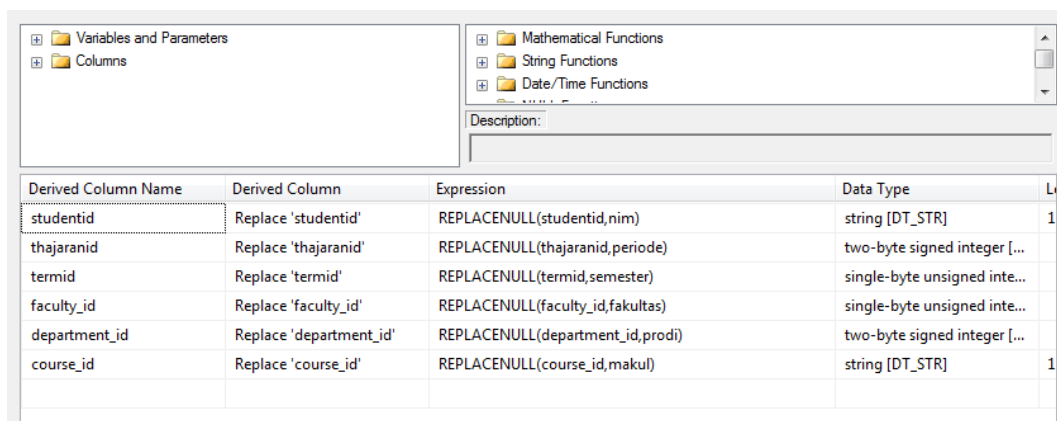
Gambar 59. *Data flow* tahap kedua temp_fact_perkuliahan



Gambar 60. FULL JOIN tahap kedua temp_fact_perkuliahan

Sebelum masuk ke tahap ketiga, terdapat satu *task* yang bernama *derived column*. *Task* ini memiliki fungsi untuk mengganti nilai dari suatu kolom. Dalam proses penggabungan tabel ods.transkrip dan hasil tahap pertama ternyata

ditemukan bahwa masih banyak data mahasiswa pada tabel.student_course_krs tidak tersedia pada tabel ods.transkrip dan begitu pula sebaliknya. *Task* ini bertujuan mengganti nilai yang tidak tersedia tersebut atau *null*. Adapun penerapan dari *derived column* ditunjukkan pada Gambar 61.



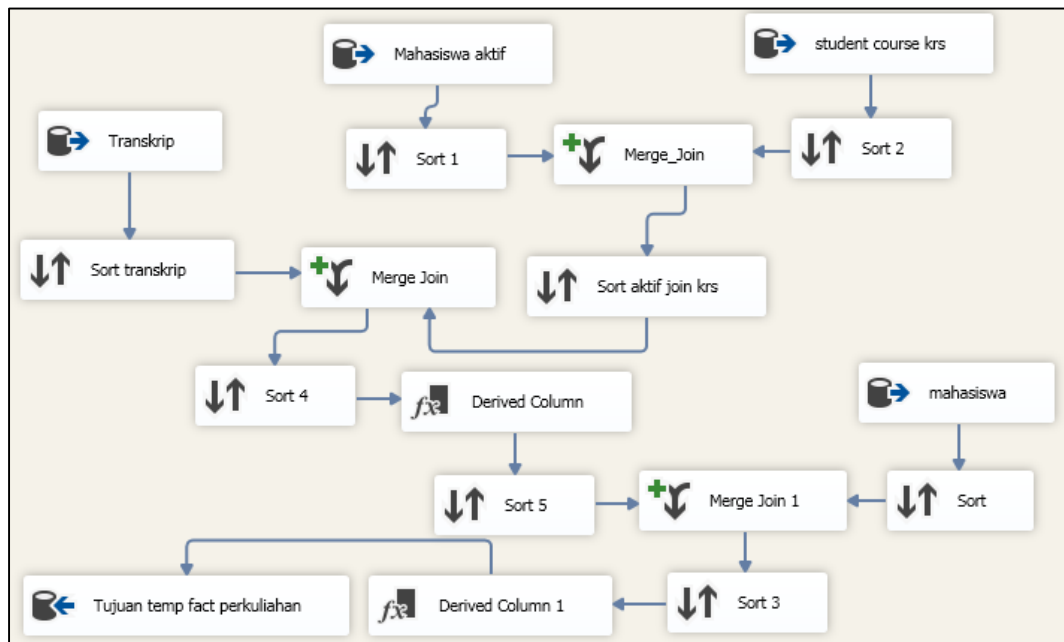
Derived Column Name	Derived Column	Expression	Data Type
studentid	Replace 'studentid'	REPLACENULL(studentid,nim)	string [DT_STR]
thajaranid	Replace 'thajaranid'	REPLACENULL(thajaranid,periode)	two-byte signed integer [...]
termid	Replace 'termid'	REPLACENULL(termid,semester)	single-byte unsigned inte...
faculty_id	Replace 'faculty_id'	REPLACENULL(faculty_id,fakultas)	single-byte unsigned inte...
department_id	Replace 'department_id'	REPLACENULL(department_id,prodi)	two-byte signed integer [...]
course_id	Replace 'course_id'	REPLACENULL(course_id,makul)	string [DT_STR]

Gambar 61. *Derived column* temp_fact_perkuliahan

Tahap ketiga adalah penggabungan antara hasil tahap kedua dan tabel ods.mahasiswa. Kolom yang digunakan sebagai penghubung kedua tabel tersebut adalah studentid. Penggabungan ini bertujuan untuk mendapatkan beberapa kolom yang berasal dari tabel ods.mahasiswa, meliputi department_id, faculty_id, thajaranid, termid, status_transfer, classprog_id, status_terdaftar_id, dan status_mahasiswa_id. Semua kolom tersebut digunakan sebagai relasi terhadap kolom *surrogate key* pada setiap tabel *dimension*.

Hasil dari tahap ketiga kemudian di proses kembali untuk *cleansing* data melalui *derived column 1*. Dalam proses dilakukan *cleansing* data yang *null* pada kolom course_id, status_mahasiswa_id, sks, bobot_nilai, dan nilai_huruf. Adapun

hasil dari tahap ketiga dan keseluruhan *data flow task* temp_fact_perkuliahan terlihat pada Gambar 62.



Gambar 62. *Data flow* task temp_fact_perkuliahan

Seluruh data yang telah dimuat temp_fact_perkuliahan di proses kembali untuk dimuat ke dalam tabel fact_perkuliahan. Dalam proses ETL dari temp_fact_perkuliahan menuju fact_perkuliahan dibantu menggunakan *Lookup*. Proses ini bertujuan untuk mengambil kolom *surrogate key* dari setiap tabel *dimension* melalui kolom-kolom yang telah disebutkan pada tahap ketiga.

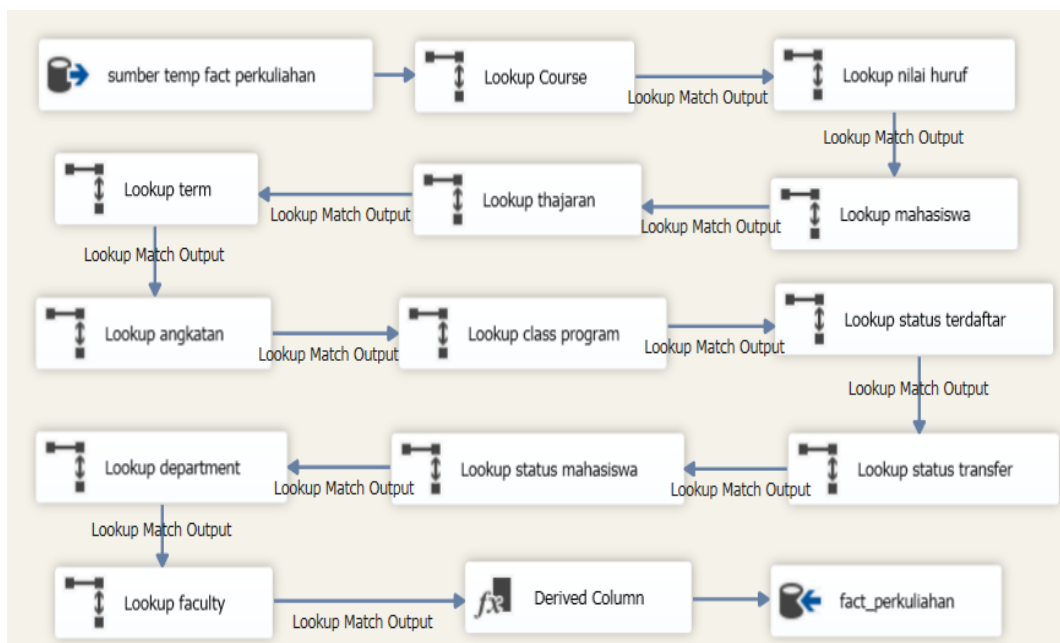
Sebelum melakukan proses *lookup* terdapat suatu proses untuk melakukan perhitungan IP per semester dan IPK. Proses perhitungan ini dilakukan saat pengambilan data dari tabel temp_fact_perkuliahan. Adapun proses perhitungan ditampilkan pada Gambar 63.

```

SELECT [studentid],[angkatan_id],[thajaranid]
,[termid],[faculty_id],[department_id]
,[classprog_id],[status_terdaftar_id]
,[status_transfer],[status_mahasiswa_id]
,[course_id],[nilai_huruf]
,[sks],[bobot_nilai]
,[sum(sks * bobot_nilai) over (partition by studentid,thajaranid,termid) /
nullif(sum(sks) over(partition by studentid,thajaranid,termid),0) IP
,[sum(sks * bobot_nilai) over (partition by studentid order by thajaranid,termid) /
nullif(sum(sks) over(partition by studentid order by thajaranid,termid),0) IPK
FROM [DW_Mahasiswa_ODS_SKRIPSI].[dbo].[temp_fact_perkuliahan]
group by [studentid],[angkatan_id]
,[thajaranid],[termid]
,[faculty_id],[department_id]
,[classprog_id],[status_terdaftar_id]
,[status_transfer],[status_mahasiswa_id]
,[course_id],[nilai_huruf]
,[sks],[bobot_nilai]
order by studentid,thajaranid,termid

```

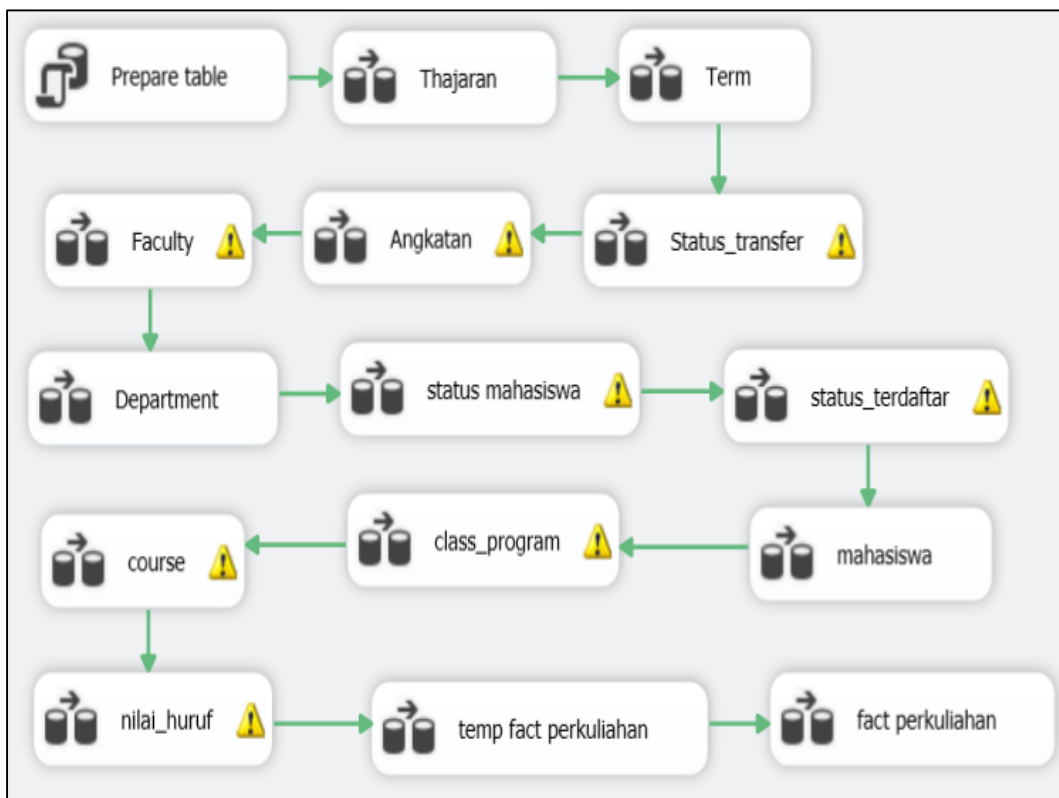
Gambar 63. Query perhitungan IP dan IPK



Gambar 64. Data flow task fact_perkuliahan

Pada Gambar 64 merupakan *data flow task* yang terdiri dari beberapa *lookup* dan kemudian hasil akhir dari proses ini dimuat ke dalam tabel *fact_perkuliahan*. Hasil akhir dari proses ini membentuk sebuah tabel *fact* yang terdiri dari beberapa

foreign key dan kolom yang bernilai *measure*. Adapun hasil dari *control flow task* *fact_perkuliahan* dan hasil pangkalan data dalam bentuk *star schema* ditunjukkan pada Gambar 65 dan Gambar 66. Pada Gambar 67 ditampilkan hasil sebagian dari tabel *fact_perkuliahan*. Total data yang ada pada tabel *fact_perkuliahan* sejumlah 1067469.



Gambar 65. Control flow task *fact_perkuliahan*



Gambar 66. Star schema pangkalan data.

student_key	angkatan_key	thajaran_key	term_key	faculty_key	department_key	classprog_key	status_terdaftar_key	status_transfer_key	status_mahasiswa_key	course_key	nilai_huruf_key	sks	bobot_nilai	IP	IPK
26	167	8	11	2	1	1	1	0	2	88	1	4.0	4.00	3.08	2.40
27	167	8	11	2	1	1	1	0	2	154	3	2.0	3.00	3.08	2.40
28	167	8	11	2	1	1	1	0	2	157	5	2.0	2.00	3.08	2.40
29	167	8	11	2	1	1	1	0	2	49	1	1.0	4.00	3.08	2.40
30	167	8	11	2	1	1	1	0	2	51	3	1.0	3.00	3.08	2.40
31	390	9	11	1	1	2	1	0	2	406	11	2.0	4.00	4.00	4.00
32	512	10	9	2	1	1	1	0	2	64	6	2.0	1.00	2.00	2.00
33	512	10	9	2	1	1	1	0	2	77	5	2.0	2.00	2.00	2.00
34	512	10	9	2	1	1	1	0	2	65	3	2.0	3.00	2.00	2.00
35	512	10	9	3	1	1	1	0	2	14	5	3.0	2.00	1.60	1.81

Gambar 67. Hasil sebagian tabel fact_perkuliahan

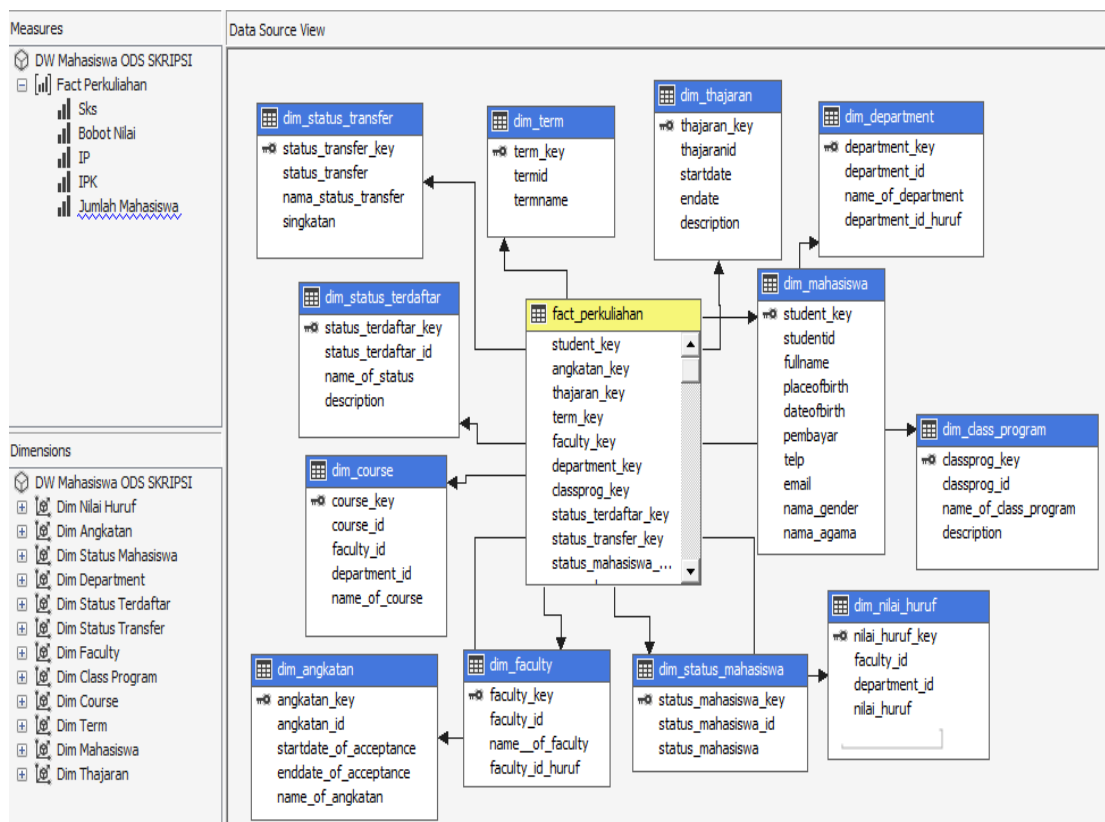
student_key	studentid	fullname	placeofbirth	dateofbirth	pembayar	telp	email	nama_gender	nama_agama	
1	29084	20110110001	Endah Melati Dewi	Wonosobo	1993-05-10 00:00:00.000	1100610087	0286 325436	endah.melati.2011@ft.umy.ac.id	Perempuan	Islam
2	29085	20110110002	Angki Yudha Mahardika	Yogyakarta	1992-03-03 00:00:00.000	1199130013	0274798525	angki.yudha.2011@ft.umy.ac.id	Laki-Laki	Islam
3	29086	20110110003	Atikah Istafada Maha	Cilacap	1994-05-26 00:00:00.000	1100610193	0274647405	atikah.istafada.2011@ft.umy.ac.id	Perempuan	Islam
4	29087	20110110004	Puspa Presisda Romasari	Magelang	1993-03-09 00:00:00.000	1100710355	0293 3285216	puspa.presisda.2011@ft.umy.ac.id	Perempuan	Islam
5	29088	20110110005	Yuli Triawan	Mulya Bhakti	1993-07-02 00:00:00.000	1100710234	085764005331	yuli.triawan.2011@ft.umy.ac.id	Laki-Laki	Islam
6	29089	20110110006	Audika Elba Sabila	Yogyakarta	1993-08-06 00:00:00.000	1100110165	0967 591989	audika.elba.2011@ft.umy.ac.id	Laki-Laki	Islam
7	29090	20110110007	Khairil Fajri Indraga	Curup	1993-10-09 00:00:00.000	1100510086	085664908877	khairil.fajri.2011@ft.umy.ac.id	Laki-Laki	Islam
8	29091	20110110008	Nuzul Fathoni	Blora	1993-03-22 00:00:00.000	1100610058	0296532079	nuzul.fathoni.2011@ft.umy.ac.id	Perempuan	Islam
9	29092	20110110009	Dendi Rizki Prabowo	Tanjung Pandan	1993-12-07 00:00:00.000	1199110785	081949168193	dendi.rizki.2011@ft.umy.ac.id	Laki-Laki	Islam
10	29093	20110110010	Dwi Cahyo Nugroho	Sleman	1992-02-24 00:00:00.000	1199110709	085213130880	dwi.cahyo.2011@ft.umy.ac.id	Laki-Laki	Islam
11	29094	20110110011	Arif Rahman Sutejo	Palembang	1992-11-29 00:00:00.000	1100710270	0291 438437	arif.rahman.2011@ft.umy.ac.id	Laki-Laki	Islam
12	29095	20110110012	Feri Muhammad	Jejara	1993-01-02 00:00:00.000	1100510036	085290312221	feri.muhammad.2011@ft.umy.ac.id	Laki-Laki	Islam

Gambar 68. Hasil sebagian tabel dim_mahasiswa

4.5. Hasil Pengujian

Dalam penelitian ini pengujian yang dilakukan adalah pengujian fungsional, validitas data dan waktu proses *query*. Pengujian dilakukan dengan menggunakan *tool* yang tersedia pada SSDT yaitu *Analysis Service* atau lebih dikenal dengan nama *SQL Server Analysis Service (SSAS)*.

SSAS mengubah data dimensional menjadi ke bentuk multidimensional atau juga dikenal *cube*. Adapun penerapan *cube* pada SSAS ditunjukkan pada Gambar 69.



Gambar 69. *Cube* di SSAS

Pengujian fungsional yang dilakukan adalah memastikan sistem pangkalan data mampu memenuhi kebutuhan informasi yang ada. Kebutuhan informasi yang utama adalah untuk memenuhi kebutuhan pada borang akreditasi standar 3 dengan subjek mahasiswa. Dalam tingkat universitas terdapat tiga tingkatan dokumen borang yaitu, universitas, fakultas, dan program studi. Adapun hasil dari pengujian ditunjukkan pada Tabel 24, Tabel 25 dan Tabel 26.

Tabel 24. Format laporan borang tingkat universitas

Tahun Akademik	Jumlah Mahasiswa Baru		Jumlah Total Mahasiswa (<i>Student Body</i>)	
	Regular bukan Transfer	Transfer	Regular bukan Transfer	Transfer
Program Akademik (S-1, S-2, S-3)				
TS-4	2420	-	9731	-
TS-3	3281	-	10621	-
TS-2	4863	-	13369	-
TS-1	6308	-	17130	-
TS	5753	-	19435	-

Tabel 25. Format laporan borang tingkat program studi

Tahun Akademik	Jumlah Mahasiswa Baru		Jumlah Total Mahasiswa	
	Regular bukan Transfer	Transfer	Regular bukan Transfer	Transfer
TS-4	66	-	66	-
TS-3	76	-	113	-
TS-2	129	-	222	-
TS-1	210	-	385	-
TS	115	-	418	-

Tabel 26. Format laporan borang tingkat fakultas

No.	Hal		Jumlah Mahasiswa pada PS:				Total Mahasiswa pada Fakultas
			Teknik Elektro	Teknik Mesin	Teknik Sipil	Teknologi Informasi	
1	Program reguler	1. Mhs. baru bukan transfer	233	283	286	115	917
		2. Mhs. baru transfer	-	-	-	-	-
		3. Total mhs. regular (<i>Student Body</i>)	574	977	1160	418	3129
2	Program non-reguler	1. Mhs. baru bukan transfer	-	-	-	-	-
		2. Mhs. baru transfer	-	-	-	-	-
		3. Total mhs. non-reguler (<i>Student Body</i>)	-	-	-	-	-

Pengujian tidak hanya dilakukan untuk mendapatkan kebutuhan utama tapi juga untuk mendapatkan kebutuhan tambahan lainnya serta memastikan validitas data yang dihasilkan seperti data IP dan IPK. Pengujian ini menggunakan contoh IP dan IPK dari salah satu mahasiswa mulai dari tahun ajaran 2011 hingga 2013. Adapun hasil pengujian tersebut ditampilkan pada Gambar 70.

studentid	thajaranid	temid	IP	IPK
20110140015	2011	1	3.79	3.79
20110140015	2011	2	3.11	3.45
20110140015	2012	1	3.89	3.59
20110140015	2012	2	3.64	3.60
20110140015	2013	1	3.88	3.64
20110140015	2013	2	3.33	3.61
20110140015	2014	1	0.00	3.61

(a)

thajaran	term	IP	IPK
2011	Gasal	3.79	3.79
2011	Genap	3.11	3.45
2012	Gasal	3.89	3.59
2012	Genap	3.64	3.6
2013	Gasal	3.88	3.64
2013	Genap	3.33	3.61
2014	Gasal	0	3.61

(b)

Gambar 70. Hasil pengujian validitas IP dan IPK

Pada Gambar 70 bagian (a) merupakan hasil pencarian dari pangkalan data dan bagian (b) dari excel. Hasil dari excel merupakan hasil perhitungan secara manual. Hasil tersebut digunakan sebagai acuan untuk memastikan kebenaran data yang dihasilkan oleh pangkalan data. Dari gambar tersebut dapat dilihat bahwa nilai yang dihasilkan pangkalan data sesuai dengan nilai yang dihitung pada excel. Berdasarkan hasil pengujian dapat dikatakan bahwa sistem telah mampu memberikan hasil yang sesuai bagi kebutuhan informasi yang ada serta menghasilkan informasi yang benar atau valid.

Dari hasil pengujian berikutnya juga ditemukan bahwa dengan menerapkan pangkalan data akan mengurangi waktu proses *query* atau pencarian informasi. Pengujian ini menampilkan informasi nilai IP dan IPK yang merupakan kebutuhan tambahan. Adapun hasil dari pengujian beserta hasil pencarian ditampilkan pada Gambar 71, Gambar 72, Gambar 73 dan Gambar 74.

```

set statistics time on
SELECT [faculty_id],[department_id],[studentid],[course_id],[nilai_huruf]
,[thajaranid],[termid],[sks],[bobot_nilai]
,ROUND(sum(sks * bobot_nilai) over (partition by studentid,thajaranid,termid) /
nullif(sum(sks) over(partition by studentid,thajaranid,termid),0),2) IP
,ROUND(sum(sks * bobot_nilai) over (partition by studentid order by thajaranid,termid) /
nullif(sum(sks) over(partition by studentid order by thajaranid,termid),0),2) IPK
FROM [DW_Mahasiswa_ODS_SKRIPSI].[dbo].[buffer_transkrip]
where studentid = '20110140015'
group by [faculty_id],[department_id],[studentid],[course_id],[nilai_huruf]
,[thajaranid]
,[termid]
,[sks]
,[bobot_nilai]
order by studentid,course_id,thajaranid,termid

```

0 %

Results Messages

(51 row(s) affected)

SQL Server Execution Times:
CPU time = 313 ms, elapsed time = 102 ms.

Gambar 71. Pengujian waktu proses *query* dari data sumber

faculty_id	department_id	studentid	course_id	nilai_huruf	thajaranid	termid	sks	bobot_nilai	IP	IPK
1	14	20110140015	TI-104	A	2011	1	2.0	4.00	3.790000	3.790000
1	14	20110140015	TI-105	A	2011	1	2.0	4.00	3.790000	3.790000
1	14	20110140015	TI-105P	A	2011	1	1.0	4.00	3.790000	3.790000
1	14	20110140015	TI-106	A	2011	1	2.0	4.00	3.790000	3.790000
1	14	20110140015	TI-106P	A	2011	1	1.0	4.00	3.790000	3.790000
1	14	20110140015	TI-107	A	2011	1	3.0	4.00	3.790000	3.790000
1	14	20110140015	TI-107P	A	2011	1	1.0	4.00	3.790000	3.790000
1	14	20110140015	TI-201	B	2011	2	3.0	3.00	3.110000	3.450000
1	14	20110140015	TI-202	A	2011	2	2.0	4.00	3.110000	3.450000
1	14	20110140015	TI-203	B	2011	2	2.0	3.00	3.110000	3.450000
1	14	20110140015	TI-204	B	2011	2	2.0	3.00	3.110000	3.450000
1	14	20110140015	TI-204P	B	2011	2	1.0	3.00	3.110000	3.450000

Gambar 72. Hasil Pencarian IP dan IPK dari data sumber

```

set statistics time on
SELECT      dbo.dim_mahasiswa.studentid, dbo.dim_faculty.faculty_id, dbo.dim_department.department_id,
            dbo.dim_thajaran.thajaranid, dbo.dim_term.termid,
            dbo.dim_course.course_id, dbo.dim_nilai_huruf.nilai_huruf,
            dbo.fact_perkuliahan.IP, dbo.fact_perkuliahan.IPK
FROM        dbo.fact_perkuliahan INNER JOIN
            dbo.dim_course ON dbo.fact_perkuliahan.course_key = dbo.dim_course.course_key INNER JOIN
            dbo.dim_department ON dbo.fact_perkuliahan.department_key = dbo.dim_department.department_key INNER JOIN
            dbo.dim_faculty ON dbo.fact_perkuliahan.faculty_key = dbo.dim_faculty.faculty_key INNER JOIN
            dbo.dim_mahasiswa ON dbo.fact_perkuliahan.student_key = dbo.dim_mahasiswa.student_key INNER JOIN
            dbo.dim_nilai_huruf ON dbo.fact_perkuliahan.nilai_huruf_key = dbo.dim_nilai_huruf.nilai_huruf_key INNER JOIN
            dbo.dim_term ON dbo.fact_perkuliahan.term_key = dbo.dim_term.term_key INNER JOIN
            dbo.dim_thajaran ON dbo.fact_perkuliahan.thajaran_key = dbo.dim_thajaran.thajaran_key
WHERE       (dbo.dim_mahasiswa.studentid = '20110140015') order by thajaranid,termid

```

(76 row(s) affected)

SQL Server Execution Times:
CPU time = 94 ms, elapsed time = 65 ms.

Gambar 73. Pengujian waktu proses *query* dari pangkalan data

studentid	faculty_id	department_id	thajaranid	termid	course_id	nilai_huruf	IP	IPK
20110140015	1	14	2011	1	TI-106	A	3.79	3.79
20110140015	1	14	2011	1	TI-106P	A	3.79	3.79
20110140015	1	14	2011	1	TI-104	A	3.79	3.79
20110140015	1	14	2011	1	TI-105	A	3.79	3.79
20110140015	1	14	2011	1	TI-107	A	3.79	3.79
20110140015	1	14	2011	1	TI-107P	A	3.79	3.79
20110140015	1	14	2011	2	TI-204P	B	3.11	3.45
20110140015	1	14	2011	2	TI-202	A	3.11	3.45
20110140015	1	14	2011	2	TI-204	B	3.11	3.45
20110140015	1	14	2011	2	TI-207P	B	3.11	3.45
20110140015	1	14	2011	2	TI-203	B	3.11	3.45
20110140015	1	14	2011	2	TI-206	B	3.11	3.45

Gambar 74. Hasil Pencarian IP dan IPK dari pangkalan data

Dari gambar 71 dan 73 dapat ditemukan terdapat perbedaan waktu proses eksekusi *query* dimana waktu proses *query* pada pangkalan data lebih rendah dibanding pada data sumber. Hal ini disebabkan karena adanya proses kalkulasi terlebih dahulu sedangkan pada sistem pangkalan data proses kalkulasi telah dilakukan sebelum dimuat ke dalam tabel *fact*.