

BAB V

PROSES ETL, IMPLEMENTASI CUBE, DAN PEMBUATAN REPORTS

Pada bab ini akan dijelaskan proses ETL (*Extract, Transform, Loading*), implementasi *cube* dan pembuatan *reports*. Hal tersebut dilakukan agar data-data kemahasiswaan dan alumni dapat digunakan untuk mendukung dalam pengambilan keputusan. Sebelum digunakan, data-data tersebut harus dianalisis terlebih dahulu dengan membentuk sebuah *cube*. *Cube* tersebut dapat dipergunakan untuk membuat *report* yang membantu dalam pengambilan keputusan strategis untuk Universitas Atma Jaya Yogyakarta.

5.1 Proses ETL (*Extract, Transform, Loading*) SIKMA

Ada dua cara membuat *control flow items*, yaitu *full refresh* dan *incremental load*. *Full refresh* digunakan untuk menghapus terlebih dahulu semua data di *database* yang akan di-load. *Incremental load* digunakan untuk mengubah data yang datanya berubah di *database* transaksional.

5.1.1 Mempersiapkan *Database* untuk *Staging Area* dan *Data Warehouse*

Sebelum memulai proses transformasi dan *loading* data, terlebih dahulu dipersiapkan *database-database* yang akan digunakan, yaitu **SIKMA_STAGING** untuk *staging area*, **SIKMA_DW** untuk *data warehouse* sikma, dan **SIKMA_DW_ALUMNI** untuk *data warehouse* alumni. Untuk membuat proyek pada **SQL Server Integration Services** perlu diperhatikan bahwa *database-database* ini harus sudah ada di *server* karena desain yang dirancang merupakan desain untuk membangun *data warehouse*.

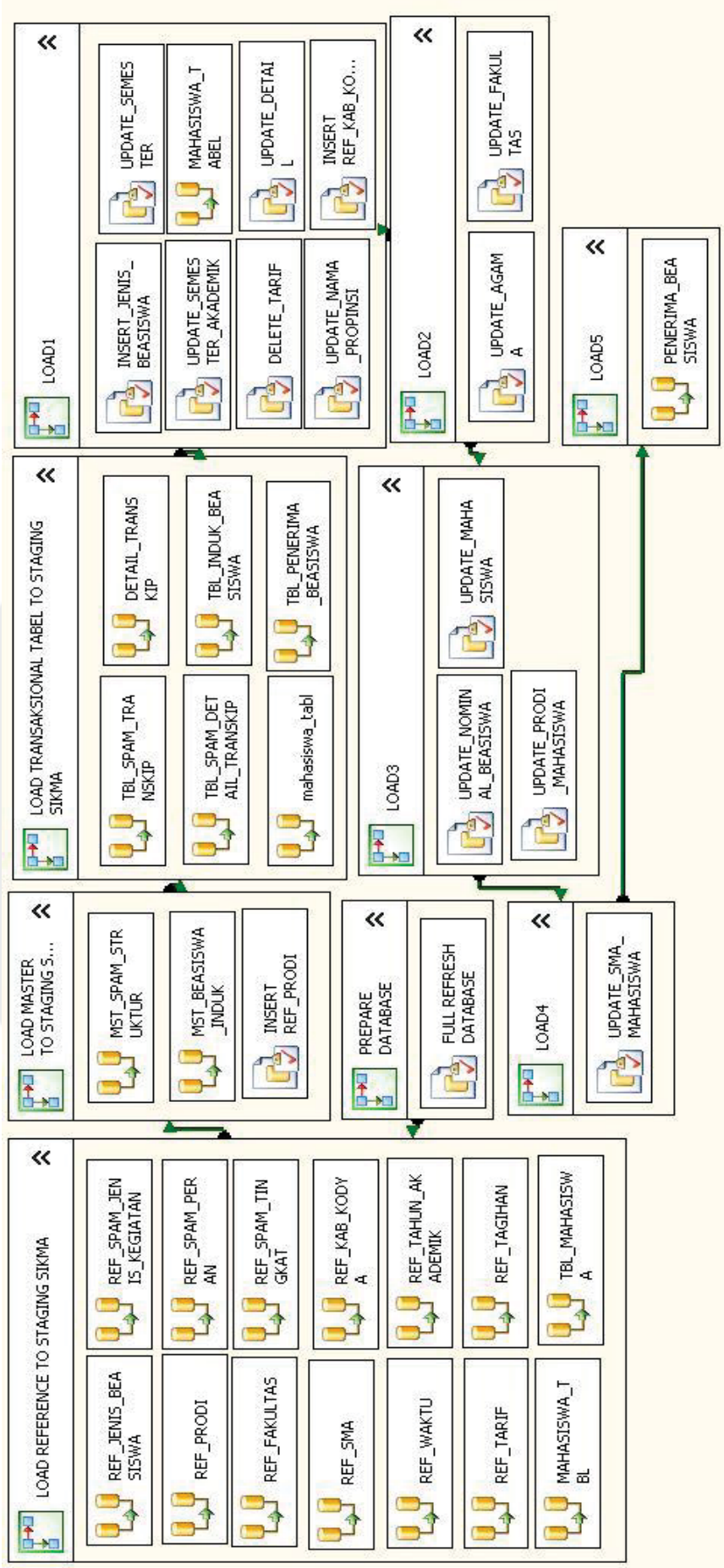
5.1.2. *Control Flow Full Refresh* SIKMA

Proses *full refresh* SIKMA terbagi menjadi dua proses. Proses pertama adalah mengisi data dari *database* sumber menuju *staging area*. Proses selanjutnya adalah mengisi data dari *staging area* menuju *data warehouse*.

5.1.2.1 Loading Data dari Sumber Data ke SIKMA_STAGING

Pada proses *loading* data dari *database* sumber ke dalam *database* SIKMA_STAGING, langkah pertama adalah menentukan *control flow items* yang sesuai dengan kebutuhan pada *tab control flow*. Untuk menggunakan *control flow items* yang telah disediakan, tarik dan letakkan ikon *control flow items* yang diperlukan ke area tengah. Kemudian proses *loading* data dari sumber data ke SIKMA_STAGING pada *package* Staging_sikma.dtsx akan terbentuk desain proses sebagai berikut :





Gambar 5.1. Desain Control Flow pada Staging_sikma – Full Refresh

Untuk membuat desain *Control Flow* di atas, langkah-langkah yang dilakukan adalah sebagai berikut:

1). *Sequence Container* **Prepare Database**

Tarik dan letakkan ikon *sequence container* dan beri nama **PREPARE DATABASE**.

2). *Execute SQL Task* **FULL_REFRESH_DATABASE**

Tarik dan letakkan ikon *execute SQL task* ke dalam item **PREPARE DATABASE** yang telah dibuat dan beri nama **FULL_REFRESH_DATABASE**. Ikon *execute SQL task* berfungsi menjalankan *query* untuk menghapus semua isi tabel yang ada pada *database* **SIKMA_STAGING**. Hal ini dilakukan agar isi tabel pada *database* **SIKMA_STAGING** selalu baru. Buat koneksi ke *database* pada *server* **DATA-WAREHOUSE**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA_STAGING**). Langkah-langkah pembuatan koneksi **DATA-WAREHOUSE.SIKMA_STAGING** dapat diamati pada lampiran tugas akhir ini. Setelah proses tersebut, lakukan konfigurasi dengan menuliskan *sintaks* berikut:

```
DELETE FROM REF_SPAM_JENIS_KEGIATAN; DELETE FROM REF_SPAM_PERAN; DELETE FROM REF_SPAM_TINGKAT; DELETE FROM MST_SPAM_STRUKTUR; DELETE FROM TBL_SPAM_TRANSKIP; DELETE FROM TBL_SPAM_DETAIL_TRANSKIP; DELETE FROM MST_BEASISWA_INDUK; DELETE FROM REF_SMA; DELETE FROM TBL_PENERIMA_BEASISWA; DELETE FROM REF_JENIS_BEASISWA; DELETE FROM MAHASISWA_TABEL; DELETE FROM REF_KAB_KODYA; DELETE FROM REF_PRODI; DELETE FROM MAHASISWA_TBL; DELETE FROM REF_FAKULTAS; DELETE FROM MAHASISWA; DELETE FROM REF_TAHUN_AKADEMIK; DELETE FROM Mahasiswa_tabl; DELETE FROM REF_WAKTU; DELETE FROM REF_TARIF; DELETE FROM REF_TAGIHAN; DELETE FROM PENERIMA_BEASISWA;
```

3). *Sequence Container* **LOAD REFERENCE TO STAGING SIKMA**

Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD REFERENCE TO STAGING SIKMA**.

4). *Data Flow Task* **LOAD REFERENCE TO STAGING SIKMA**

Control flow item yang digunakan untuk ekstrak, *transform*, dan *load* data adalah *data flow task*. Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* **LOAD REFERENCE TO STAGING SIKMA**. Karena

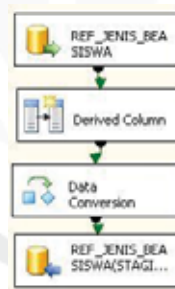
ada 14 data sumber yang akan di-load maka dibuat 14 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **REF_JENIS_BEASISWA**, **REF_PRODI**, **REF_FAKULTAS**, **REF_SMA**, **REF_WAKTU**, **REF_TARIF**, **MAHASIWA_TBL**, **REF_SPAM_JENIS_KEGIATAN**, **REF_SPAM_TINGKAT**, **REF_SPAM_PERAN**, **REF_KAB_KODYA**, **REF_TAHUN_AKADEMIK**, **REF_TAGIHAN**, dan **TBL_MAHASIWA**.

Hubungkan **PREPARE DATABASE** *sequence container* dengan *sequence container* **LOAD REFERENCE TO STAGING SIKMA**.

5). *Data Flow Task* pada *Sequence Container* **LOAD REFERENCE TO STAGING**

Langkah-langkah untuk membuat *data flow task* pada **LOAD REFERENCE TO STAGING** sebagai berikut:

1. REF_JENIS_BEASISWA



Gambar 5.2. Data Flow Task REF_JENIS_BEASISWA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon **ADO.NET SOURCE** dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax
Data Access Mode : Table or view
Name of table or view : [dbo].[REF_JENIS_BEASISWA]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Fungsi *data conversion* digunakan untuk mengkonversi **JENIS_BEASISWA** menjadi data *varchar*.

- d. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

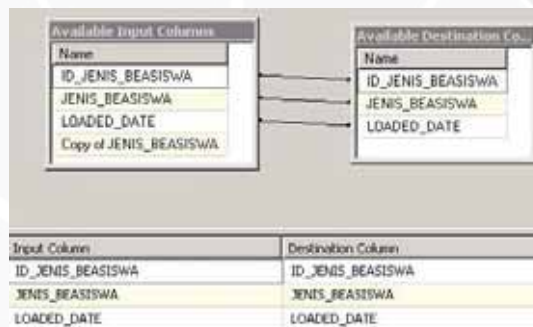
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[REF_JENIS_BEASISWA]

Karena tabel **REF_JENIS_BEASISWA** belum terbentuk pada *database*, maka buat tabel **REF_JENIS_BEASISWA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

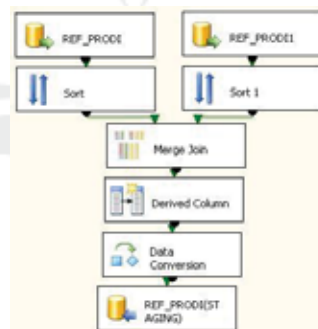
```
CREATE TABLE [REF_JENIS_BEASISWA]([ID_JENIS_ BEASISWA] INT,
[JENIS_BEASISWA] VARCHAR(60), [LOADED_DATE] DATETIME)
```

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.3. Mapping *Data Flow Task* REF_JENIS_BEASISWA – Full Refresh

2. REF_PRODI



Gambar 5.4. *Data Flow Task* REF_PRODI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) REF_PRODI1

Connection Manager :DATA-WAREHOUSE.siatmax

Data Access Mode :Table or view

Name of table or view :[dbo].[REF_PRODI]

b) REF_PRODI

Connection Manager : DATA-WAREHOUSE.biadw

Data Access Mode : Table or view

Name of table or view : [dbo].[prodi]

- b. Pada *data flow task* **REF_PRODI** dan **REF_PRODI1** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Kemudian data digabungkan berdasarkan **id_prodi** yang telah diurutkan untuk mengambil data **nama_prodi**.
- c. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.
- d. Fungsi *data conversion* digunakan untuk mengkonversi **ID_PRODI** menjadi data *varchar*.
- e. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

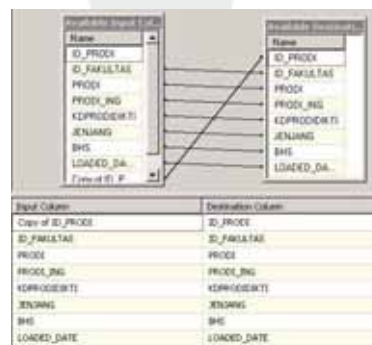
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[REF_PRODI]

Karena tabel **REF_PRODI** belum terbentuk pada *database*, maka buat tabel **REF_PRODI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

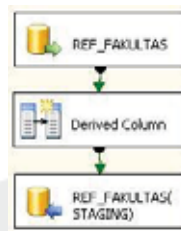
```
CREATE TABLE [REF_PRODI]( [ID_PRODI] VARCHAR (2), [ID_FAKULTAS] INT, [PRODI] VARCHAR(50), [PRODI_ING] VARCHAR (50), [KDPRODIKTI] VARCHAR (5), [JENJANG] VARCHAR(2), [BHS] VARCHAR(1), [LOADED_DATE] DATETIME)
```

- f. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.5. Mapping Data Flow Task REF_PRODI – Full Refresh

3. REF_FAKULTAS



Gambar 5.6. Data Flow Task REF_FAKULTAS – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax
Data Access Mode : Table or view
Name of table or view : [dbo].[REF_FAKULTAS]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

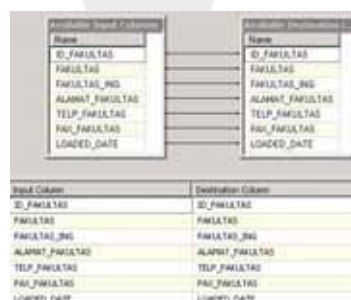
Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Use a Table or View : [dbo].[REF_FAKULTAS]

Karena tabel **REF_FAKULTAS** belum terbentuk pada *database*, maka buat tabel **REF_FAKULTAS** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

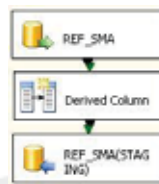
```
CREATE TABLE [REF_FAKULTAS]([ID_FAKULTAS] INT, [FAKULTAS]
VARCHAR(30), [FAKULTAS_ING] VARCHAR(30), [ALAMAT_ FAKULTAS]
VARCHAR(80), [TELP_FAKULTAS] VARCHAR(12), [FAX_FAKULTAS]
VARCHAR(12), [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.7. Mapping Data Flow Task REF_FAKULTAS – Full Refresh

4. REF_SMA



Gambar 5.8. Data Flow Task REF_SMA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.biadw
Data Access Mode : Table or view
Name of table or view : [dbo].[sma]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

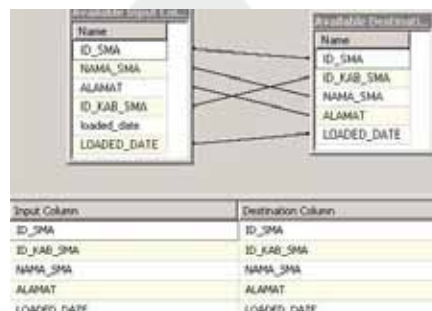
Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Use a Table or View : [dbo],[REF_SMA]

Karena tabel **REF_SMA** belum terbentuk pada *database*, maka buat tabel **REF_SMA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

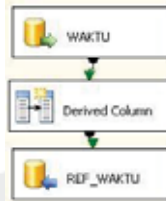
```
CREATE TABLE [REF_SMA]([ID_SMA] INT, [ID_KAB_SMA] INT,
[NAMA_SMA] VARCHAR(200), [ALAMAT] VARCHAR(150),
[LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.9. Mapping Data Flow Task REF_SMA – Full Refresh

5. REF_WAKTU



Gambar 5.10. Data Flow Task REF_WAKTU – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.biadw

Data Access Mode : Table or view

Name of table or view : [dbo].[time]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

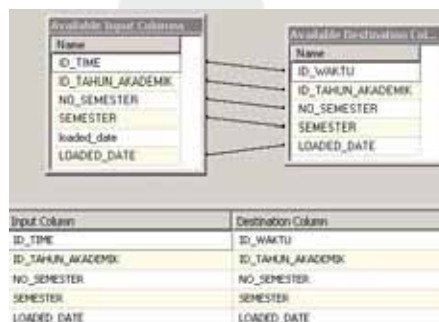
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[REF_WAKTU]

Karena tabel **REF_WAKTU** belum terbentuk pada *database*, maka buat tabel **REF_WAKTU** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

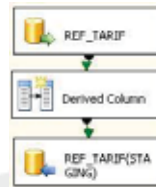
```
CREATE TABLE [REF_WAKTU]([ID_WAKTU] INT, [ID_ TAHUN_
AKADEMIK] INT, [NO_SEMESTER] INT, [SEMESTER] VARCHAR(7),
[LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.11. Mapping Data Flow Task REF_WAKTU – Full Refresh

6. REF_TARIF



Gambar 5.12. Data Flow Task REF_TARIF – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.sikeuDB

Data Access Mode : Table or view

Name of table or view : [dbo].[trnsTarif]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.
- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

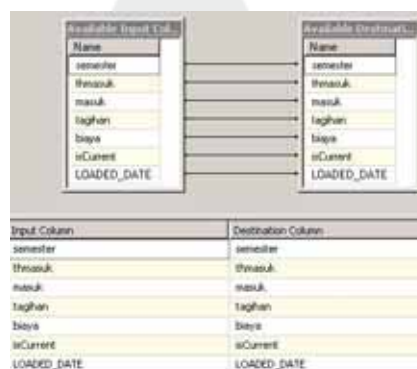
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[REF_TARIF]

Karena tabel **REF_TARIF** belum terbentuk pada *database*, maka buat tabel **REF_TARIF** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

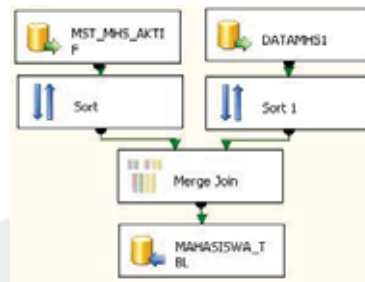
```
CREATE TABLE [REF_TARIF]([semester] VARCHAR (5), [thnmasuk] INT,
[masuk] VARCHAR(2), [tagihan] VARCHAR (3), [biaya] INT, [isCurret] BIT,
[LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.13. Mapping Data Flow Task REF_TARIF – Full Refresh

7. MAHASISWA_TBL



Gambar 5.14. Data Flow Task MAHASISWA_TBL – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **DATAMHS1**

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

Name of table or view : [dbo].[datamhs1]

b) **MST_MHS_AKTIF**

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

Name of table or view : [dbo].[MST_MHS_AKTIF]

- b. Pada *data flow task* **DATAMHS1** dan **MST_MHS_AKTIF** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **NAMA_MHS**, **TMP_LAHIR**, **JNS_KEL**, dan **AGAMA**.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

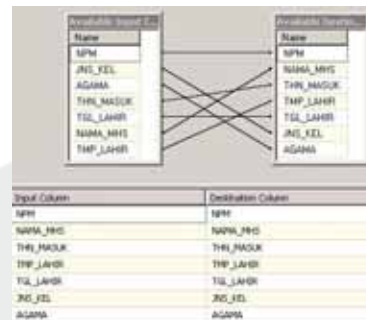
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[MAHASISWA_TBL]

Karena tabel **MAHASISWA_TBL** belum terbentuk pada *database*, maka buat tabel **MAHASISWA_TBL** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

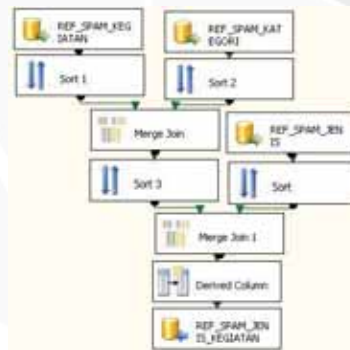
```
CREATE TABLE [MAHASISWA_TBL]([NPM] VARCHAR (20), [NAMA_ MHS]
VARCHAR(150), [THN_MASUK] NUMERIC(4,0), [TMP_ LAHIR]
NVARCHAR(70), [TGL_LAHIR] DATETIME, [JNS_KEL] VERCHAR(1),
[AGAMA] VARCHAR(15))
```

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.15. Mapping *Data Flow Task* MAHASISWA_TBL – Full Refresh

8. REF_SPAM_JENIS_KEGIATAN



Gambar 5.16. *Data Flow Task* REF_SPAM_JENIS_KEGIATAN – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) REF_SPAM_KEGIATAN

Connection Manager : DATA-WAREHOUSE.siatmax
 Data Access Mode : Table or view
 Name of table or view : [dbo].[REF_SPAM_KEGIATAN]

b) REF_SPAM_KATEGORI

Connection Manager : DATA-WAREHOUSE.siatmax
 Data Access Mode : Table or view
 Name of table or view : [dbo].[REF_SPAM_KATEGORI]

c) REF_SPAM_JENIS

Connection Manager : DATA-WAREHOUSE.siatmax
 Data Access Mode : Table or view
 Name of the table or view : [dbo].[REF_SPAM_JENIS]

- b. Pada *data flow task* REF_SPAM_KEGIATAN dan REF_SPAM_KATEGORI digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih

- dahulu. Data digabungkan berdasarkan **KODE_KATEGORI** yang telah diurutkan untuk mengambil data **KETERANGAN_KATEGORI**.
- c. Pada *data flow task* **REF_SPAM_JENIS** dan **MERGE JOIN** digabungkan dengan fungsi *merge join*. Masing-masing harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE_KEGIATAN** yang telah diurutkan untuk mengambil data **KETERANGAN_KEGIATAN**.
- d. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- e. Mendefinisikan *data flow destination* dengan tarik dan letakkan **ADO.NET DESTINATION** dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

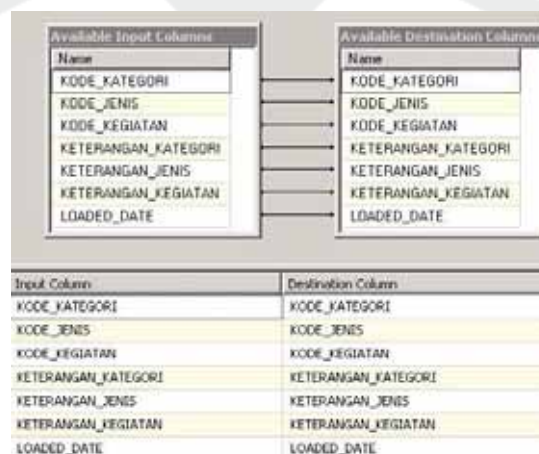
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[REF_SPAM_JENIS_KEGIATAN]

Karena tabel **REF_SPAM_JENIS_KEGIATAN** belum terbentuk pada *database*, maka buat tabel **REF_SPAM_JENIS_KEGIATAN** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

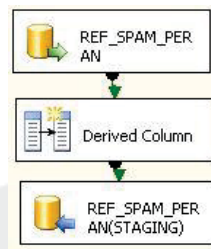
```
CREATE TABLE [REF_SPAM_JENIS_KEGIATAN] ([KODE_KATEGORI]
VARCHAR(1), [KODE_JENIS] INT, [KODE_KEGIATAN] INT,
[KETERANGAN_KATEGORI] VARCHAR(80), [KETERANGAN_JENIS]
VARCHAR(100), [KETERANGAN_KEGIATAN] VARCHAR (40),
[LOADED_DATE] DATETIME)
```

- f. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.17. Mapping *Data Flow Task* REF_SPAM_JENIS_KEGIATAN – Full Refresh

9. REF_SPAM_PERAN



Gambar 5.18. Data Flow Task REF_SPAM_PERAN – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax

Data Access Mode : Table or view

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

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

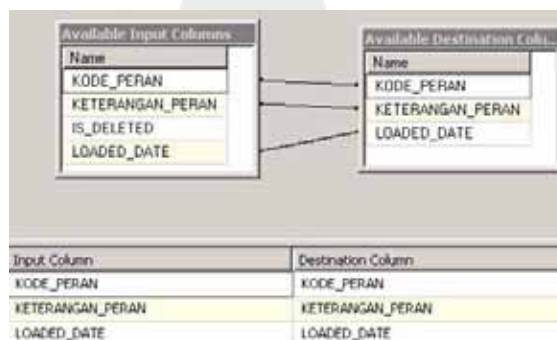
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[REF_SPAM_PERAN]

Karena tabel **REF_SPAM_PERAN** belum terbentuk pada *database*, maka buat tabel **REF_SPAM_PERAN** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

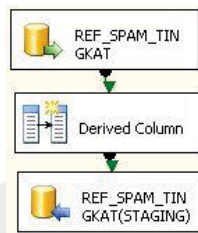
```
CREATE TABLE [REF_SPAM_PERAN]([KODE_PERAN] INT,
[KETERANGAN_PERAN] VARCHAR(30), [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.19. Mapping Data Flow Task REF_SPAM_PERAN – Full Refresh

10. REF_SPAM_TINGKAT



Gambar 5.20. Data Flow Task REF_SPAM_TINGKAT – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax

Data Access Mode : Table or view

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

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

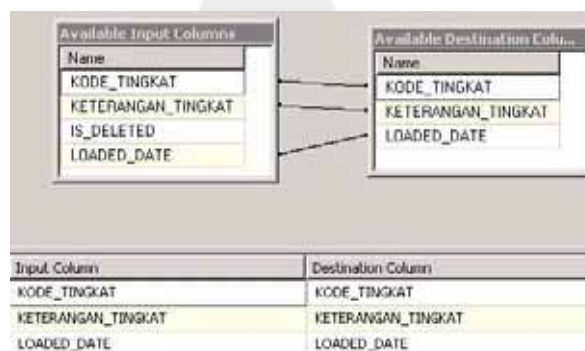
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[REF_SPAM_TINGKAT]

Karena tabel **REF_SPAM_TINGKAT** belum terbentuk pada *database*, maka buat tabel **REF_SPAM_TINGKAT** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

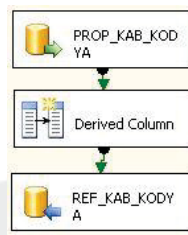
```
CREATE TABLE [REF_SPAM_TINGKAT]([KODE_TINGKAT] INT, [KETERANGAN_TINGKAT] VARCHAR(20), [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.21. Mapping Data Flow Task REF_SPAM_TINGKAT – Full Refresh

11. REF_KAB_KODYA



Gambar 5.22. Data Flow Task REF_KAB_KODYA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.biadw
Data Access Mode : Table or view
Name of the table or view : [dbo].[prop_kab_kodya]

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

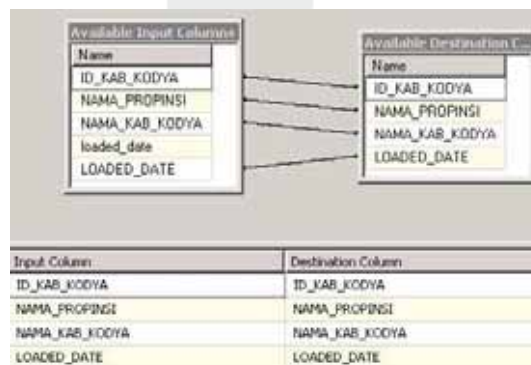
- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Use a Table or View : [dbo].[REF_KAB_KODYA]

Karena tabel **REF_KAB_KODYA** belum terbentuk pada *database*, maka buat tabel **REF_KAB_KODYA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

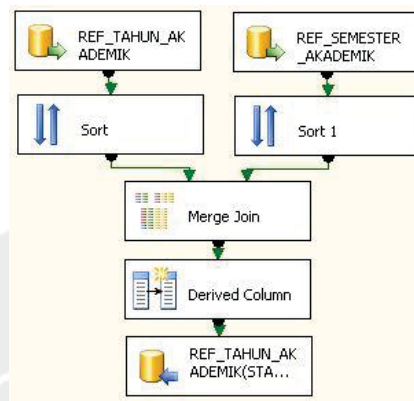
```
CREATE TABLE [REF_KAB_KODYA]([ID_KAB_KODYA] INT, [NAMA_PROPINSI] VARCHAR(150), [NAMA_KAB_KODYA] VARCHAR(150), [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.23. Mapping Data Flow Task REF_KAB_KODYA – Full Refresh

12. REF_TAHUN_AKADEMIK



Gambar 5.24. Data Flow Task REF_TAHUN_AKADEMIK – Full Refresh

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) REF_TAHUN_AKADEMIK

Connection Manager : DATA-WAREHOUSE.siatma
 Data Access Mode : Table or view
 Name of table or view : [dbo].[TBL_TAHUN_AKADEMIK]

b) REF_SEMESTER_AKADEMIK

Connection Manager : DATA-WAREHOUSE.siatma
 Data Access Mode : Table or view
 Name of table or view : [dbo].[TBL_SEMESTER_AKADEMIK]

b. Pada *data flow task* REF_TAHUN_AKADEMIK dan REF_SEMESTER_AKADEMIK digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus di urutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan ID_TAHUN_AKADEMIK yang telah diurutkan untuk mengambil data TAHUN_AKADEMIK.

c. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

d. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
 Use a Table or View : [dbo].[REF_TAHUN_AKADEMIK]

Karena tabel **REF_TAHUN_AKADEMIK** belum terbentuk pada *database*, maka buat tabel **REF_TAHUN_AKADEMIK** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

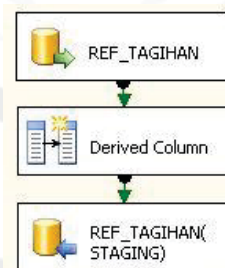
```
CREATE TABLE [REF_TAHUN_AKADEMIK] ([ID_TAHUN_AKADEMIK] INT,
[TAHUN_AKADEMIK] VARCHAR(10), [NO_SEMESTER] INT,
[SEMESTER_AKADEMIK] VARCHAR(30), [SEMESTER_AKADEMIK_ENG]
VARCHAR(30), [LOADED_DATE] DATETIME)
```

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.25. Mapping Data Flow Task REF_TAHUN_AKADEMIK – Full Refresh

13. REF_TAGIHAN



Gambar 5.26. Data Flow Task REF_TAGIHAN – Full Refresh

- a. Pilih *ADO.NET Source* karena data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : **DATA-WAREHOUSE.sikeuDB**
Data Access Mode : **Table or view**
Name of the table or view : **[dbo],[refTagihan]**

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

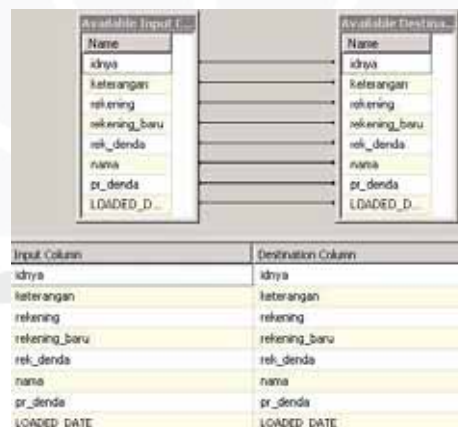
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[REF_TAGIHAN]

Karena tabel **REF_TAGIHAN** belum terbentuk pada *database*, maka buat tabel **REF_TAGIHAN** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

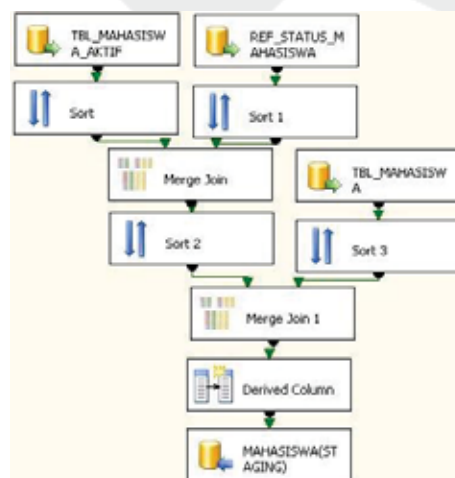
```
CREATE TABLE [REF_TAGIHAN] ([idnya] VARCHAR (3), [keterangan] VARCHAR(40), [rekening] VARCHAR(50), [rekening_ baru] VARCHAR(50), [rek_denda] VARCHAR(50), [nama] VARCHAR(50), [pr_denda] INT, [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.27. Mapping *Data Flow Task* REF_TAGIHAN – Full Refresh

14. TBL_MAHASISWA



Gambar 5.28. *Data Flow Task* TBL_MAHASISWA – Full Refresh

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) TBL_MAHASISWA

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

Name of the table or view : [dbo].[tbl_induk_mhs_3]

b) TBL_MAHASISWA_AKTIF

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

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

c) REF_STATUS_MAHASISWA

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

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

b. Pada *data flow task* TBL_MAHASISWA dan TBL_MAHASISWA_AKTIF digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan KD_STATUS_MHS yang telah diurutkan untuk mengambil data STATUS_MHS.

c. Pada *data flow task* REF_STATUS_MAHASISWA dan MERGE JOIN digabungkan dengan fungsi *merge join*. Masing-masing harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan NPM yang telah diurutkan untuk mengambil data ID_FAKULTAS, ID_PRODI, dan Nama_MHS.

d. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

e. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

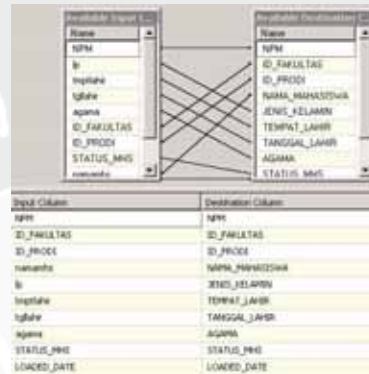
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo],[TBL_MAHASISWA]

Karena tabel TBL_MAHASISWA belum terbentuk pada *database*, maka buat tabel TBL_MAHASISWA dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [TBL_MAHASISWA] ([NPM] VARCHAR (20),
[ID_FAKULTAS] INT, [ID_PRODI] VARCHAR (2), [NAMA_MAHASISWA]
VARCHAR(255), [JENIS_KELAMIN] VARCHAR(255), [TEMPAT_LAHIR]
DATE TIME, [TANGGAL_LAHIR] DATETIME, [AGAMA] VARCHAR(255),
[LOADED_DATE] DATETIME)
```

- f. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.29. Mapping Data Flow Task TBL_MAHASISWA – Full Refresh

6). *Sequence Container* **LOAD MASTER TO STAGING SIKMA**

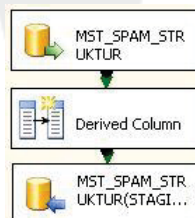
Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD MASTER TO STAGING SIKMA**.

7). *Data Flow Task* **LOAD MASTER TO STAGING SIKMA**

Tarik dan letakkan ikon *data flow task* ke dalam item **LOAD MASTER TO STAGING SIKMA** *sequence container* yang telah dibuat dan beri nama **MST_SPAM_STRUKTUR**. Lakukan hal yang sama untuk membuat *data flow task* **MST_BEASISWA_INDUK**.

8). *Data Flow Task* pada *Sequence Container* **LOAD MASTER TO STAGING SIKMA**

1. MST_SPAM_STRUKTUR



Gambar 5.30. Data Flow Task MST_SPAM_STRUKTUR – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax
Data Access Mode : Table or view
Name of the table or view : [dbo].[MST_SPAM_STRUKTUR]

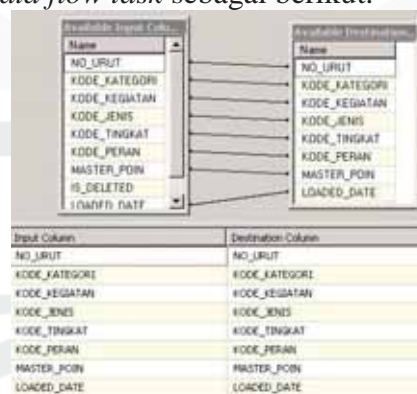
- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *derived column*. Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Use a Table or View : [dbo].[MST_SPAM_STRUKTUR]

Karena tabel **MST_SPAM_STRUKTUR** belum terbentuk pada *database*, maka buat tabel **MST_SPAM_STRUKTUR** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

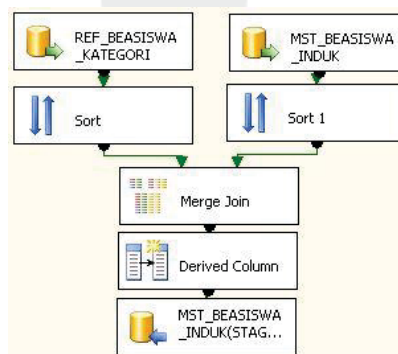
```
CREATE TABLE [MST_SPAM_STRUKTUR] ([NO_URUT] INT,
[KODE_KATEGORI] VARCHAR(1), [KODE_KEGIATAN] INT, [KODE_JENIS]
INT, [KODE_TINGKAT] INT, [KODE_PERAN] INT, [MASTER_POIN] INT,
[LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.31. Mapping *Data Flow Task* MST_SPAM_STRUKTUR – Full Refresh

2. MST_BEASISWA_INDUK



Gambar 5.32. *Data Flow Task* MST_BEASISWA_INDUK – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) REF_BEASISWA_KATEGORI

Connection Manager : DATA-WAREHOUSE.siatmax
Data Access Mode : Table or view
Name of table or view : [dbo].[REF_BEASISWA_KATEGORI]

b) MST_BEASISWA_INDUK

Connection Manager : DATA-WAREHOUSE.siatmax
Data Access Mode : Table or view
Name of table or view : [dbo].[MST_BEASISWA_INDUK]

- b. Pada *data flow task* **REF_BEASISWA_KATEGORI** dan **MST_BEASISWA_INDUK** digabungkan dengan fungsi *merge join*.

Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NO_KATEGORI** yang telah diurutkan untuk mengambil data **KETERANGAN KATEGORI**

- c. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

- d. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Use a Table or View : [dbo].[MST_BEASISWA_INDUK]

Karena tabel **MST_BEASISWA_INDUK** belum terbentuk pada *database*, maka buat tabel **MST_BEASISWA_INDUK** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [MST_BEASISWA_INDUK] ([NO_BEASISWA] INT, [KETERANGAN] VARCHAR (200), [NO_SPONSOR] INT, [NAMA_BEASISWA] VARCHAR (100), [TANGGAL_MULAI] DATETIME, [TANGGAL_SELESAI] DATETIME, [BESAR_BEASISWA] INT, [PERIODE] INT, [QUOTA] INT, [SYARAT_IPK] FLOAT, [SYARAT_KEMAHASISWAAN] BIT, [SYARAT_MIN_SEMESTER] INT, [SYARAT_SKS] INT, [SYARAT_EKONOMI] BIT, [SYARAT_BEASISWA] BIT, [SYARAT_LAIN] VARCHAR(500), [JUMLAH_TERMIN] INT, [FILE_NAME_FORMULIR] VARCHAR(80), [DESKRIPSI] VARCHAR(600), [TANGGAL_ENTRY]
```

DATETIME, [STATUS] VARCHAR(10), [FILE_ FORMULIR] IMAGE, [LOADED_DATE] DATETIME)

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.33. Mapping Data Flow Task MST_BEASISWA_INDUK – Full Refresh

9). **Execute SQL Task LOAD MASTER TO STAGING SIKMA**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD MASTER TO STAGING SIKMA** dan beri nama **INSERT REF_PRODI**. Ikon *Execute SQL Task* ini berfungsi menjalankan *query* untuk menambah data *ref_prodi* dan *ref_fakultas* yang ada pada *database SIKMA_STAGING*. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA_STAGING**).

Mendefinisikan *SQL Statement* pada item **LOAD MASTER TO STAGING SIKMA**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT REF_PRODI VALUES(25,7,NULL,NULL,NULL,NULL, NULL,NULL); INSERT INTO REF_FAKULTAS VALUES(7,NULL,NULL,NULL,NULL,NULL, NULL);
```

Sintaks di atas digunakan untuk menambah data *ref_prodi* dan *ref_fakultas* yang ada pada *database SIKMA_STAGING*.

10). **Sequence Container LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA**

Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA**.

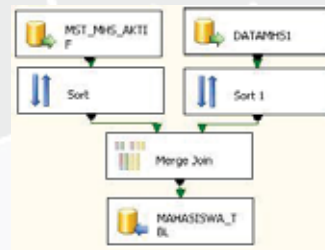
11). **Data Flow Task LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA**

Tarik dan letakkan ikon *data flow task* ke dalam item **LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA** yang telah dibuat

dan beri nama **TBL_SPAM_TRANSKIP**. Lakukan hal yang sama untuk membuat *data flow task mahasiswa_tbl*, **TBL_SPAM_DETAIL_TRANSKIP**, **TBL_PENERIMA_BEASISWA**, **TBL_INDUK_BEASISWA**, dan **DETAIL_TRANSKIP**.

12). *Data Flow Task* pada *Sequence Container* **LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA**

1. **mahasiswa_tbl**



Gambar 5.34. *Data Flow Task mahasiswa_tbl – Full Refresh*

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

1) MST_MHS_AKTIF

Connection Manager : DATA-WAREHOUSE.siatma
Data Access Mode : Table or view
Name of table or view : [dbo].[MST_MHS_AKTIF]

2) DATAMHS1

Connection Manager : DATA-WAREHOUSE.siatma
Data Access Mode : Table or view
Name of table or view : [dbo].[datamhs1]

b. Pada *data flow task* **MST_MHS_AKTIF** dan **DATAMHS1** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus di urutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **THN_MASUK** dan **TGL_LAHIR**

c. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

d. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

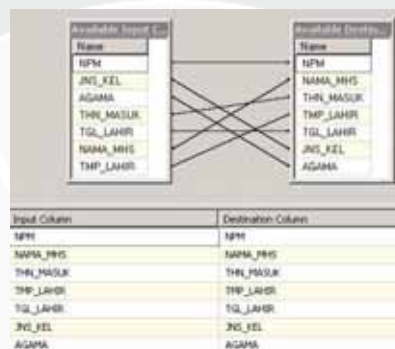
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View :[dbo].[MAHASISWA_TBL]

Karena tabel **MAHASISWA_TBL** belum terbentuk pada *database*, maka buat tabel **MAHASISWA_TBL** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

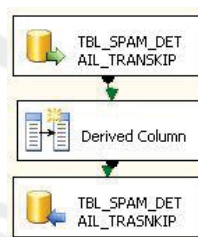
```
CREATE TABLE [MAHASISWA_TBL] ([NPM] VARCHAR(20), [NAMA_MHS]
VARCHAR (150), [THN_MASUK] NUMERIC(4,0), [TMP_LAHIR] NVARCHAR
(70), [TGL_LAHIR] DATETIME, [JNS_KEL] VARCHAR(1), [AGAMA]
VARCHAR(15))
```

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.35. Mapping Data Flow Task MAHASISWA_TBL – Full Refresh

2. TBL_SPAM_DETAIL_TRANSKIP



Gambar 5.36. Data Flow Task TBL_SPAM_DETAIL_TRANSKIP – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax

Data Access Mode : Table or view

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

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Use a Table or View : [dbo].[TBL_SPAM_DETAIL_TRANSKIP]

Karena tabel **TBL_SPAM_DETAIL_TRANSKIP** belum terbentuk pada *database*, maka buat tabel **TBL_SPAM_DETAIL_TRANSKIP** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

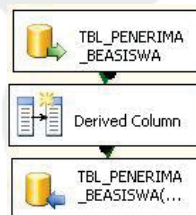
```
CREATE TABLE [TBL_SPAM_DETAIL_TRANSKIP] ([NO_URUT_DETAIL] INT,
[ID_TRANSKIP_SPAM] INT, [KODE_KATEGORI] VARCHAR(1),
[KODE_KEGIATAN] INT, [KODE_JENIS] INT, [KODE_TINGKAT] INT,
[KODE_PERAN] INT, [NAMA_KEGIATAN] VARCHAR(MAX), [DES KRIPSI]
VARCHAR (MAX), [TGL_MULAI] DATETIME, [TGL_SELESAI] DATETIME,
[PENYELENGGARA] VARCHAR(500), [TEMPAT] VARCHAR (500),
[ID_TAHUN_AKADEMIK] INT, [NO_SEMESTER] INT, [POIN] INT,
[NO_URUT_UPLOAD] INT, [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.37. Mapping Data Flow Task MAHASISWA_TBL – Full Refresh

3. TBL_PENERIMA_BEASISWA



Gambar 5.38. Data Flow Task TBL_PENERIMA_BEASISWA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax
Data Access Mode : Table or view
Name of the table or view : [dbo].[TBL_PENERIMA_BEASISWA]

b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

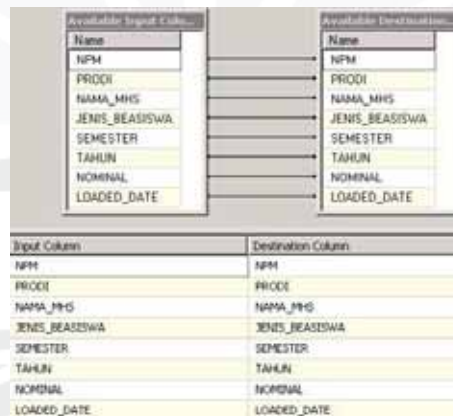
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[TBL_PENERIMA_BEASISWA]

Karena tabel **TBL_PENERIMA_BEASISWA** belum terbentuk pada *database*, maka buat tabel **TBL_PENERIMA_BEASISWA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

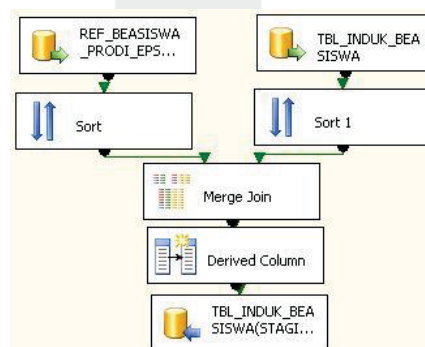
```
CREATE TABLE [TBL_PENERIMA_BEASISWA] ([NPM] VARCHAR(20),
[PRODI] VARCHAR(40), [NAMA_MHS] VARCHAR(400), [JENIS_BEASISWA]
VARCHAR(400), [SEMESTER] VARCHAR (40), [TAHUN] INT, [NOMINAL] INT,
[LOADED_DATE] DATETIME)
```

d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.39. Mapping *Data Flow Task* TBL_PENERIMA_BEASISWA – Full Refresh

4. TBL_INDUK_BEASISWA



Gambar 5.40. *Data Flow Task* TBL_INDUK_BEASISWA – Full Refresh

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) REF_BEASISWA_PRODI_EPSBED

Connection Manager : DATA-WAREHOUSE.siatmax
Data Access Mode : Table or view
Name of table or view : [dbo].[REF_BEASISWA_PRODI_EPSBED]

b) TBL_INDUK_BEASISWA

Connection Manager : DATA-WAREHOUSE.siatmax
Data Access Mode : Table or view
Name of table or view : [dbo].[TBL_INDUK_BEASISWA]

b. Pada *data flow task* **REF_BEASISWA_PRODI_EPSBED** dan **TBL_INDUK_BEASISWA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus di urutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID_PRODI** yang telah diurutkan untuk mengambil data **ID_PRODI_EPSBED**

c. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

d. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Use a Table or View : [dbo].[TBL_INDUK_BEASISWA]

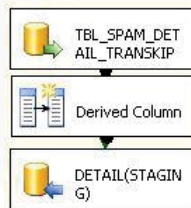
Karena tabel **TBL_INDUK_BEASISWA** belum terbentuk pada *database*, maka buat tabel **TBL_INDUK_BEASISWA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [TBL_INDUK_BEASISWA] ([NPM] VARCHAR(20), [KDPTI]
VARCHAR (6), [ID_JENIS_BEASISWA] INT, [COUNTER] INT, [NAMA_MHS]
NTEXT, [ID_JENIS_KELAMIN] INT, [ID_PRODI_ EPSBED] NVARCHAR(6),
[ID_JENJANG] INT, [SEMESTER] INT, [IPK] NUMERIC(3,2), [ID_PKERJAAN]
INT, [JML_TANGGUNGAN] INT, [PENGHASILAN] NUMERIC(38,19),
[PRESTASI] NTEXT, [MU LAI_BULAN] DATETIME, [SELESAI_BULAN]
DATETIME, [TAHUN] SMALLINT, [KETERANGAN] NTEXT, [ALAMAT]
NTEXT, [TELEPON] NTEXT, [NAMA_PEMILIK] NVARCHAR(300),
[NAMA_BANK] NVARCHAR (300), [CABANG] NVARCHAR(300),
[NOREKENING] NVARCHAR(50), [STATUS] NTEXT, [TGL_UPLOAD]
DATETIME, [LOADED_DATE] DATETIME)
```

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.41. Mapping *Data Flow Task* TBL_INDUK_BEASISWA – Full Refresh
5. DETAIL_TRANSKIP



Gambar 5.42. *Data Flow Task* DETAIL_TRANSKIP – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax
Data Access Mode : Table or view
Name of the table or view : [dbo].[TBL_SPAM_DETAIL_TRANSKIP]

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *derived column*.
Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Use a Table or View : [dbo].[DETAIL]

Karena tabel **DETAIL** belum terbentuk pada *database*, maka buat tabel **DETAIL** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DETAIL] ([NO_URUT_DETAIL] INT, [ID_TRANSKIP_ SPAM]
INT,[KODE_KATEGORI] NVARCHAR(1),[KODE_KEGIATAN] INT,
```


[KODE_JENIS] INT, [KODE_TINGKAT] INT, [KODE_PERAN] INT, [NAMA_KEGIATAN] NTEXT, [DESKRIPSI] NTEXT, [TGL_MULA] DATETIME, [TGL_SELESAI] DATETIME, [PENYELENGGARA] NVARCHAR (500), [TEMPAT] NVARCHAR(500), [ID_TAHUN_AKADEMIK] INT, [NO_SEMESTER] INT, [POIN] INT, [NO_URUT_UPLOAD] INT, [LOADED_DATE] DATETIME)

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



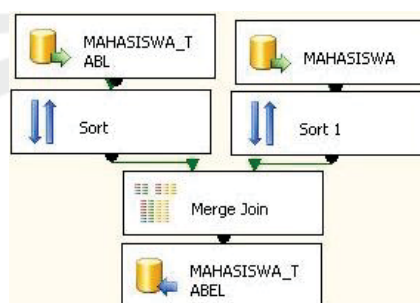
Gambar 5.43. Mapping *Data Flow Task* **DETAIL** – *Full Refresh*

13). *Sequence Container* **LOAD1**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD1**.

14). *Data Flow Task* **LOAD1**

Tarik dan letakkan ikon *Data Flow Task* ke dalam item **LOAD1** *Sequence Container* yang telah dibuat sebelumnya dan beri nama **MAHASISWA_TABEL**.



Gambar 5.44. *Data Flow Task* **MAHASISWA_TABEL** – *Full Refresh*

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) MAHASISWA_TABEL

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
 Data Access Mode : Table or view
 Name of table or view : [dbo].[Mahasiswa_tabl]

b) MAHASISWA

Connection Manager : DATA-WAREHOUSE.biadw

Data Access Mode : Table or view

Name of table or view : [dbo].[mahasiswa]

- b. Pada *data flow task* **MAHASISWA_TABL** dan **MAHASISWA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus di urutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **ID_KAB_ORTU** dan **ID_SMA**.

- c. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

- d. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

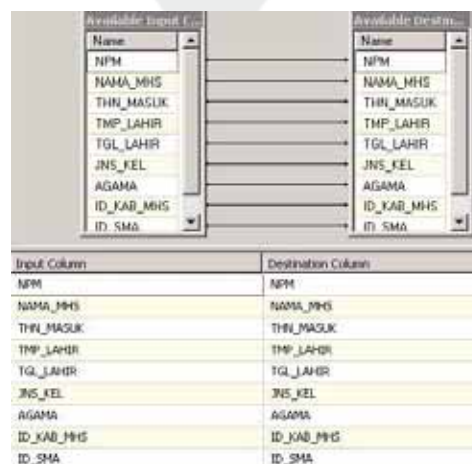
Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Use a Table or View : [dbo].[MAHASISWA_TABEL]

Karena tabel **MAHASISWA_TABEL** belum terbentuk pada *database*, maka buat tabel **MAHASISWA_TABEL** dengan menekan tombol *New* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [MAHASISWA_TABEL] ([NPM] VARCHAR(20), [NAMA_MHS] VARCHAR (150), [THN_MASUK] INT, [TMP_LAHIR] VARCHAR(255), [TGL_LAHIR] DATETIME, [JNS_KEL] VARCHAR(1), [AGAMA] VARCHAR(15), [ID_KAB_MHS] INT, [ID_SMA] INT)
```

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.45. Mapping Data Flow Task MAHASISWA_TABEL – Full Refresh

15). *Execute SQL Task* pada **LOAD1**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD1** Sequence Container sebanyak 7 kali dan beri nama **INSERT_JENIS_BEASISWA**, **UPDATE_NAMA_PROPINSI**, **UPDATE_SEMESTER_AKADEMIK**, **DELETE_TARIF**, **INSERT REF_KAB_KODYA**, **UPDATE_SEMESTER**, dan **UPDATE_DETAIL**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA_STAGING**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

- a. Ikon *Execute SQL Task* **INSERT_JENIS_BEASISWA** berfungsi menjalankan *query* untuk menambah data jenis beasiswa yang ada pada *database* **SIKMA_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT INTO REF_JENIS_BEASISWA VALUES(3,'SPP 50% Tetap', NULL); INSERT
INTO REF_JENIS_BEASISWA VALUES(4,'SPP Tetap', NULL); INSERT INTO
REF_JENIS_BEASISWA VALUES(5,'Mandiri', NULL); INSERT INTO
REF_JENIS_BEASISWA VALUES (6,'Djarum', NULL);INSERT INTO
REF_JENIS_BEASISWA VALUES(7,'BRI',NULL); INSERT INTO REF_JENIS_
BEASISWA VALUES(8,'Toyota',NULL); INSERT INTO REF_JENIS_BEASISWA
VALUES(9,'Vivere',NULL); INSERT INTO REF_JENIS_BEASISWA VALUES
(10,'SWA',NULL); INSERT INTO REF_JENIS_BEASISWA
VALUES(11,'YPKAAR',NULL)
```

- b. Ikon *Execute SQL Task* **UPDATE_NAMA_PROPINSI** berfungsi untuk mengubah data nama propinsi yang ada pada *database* **SIKMA_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE REF_KAB_KODYA SET NAMA_PROPINSI='Nanggroe Aceh Darusalam'
WHERE ID_KAB_KODYA IN(SELECT ID_KAB_KODYA WHERE
NAMA_PROPINSI='NANGROE ACEH DARUSALAM'); UPDATE
REF_KAB_KODYA SET NAMA_PROPINSI='Bangka Belitung' WHERE
ID_KAB_KODYA IN( SELECT ID_KAB_KODYA WHERE NAMA_PROPINSI=
'KEPULAUAN BANGKA BELITUNG');
```

- c. Ikon *Execute SQL Task* **UPDATE_SEMESTER_AKADEMIK** berfungsi untuk mengubah data semester yang ada pada *database* **SIKMA_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE REF_WAKTU_PRESTASI SET SEMESTER=1 WHERE DateKey IN(
SELECT DateKey FROM REF_WAKTU_ PRESTASI WHERE QuarterOfYear IN(1,2));
UPDATE REF_WAKTU_PRESTASI SET SEMESTER=2 WHERE DateKey
IN(SELECT DateKey FROM REF_WAKTU_PRESTASI WHERE QuarterOfYear
IN(3,4));UPDATE REF_WAKTU_PRESTASI SET SEMESTER _AKADEMIK=2
WHERE DateKey IN(SELECT DateKey FROM REF_WAKTU_ PRESTASI WHERE
Quarter OfYear IN(1,2));UPDATE REF_WAKTU_PRES TASI SET
SEMESTER_AKADEMIK=1 WHERE DateKey IN(SELECT DateKey FROM
REF_WAKTU_PRESTASI WHERE Quarter OfYear IN(3,4));
```

- d. Ikon *Execute SQL Task* **DELETE_TARIF** berfungsi untuk menghapus data tarif selain id tarif 10 yang ada pada *database SIKMA_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
DELETE FROM REF_TARIF WHERE tagihan !=10
```

- e. Ikon *Execute SQL Task* **INSERT_REF_KAB_KODYA** berfungsi untuk menambah data kabupaten dan kodya yang ada pada *database SIKMA_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT INTO REF_KAB_KODYA VALUES(533,NULL,NULL, NULL);INSERT INTO
REF_KAB_KODYA VALUES(475,NULL, NULL,NULL);INSERT INTO
REF_KAB_KODYA VALUES(130, NULL,NULL,NULL);INSERT INTO
REF_KAB_KODYA VALUES(474,NULL,NULL,NULL);
```

- f. Ikon *Execute SQL Task* **UPDATE_SEMESTER** berfungsi untuk mengubah data semester yang ada pada *database SIKMA_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE REF_WAKTU SET SEMESTER='Ganjil' WHERE ID_WAKTU IN( SELECT
ID_WAKTU WHERE SEMESTER= 'Gasal')
```

- g. Ikon *Execute SQL Task* **UPDATE_DETAIL** berfungsi untuk mengubah data detail transkip spama yang ada pada *database SIKMA_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE TBL_SPAM_DETAIL_TRANSKIP SET NAMA_KEGIATAN= 'LDKPM_
UAJY' WHERE NO_URUT_DETAIL IN (SELECT NO_URUT_DETAIL FROM
TBL_SPAM_DETAIL_TRANSKIP WHERE NAMA_KEGIATAN= 'LDPKM');UPDATE
TBL_SPAM_DETAIL_TRANSKIP SET NAMA_KEGIATAN= 'INISIASI_UAJY'
WHERE NO_URUT_DETAIL IN(SELECT NO_URUT_DETAIL FROM TBL_SPAM_
DETAIL_TRANSKIP WHERE NAMA_KEGIATAN= 'INISIASI ');UPDATE TBL_
SPAM_DETAIL_TRANSKIP SET NAMA_KEGIATAN= 'SEMI NAR INTEGRITAS
```

```
ENTREPRENEURSHIP HMPSM UAJY' WHERE NO_URUT_DETAIL IN(SELECT
NO_URUT_DETAIL FROM TBL_SPAM_DETAIL_TRANS KIP WHERE NAMA_
KEGIATAN LIKE'SEMINAR INTEGRITAS ENT%');UPDATE TBL_SPAM_
DETAIL_TRANSKIP SET DESKRIPSI= 'GREEN_ACTION' WHERE NO_URUT_
DETAIL IN(SELECT NO_URUT_DETAIL FROM TBL_SPAM_DETAIL_
TRANSKIP WHERE NAMA_KEGIATAN LIKE'GREEN ACT%');
```

16). *Sequence Container* **LOAD2**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD2**.

17). *Execute SQL Task* **LOAD2**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD2** sebanyak 2 kali dan beri nama **UPDATE_AGAMA** dan **UPDATE_FAKULTAS**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA_STAGING**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

a. Ikon *Execute SQL Task* **UPDATE_AGAMA** berfungsi untuk mengubah data agama yang ada pada *database* **SIKMA_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE MAHASISWA_TABEL SET AGAMA='BUDDHISM' WHERE NPM
IN(SELECT NPM FROM MAHASISWA_TABEL WHERE AGAMA=
'Budha');UPDATE MAHASISWA_TABEL SET AGAMA='CHRISTIAN' WHERE
NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE AGAMA=
'Protestan');UPDATE MAHASISWA_TABEL SET AGAMA='CATHOLIK' WHERE
NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE AGAMA=
'Katolik');UPDATE MAHASISWA_TABEL SET AGAMA='MOHAMMEDANISM'
WHERE NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE
AGAMA='Islam');UPDATE MAHASISWA_TABEL SET AGAMA='HINDUISM'
WHERE NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE
AGAMA='Hindu');UPDATE MAHASISWA_TABEL SET AGAMA='N/A' WHERE
NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE AGAMA IS NULL);
```

b. Ikon *Execute SQL Task* **UPDATE_FAKULTAS** berfungsi untuk mengubah data fakultas yang ada pada *database* **SIKMA_STAGING**.

Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE TBL_MAHASISWA SET ID_FAKULTAS=5 WHERE ID_PRODI
IN(01,02,13);UPDATE TBL_MAHASISWA SET ID_FAKULTAS=1 WHERE
ID_PRODI IN(03,04,11,12,15, 20);UPDATE TBL_MAHASISWA SET
ID_FAKULTAS=2 WHERE ID_PRODI IN(05,21);UPDATE TBL_MAHASISWA SET
ID_FAKULTAS=3 WHERE ID_PRODI =08;UPDATE TBL_MAHASISWA SET
```

```
ID_FAKULTAS=4 WHERE ID_PRODI IN(06,07,14,24);UPDATE
TBL_MAHASISWA SET ID_FAKULTAS=6 WHERE ID_PRODI IN(09,10);
```

18). *Sequence Container* **LOAD3**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD3**.

19). *Execute SQL Task* **LOAD3**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD2** sebanyak 3 kali dan beri nama **UPDATE_NOMINAL_BEASISWA**, **UPDATE_PRODI_MAHASISWA**, dan **UPDATE_MAHASISWA**. Pilih koneksi tersebut (**DATA-WAREHOUSE. SIKMA_STAGING**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

- a. Ikon *Execute SQL Task* **UPDATE_NOMINAL_BEASISWA** berfungsi untuk mengubah data nominal beasiswa yang ada pada *database SIKMA_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE PENERIMA_BEASISWA SET NOMINAL=BIAYA_SPP WHERE
JENIS_BEASISWA='SPP Tetap' and NPM IN(SELECT NPM FROM
PENERIMA_BEASISWA WHERE JENIS_ BEASISWA='SPP Tetap');UPDATE
PENERIMA_BEASISWA SET NOMINAL=(0.5*BIAYA_SPP) WHERE JENIS_
BEASISWA ='SPP 50% Tetap' and NPM IN(SELECT NPM FROM
PENERIMA_BEASISWA WHERE JENIS_BEASISWA='SPP 50% Tetap');
```

- b. Ikon *Execute SQL Task* **UPDATE_PRODI_MAHASISWA** berfungsi untuk mengubah data prodi mahasiswa yang ada pada *database SIKMA_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE TBL_MAHASISWA SET ID_PRODI = 25 WHERE ID_PRODI IS NULL;
UPDATE TBL_MAHASISWA SET ID_FAKULTAS=7 WHERE ID_FAKULTAS IS
NULL;
```

- c. Ikon *Execute SQL Task* **UPDATE_MAHASISWA** berfungsi untuk mengubah data mahasiswa yang ada pada *database SIKMA_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE MAHASISWA_TABEL SET NAMA_MHS='N/A' WHERE (NPM IS NULL)
OR (NAMA_MHS IS NULL); UPDATE MAHASISWA_TABEL SET TMP_
LAHIR='N/A' WHERE (TMP_LAHIR IS NULL); UPDATE MAHASISWA_TABEL SET
JNS_KEL='-' WHERE (JNS_KEL IS NULL); UPDATE MAHASISWA_TABEL SET
ID_KAB_MHS=1 WHERE (ID_KAB_MHS IS NULL);
```


20). *Sequence Container* **LOAD4**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD4**.

21). *Execute SQL Task* **LOAD4**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD4** dan beri nama dengan **UPDATE_SMA_MAHASISWA**. Ikon *Execute SQL Task* ini berfungsi untuk mengubah data sma yang ada pada *database SIKMA_STAGING*. Pilih koneksi pada *Execute SQL Task* tersebut (**DATA-WAREHOUSE.SIKMA_STAGING**). Lakukan konfigurasi dengan menuliskan sintaks berikut:

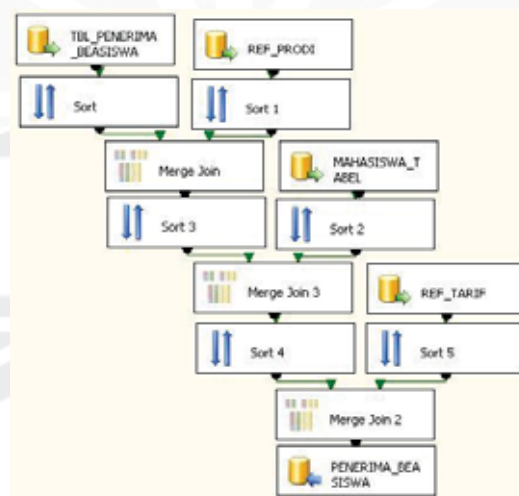
```
UPDATE MAHASISWA_TABEL SET ID_SMA=1 WHERE ID_SMA IS NULL;UPDATE
MAHASISWA_TABEL SET ID_SMA=1 WHERE ID_SMA=1;
```

22). *Sequence Container* **LOAD5**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD5**.

23). *Data Flow Task* pada **LOAD5**

Tarik dan letakkan ikon *data flow task* ke dalam item **LOAD5** dan beri nama dengan **PENERIMA_BEASISWA**.



Gambar 5.46. Data Flow Task PENERIMA_BEASISWA – Full Refresh

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) TBL_PENERIMA_BEASISWA

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Data Access Mode : Table or view
Name of table or view : [dbo].[TBL_PENERIMA_BEASISWA]

b) REF_PRODI

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Data Access Mode : Table or view
Name of table or view : [dbo].[REF_PRODI]

c) MAHASISWA_TABEL

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Data Access Mode : Table or view
Name of table or view : [dbo].[MAHASISWA_TABEL]

d) REF_TARIF

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Data Access Mode : Table or view
Name of table or view : [dbo].[REF_TARIF]

- b. Pada *data flow task* **TBL_PENERIMA_BEASISWA** dan **REF_PRODI** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **PRODI** yang telah diurutkan untuk mengambil data **ID_PRODI**.
- c. Pada *data flow task* **MAHASISWA_TABEL** dan hasil *merge* sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **THN_MASUK**.
- d. Pada *data flow task* **REF_TARIF** dan hasil *merge* sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID_PRODI** dan **THN_MASUK** yang telah diurutkan untuk mengambil data **biaya**.
- e. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Use a Table or View : [dbo].[PENERIMA_BEASISWA]

Karena tabel **PENERIMA_BEASISWA** belum terbentuk pada *database*, maka buat tabel **PENERIMA_BEASISWA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [PENERIMA_BEASISWA] ([NPM] VARCHAR(20), [PRODI]
VARCHAR(40), [NAMA_MHS] VARCHAR(400), [JENIS_BEASISWA]
VARCHAR(400), [SEMESTER] VARCHAR(40), [TAHUN] INT, [NOMINAL] INT,
```

[LOADED_DATE] DATETIME, [ID_PRODI] VARCHAR(2), [THN_MASUK] INT, [BIAYA_SPP] INT)

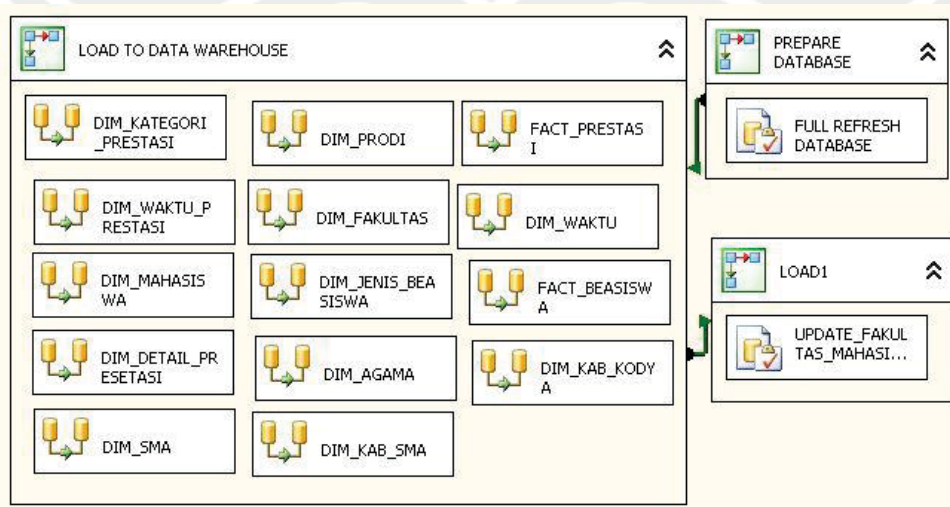
- f. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.47. Mapping *Data Flow Task* PENERIMA_BEASISWA – Full Refresh

5.1.2.2 Loading Data dari SIKMA_STAGING ke SIKMA_DW

Dalam tahap ini, data pada *database* SIKMA_STAGING akan di-load ke dalam *data warehouse* yaitu SIKMA_DW. Pada proses *loading* data dari *database* SIKMA_STAGING ke dalam *database* SIKMA_DW, langkah pertama adalah menentukan *control flow items* yang sesuai dengan kebutuhan pada *tab Control Flow*. Proses *loading* data dari SIKMA_STAGING ke SIKMA_DW pada *package* Warehouse_sikma.dtsx akan dibuat desain proses sebagai berikut :



Gambar 5.48. Desain *Control Flow* pada Warehouse_sikma – Full Refresh

Untuk membuat desain *Control Flow* di atas, langkah-langkah yang dilakukan adalah sebagai berikut:

1). *Sequence Container* **Prepare Database**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **PREPARE DATABASE**.

2). *Execute SQL Task* **FULL_REFRESH_DATABASE**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **PREPARE DATABASE** *sequence container* yang telah dibuat dan beri nama **FULL_REFRESH_DATABASE**. Ikon *Execute SQL Task* ini berfungsi menjalankan *query* untuk menghapus semua isi tabel yang ada pada *database* **SIKMA_DW**. Hal ini dilakukan agar isi tabel pada *database* **SIKMA_DW** selalu baru. Buat koneksi ke *database* pada *server* **DATA-WAREHOUSE**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA_DW**). Langkah-langkah pembuatan koneksi **DATA-WAREHOUSE.SIKMA_DW** dapat diamati pada lampiran tugas akhir ini. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
DELETE FROM DIM_AGAMA;DELETE FROM FACT_BEASISWA; DELETE FROM
DIM_SMA; DELETE FROM FACT_PRESTASI; DELETE FROM DIM_KATE
GORI_PRESTASI;DELETE FROM DIM_JENIS_BEASISWA;DELETE FROM DIM_
MAHASISWA;DELETE FROM DIM_PRODI;DELETE FROM DIM_FAKULTAS;DELETE
FROM DIM_WAKTU;DELETE FROM DIM_KAB_KODYA;DELETE FROM DIM_KAB_
SMA;DELETE FROM DIM_WAKTU_PRESTASI;DELETE FROM DIM_DETAIL_
PRESTASI;
```

3). *Sequence Container* **LOAD TO DATA WAREHOUSE**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD TO DATA WAREHOUSE**.

4). *Data Flow Task* **LOAD TO DATA WAREHOUSE**

Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* tersebut. Karena ada 14 data sumber yang akan di-load maka dibuat 14 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **DIM_KATEGORI_PRESTASI, DIM_WAKTU_PRESTASI, DIM_MAHASISWA, DIM_DETAIL_PRESTASI, DIM_SMA, DIM_PRODI, DIM_FAKULTAS, DIM_JENIS_BEASISWA, DIM_AGAMA, DIM_**

KAB_SMA, FACT_PRESTASI, DIM_WAKTU, FACT_BEASISWA, dan DIM_KAB_KODYA.

Hubungkan **PREPARE DATABASE** *sequence container* dengan *sequence container* **LOAD TO DATA WAREHOUSE**.

5). *Data Flow Task* pada *Sequence Container* **LOAD TO DATA WAREHOUSE**

Langkah-langkah untuk membuat *data flow task* **LOAD TO DATA WAREHOUSE** sebagai berikut:

1. **DIM_JENIS_BEASISWA**



Gambar 5.49. Data Flow Task DIM_JENIS_BEASISWA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon **ADO.NET SOURCE** dan lakukan konfigurasi

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Data Access Mode : Table or view

Name of the table or view : [dbo],[REF_JENIS_BEASISWA]

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan **ADO.NET DESTINATION** dan dihubungkan dengan **ADO.NET.SOURCE**. Lakukan konfigurasi berikut:

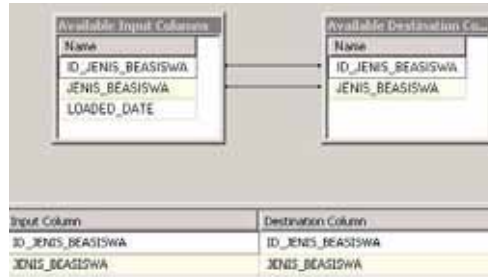
Connection Manager : DATA-WAREHOUSE.SIKMA_DW

Use a Table or View : [dbo],[DIM_JENIS_BEASISWA]

Karena tabel **DIM_JENIS_BEASISWA** belum terbentuk pada *database*, maka buat tabel **DIM_JENIS_BEASISWA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_JENIS_BEASISWA] ([ID_JENIS_BEASISWA] INT,
[JENIS_BEASISWA] VARCHAR(60), CONSTRAINT PK_JENIS PRIMARY
KEY(ID_JENIS_BEASISWA))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.50. Mapping Data Flow Task DIM_JENIS_BEASISWA – Full Refresh

2. DIM_WAKTU



Gambar 5.51. Data Flow Task DIM_WAKTU – Full Refresh

- a. Data yang akan di-load bersumber dari server, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

ADO.NET Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Data Access Mode : Table or view
Name of the table or view : [dbo].[REF_WAKTU]

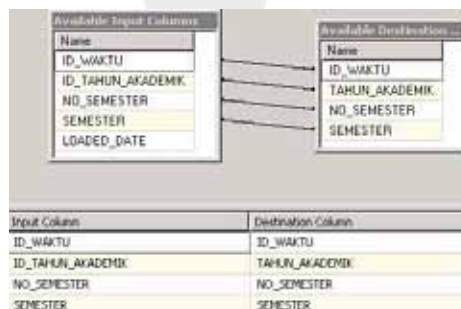
- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_DW
Use a Table or View : [dbo].[DIM_WAKTU]

Karena tabel **DIM_WAKTU** belum terbentuk pada *database*, maka buat tabel **DIM_WAKTU** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

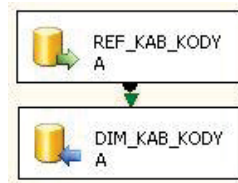
```
CREATE TABLE [DIM_WAKTU] ([ID_WAKTU] INT, [TAHUN_AKADEMIK] INT,
[NO_SEMESTER] INT, [SEMESTER] VARCHAR(7), CONSTRAINT PK_WAKTU
PRIMARY KEY(ID_WAKTU))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.52. Mapping Data Flow Task DIM_WAKTU – Full Refresh

3. DIM_KAB_KODYA



Gambar 5.53. Data Flow Task DIM_KAB_KODYA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

ADO.NET Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Data Access Mode : Table or view

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

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

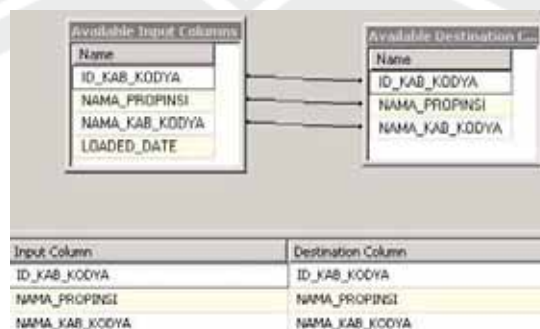
Connection Manager : DATA-WAREHOUSE.SIKMA_DW

Use a Table or View : [dbo].[DIM_KAB_KODYA]

Karena tabel **DIM_KAB_KODYA** belum terbentuk pada *database*, maka buat tabel **DIM_KAB_KODYA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_KAB_KODYA] ([ID_KAB_KODYA] INT,
[NAMA_PROPINSI] VARCHAR(150), [NAMA_KAB_KODYA] VARCHAR(150),
CONSTRAINT PK_KAB PRIMARY KEY(ID_KAB_KODYA))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.54. Mapping Data Flow Task DIM_KAB_KODYA – Full Refresh

4. DIM_KAB_SMA



Gambar 5.55. Data Flow Task DIM_KAB_KODYA – Full Refresh

- a. Data yang akan di-load bersumber dari server, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

ADO.NET Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Data Access Mode : Table or view
Name of the table or view : [dbo].[REF_KAB_KODYA]

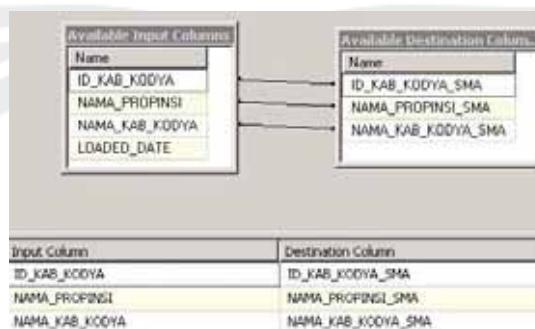
- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_DW
Use a Table or View : [dbo].[DIM_KAB_SMA]

Karena tabel **DIM_KAB_SMA** belum terbentuk pada *database*, maka buat tabel **DIM_KAB_SMA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_KAB_SMA] ([ID_KAB_KODYA_SMA] INT, [NAMA_PROPINSI_SMA] VARCHAR(150), [NAMA_KAB_KODYA_SMA] VARCHAR (150), CONSTRAINT PK_KAB_SMA PRIMARY KEY(ID_KAB_KODYA_SMA))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.56. Mapping Data Flow Task DIM_KAB_SMA – Full Refresh

5. DIM_FAKULTAS



Gambar 5.57. Data Flow Task DIM_FAKULTAS – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

ADO.NET Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Data Access Mode : Table or view

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

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

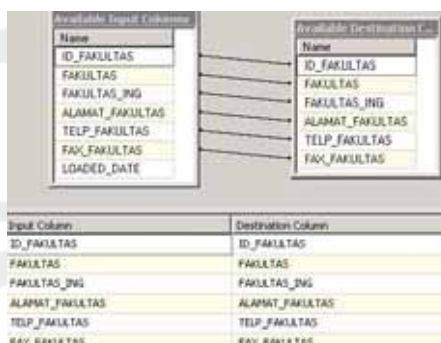
Connection Manager : DATA-WAREHOUSE.SIKMA_DW

Use a Table or View : [dbo].[DIM_FAKULTAS]

Karena tabel **DIM_FAKULTAS** belum terbentuk pada *database*, maka buat tabel **DIM_FAKULTAS** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_FAKULTAS] ([ID_FAKULTAS] INT, [FAKULTAS]
VARCHAR(30), [FAKULTAS_ING] VARCHAR(30), [ALAMAT_FAKULTAS]
VARCHAR(80), [TELP_FAKULTAS] VARCHAR(12), [FAX_FAKULTAS]
VARCHAR(12), CONSTRAINT PK_FAKUL PRIMARY KEY(ID_FAKULTAS))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.58. Mapping Data Flow Task DIM_FAKULTAS – Full Refresh

6. DIM_PRODI



Gambar 5.59. Data Flow Task DIM_PRODI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

ADO.NET Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Data Access Mode : Table or view

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

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

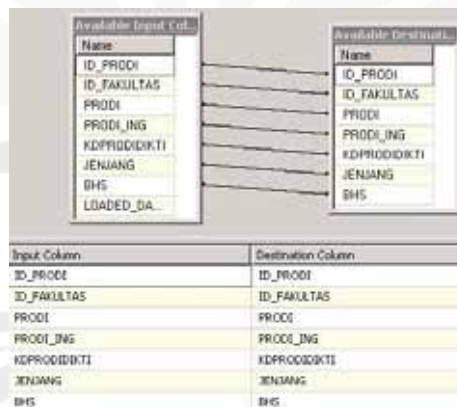
Connection Manager : DATA-WAREHOUSE.SIKMA_DW

Use a Table or View : [dbo].[DIM_PRODI]

Karena tabel **DIM_PRODI** belum terbentuk pada *database*, maka buat tabel **DIM_PRODI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

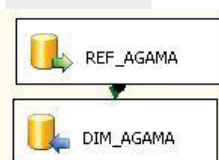
```
CREATE TABLE [DIM_PRODI] ([ID_PRODI] VARCHAR(2), [ID_FAKULTAS] INT, [PRODI] VARCHAR(50), [PRODI_ING] VARCHAR(50), [KD_PRODIDIKTI] VARCHAR(5), [JENJANG] VARCHAR(2), [BHS] VARCHAR(1), CONSTRAINT PK_PRODI PRIMARY KEY (ID_PRODI), CONSTRAINT FK_PRODI_FAKUL FOREIGN KEY(ID_FAKULTAS)REFERENCES DIM_FAKULTAS(ID_FAKULTAS))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.60. Mapping Data Flow Task DIM_PRODI – Full Refresh

7. DIM_AGAMA



Gambar 5.61. Data Flow Task DIM_AGAMA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Data Access Mode : Table or view

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

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

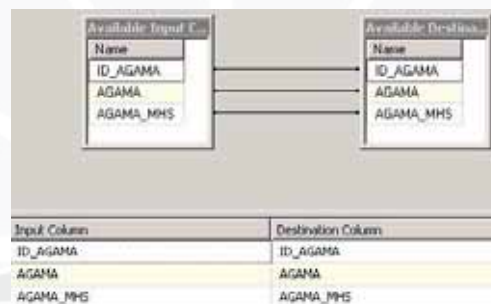
Connection Manager : DATA-WAREHOUSE.SIKMA_DW

Use a Table or View : [dbo].[DIM_AGAMA]

Karena tabel **DIM_AGAMA** belum terbentuk pada *database*, maka buat tabel **DIM_AGAMA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

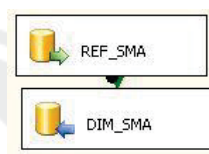
```
CREATE TABLE [DIM_AGAMA] ([ID_AGAMA] INT, [AGAMA] VARCHAR(15), [AGAMA_MHS] VARCHAR(15), CONSTRAINT PK_AGAMA PRIMARY KEY (ID_AGAMA))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.62. Mapping Data Flow Task DIM_AGAMA – Full Refresh

8. DIM_SMA



Gambar 5.63. Data Flow Task DIM_SMA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Data Access Mode : Table or view

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

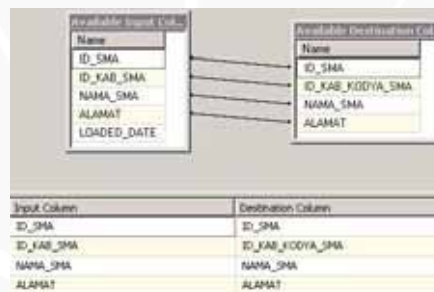
- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_DW
Use a Table or View : [dbo].[DIM_SMA]

Karena tabel **DIM_SMA** belum terbentuk pada *database*, maka buat tabel **DIM_SMA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

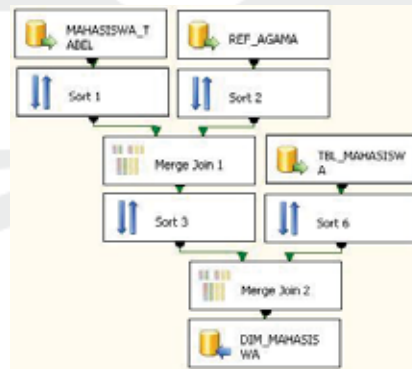
```
CREATE TABLE [DIM_SMA] ([ID_SMA] INT, [ID_KAB_KODYA_SMA] INT,
[NAMA_SMA] VARCHAR(200), [ALAMAT] VARCHAR(150), CONSTRAINT
PK_SMA PRIMARY KEY (ID_SMA), CONSTRAINT FK_SMA_KAB FOREIGN
KEY(ID_KAB_KODYA_SMA) REFERENCES DIM_KAB_SMA (ID_KAB_KODYA_
SMA))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.64. Mapping *Data Flow Task* DIM_SMA – Full Refresh

9. DIM_MAHASISWA



Gambar 5.65. *Data Flow Task* DIM_MAHASISWA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

- a) MAHASISWA_TABEL

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Data Access Mode : Table or view

Name of table or view : [dbo].[MAHASISWA_TABEL]

- b) REF_AGAMA

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Data Access Mode : Table or view
Name of table or view : [dbo].[REF_AGAMA]

c) TBL_MAHASISWA

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
Data Access Mode : Table or view
Name of table or view : [dbo].[TBL_MAHASISWA]

b. Pada *data flow task* **MAHASISWA_TABEL** dan **REF_AGAMA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **AGAMA** yang telah diurutkan untuk mengambil data **ID_AGAMA**.

c. Pada *data flow task* **TBL_MAHASISWA** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **ID_FAKULTAS** dan **ID_PRODI**.

d. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan lakukan konfigurasi berikut:

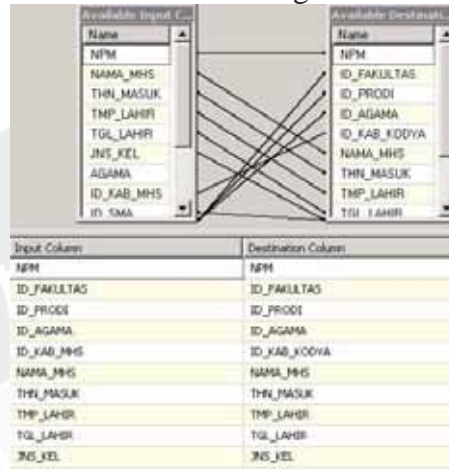
Connection Manager : DATA-WAREHOUSE.SIKMA_DW
Use a Table or View : [dbo].[DIM_MAHASISWA]

Karena tabel **DIM_MAHASISWA** belum terbentuk pada *database*, maka buat tabel **DIM_MAHASISWA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_MAHASISWA] ([NPM] VARCHAR(20), [ID_FAKULTAS] INT, [ID_PRODI] VARCHAR(2), [ID_AGAMA] INT, [ID_KAB_KODYA] INT, [NAMA_MHS] VARCHAR(150), [THN_MASUK] INT, [TMP_LAHIR] VARCHAR(255), [TGL_LAHIR] DATETIME, [JNS_KEL] VARCHAR (1), [ID_SMA] INT, CONSTRAINT PK_MAHA PRIMARY KEY(NPM), CONSTRAINT FK_MAHA_PRODI FOREIGN KEY(ID_PRODI) REFERENCES DIM_PRODI(ID_PRODI), CONSTRAINT FK_MAHA_FAKUL FOREIGN KEY(ID_FAKULTAS) REFERENCES DIM_FAKULTAS(ID_FAKULTAS), CONSTRAINT FK_MAHA_AGAMA FOREIGN KEY(ID_AGAMA) REFERENCES DIM_AGAMA(ID_AGAMA), CONSTRAINT FK_MAHA_KAB FOREIGN KEY(ID_KAB_KODYA) REFERENCES DIM_KAB_KODYA(ID_KAB_KODYA),
```

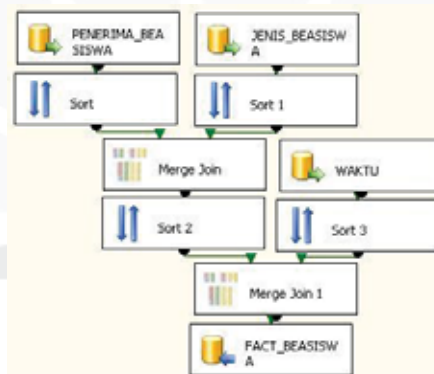

CONSTRAINT FK_MAHA_SMA FOREIGN KEY(ID_SMA) REFERENCES DIM_SMA(ID_SMA))

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.66. Mapping Data Flow Task DIM_MAHASISWA – Full Refresh

10. FACT_BEASISWA



Gambar 5.67. Data Flow Task FACT_BEASISWA – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

- a) PENERIMA_BEASISWA

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
 Data Access Mode : Table or view
 Name of table or view : [dbo].[PENERIMA_BEASISWA]

- b) JENIS_BEASISWA

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
 Data Access Mode : Table or view
 Name of table or view : [dbo].[REF_JENIS_BEASISWA]

- c) WAKTU

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING

Data Access Mode : Table or view

Name of table or view : [dbo].[REF_WAKTU]

- b. Pada *data flow task* **PENERIMA_BEASISWA** dan **JENIS_BEASISWA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **JENIS_BEASISWA** yang telah diurutkan untuk mengambil data **ID_JENIS_BEASISWA**.
- c. Pada *data flow task* **WAKTU** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **TAHUN** dan **SEMESTER** yang telah diurutkan untuk mengambil data **ID_WAKTU**.
- d. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA_DW

Use a Table or View : [dbo].[FACT_BEASISWA]

Karena tabel **FACT_BEASISWA** belum terbentuk pada *database*, maka buat tabel **FACT_BEASISWA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

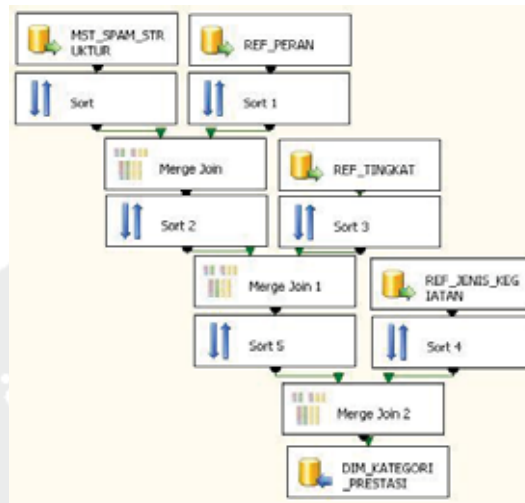
```
CREATE TABLE [FACT_BEASISWA] ([NPM] VARCHAR(20),
[ID_JENIS_BEASISWA] INT, [ID_WAKTU] INT, [NOMINAL] INT, CONSTRAINT
FK_BEA_MAHA FOREIGN KEY(NPM) REFERENCES DIM_MAHASISWA(NPM),
CONSTRAINT FK_BEA_JENIS FOREIGN KEY(ID_JENIS_BEASISWA)
REFERENCES DIM_JENIS_BEASISWA(ID_JENIS_BEASISWA), CONSTRAINT
FK_BEA_WAKTU FOREIGN KEY(ID_WAKTU) REFERENCES
DIM_WAKTU(ID_WAKTU))
```

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.68. Mapping Data Flow Task FACT_BEASISWA – Full Refresh

11. DIM_KATEGORI_PPRESTASI



Gambar 5.69. Data Flow Task DIM_KATEGORI_PPRESTASI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) MST_SPAM_STRUKTUR

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
 Data Access Mode : Table or view
 Name of table or view : [dbo].[MST_SPAM_STRUKTUR]

b) REF_PERAN

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
 Data Access Mode : Table or view
 Name of table or view : [dbo].[REF_SPAM_PERAN]

c) REF_TINGKAT

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
 Data Access Mode : Table or view
 Name of table or view : [dbo].[REF_SPAM_TINGKAT]

d) REF_JENIS_KEGIATAN

Connection Manager : DATA-WAREHOUSE.SIKMA_STAGING
 Data Access Mode : Table or view
 Name of table or view : [dbo].[REF_SPAM_JENIS_KEGIATAN]

- b. Pada *data flow task* **MST_SPAM_STRUKTUR** dan **REF_PERAN** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE_PERAN** yang telah diurutkan untuk mengambil data **KETERANGAN_PERAN**.

- c. Pada *data flow task* **REF_TINGKAT** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE_TINGKAT** yang telah diurutkan untuk mengambil data **KETERANGAN_TINGKAT**.
- d. Pada *data flow task* **REF_JENIS_KEGIATAN** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE_KEGIATAN**, **KODE_JENIS** dan **KODE_KATEGORI** yang telah diurutkan untuk mengambil data **KETERANGAN_KATEGORI**, **KETERANGAN_JENIS**, dan **KETERANGAN_KEGIATAN**.
- e. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan lakukan konfigurasi berikut:
- Connection Manager** : DATA-WAREHOUSE.SIKMA_DW
Use a Table or View : [dbo].[DIM_KATEGORI_PRESTASI]
- Karena tabel **DIM_KATEGORI_PRESTASI** belum terbentuk pada *database*, maka buat tabel **DIM_KATEGORI_PRESTASI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:
- ```
CREATE TABLE [DIM_KATEGORI_PRESTASI] ([ID_KATEGORI_PRESTASI] INT,
[KATEGORI] VARCHAR (80), [JENIS] VARCHAR(100), [KEGIATAN]
VARCHAR(40), [TINGKAT] VARCHAR(20), [PERAN] VARCHAR(30),
CONSTRAINT PK_KATEGORI PRIMARY KEY(ID_KATEGORI_PRESTASI))
```
- f. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.70. Mapping *Data Flow Task* **DIM\_KATEGORI\_PRESTASI** – *Full Refresh*

## 12. DIM\_WAKTU\_PRESTASI



Gambar 5.71. Data Flow Task DIM\_WAKTU\_PRESTASI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_WAKTU\_PRESTASI]

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

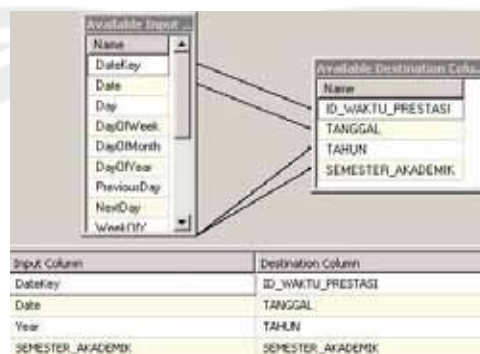
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[DIM\_WAKTU\_PRESTASI]

Karena tabel **DIM\_WAKTU\_PRESTASI** belum terbentuk pada *database*, maka buat tabel **DIM\_WAKTU\_PRESTASI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_WAKTU_PRESTASI] ([ID_WAKTU_PRESTASI] INT,
[TANGGAL] DATETIME, [TAHUN] INT, [SEMESTER_AKADEMIK] INT, CON
STRAINT PK_WAKTU_PRESTASI PRIMARY KEY (ID_WAKTU_PRESTASI))
```

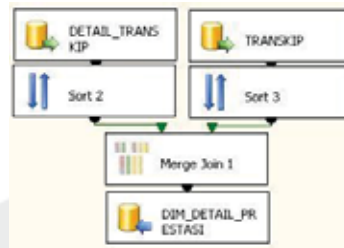
- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.72. Mapping Data Flow Task DIM\_WAKTU\_PRESTASI – Full Refresh



### 13. DIM\_DETAIL\_PRESTASI



Gambar 5.73. Data Flow Task DIM\_DETAIL\_PRESTASI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **DETAIL\_TRANSKIP**

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[TBL\_SPAM\_DETAIL\_TRANSKIP]

b) **TRANSKIP**

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[TBL\_SPAM\_TRANSKIP]

- b. Pada *data flow task* **DETAIL\_TRANSKIP** dan **TRANSKIP** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID\_TRANSKIP\_SPAM** yang telah diurutkan untuk mengambil data **NAMA\_KEGIATAN**.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[DIM\_DETAIL\_PRESTASI]

Karena tabel **DIM\_DETAIL\_PRESTASI** belum terbentuk pada *database*, maka buat tabel **DIM\_DETAIL\_PRESTASI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

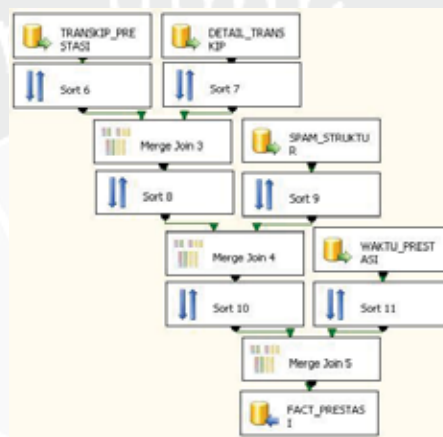
```
CREATE TABLE [DIM_DETAIL_PRESTASI] ([ID_DETAIL_PRESTASI] INT,
[NAMA_KEGIATAN] VARCHAR(1000), [DESKRIPSI] VARCHAR(1000),
[PENYELENGGARA] VARCHAR(500), [TEMPAT] VARCHAR(500), CONSTRAINT
PK_DETAIL PRIMARY KEY(ID_DETAIL_PRESTASI))
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.74. Mapping Data Flow Task DIM\_DETAIL\_PRESTASI – Full Refresh

## 14. FACT\_PRESTASI



Gambar 5.75. Data Flow Task FACT\_PRESTASI – Full Refresh

- a. Data yang akan di-load bersumber dari server, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) TRANSKIP\_PRESTASI

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[TBL\_SPAM\_TRANSKIP]

b) DETAIL\_TRANSKIP

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[TBL\_SPAM\_DETAIL\_TRANSKIP]

c) SPAM\_STRUKTUR

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[MST\_SPAM\_STRUKTUR]

d) WAKTU\_PRESTASI

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[REF\_WAKTU\_PRESTASI]

- b. Pada data flow task **TRANSKIP\_PRESTASI** dan **DETAIL\_TRANSKIP** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source*

harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID\_TRANSKIP\_SPAM** yang telah diurutkan untuk mengambil data **TOTAL\_POIN**.

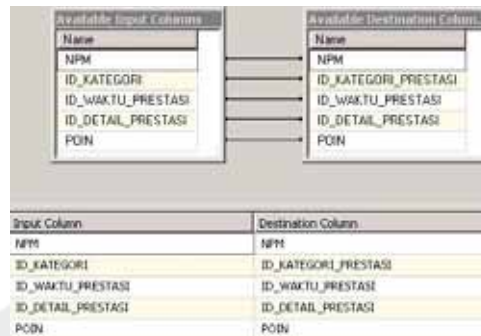
- c. Pada *data flow task* **SPAM\_STRUKTUR** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE\_KATEGORI, KODE\_KEGIATAN, KODE\_JENIS, KODE\_TINGKAT, KODE\_PERAN**, dan **POIN** yang telah diurutkan untuk mengambil data **NO\_URUT**.
- d. Pada *data flow task* **WAKTU\_PRESTASI** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **TGL\_MULAI, ID\_TAHUN\_AKADEMIK**, dan **NO\_SEMESTER** yang telah diurutkan untuk mengambil data **ID\_WAKTU**.
- e. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[FACT\_PRESTASI]

Karena tabel **FACT\_PRESTASI** belum terbentuk pada *database*, maka buat tabel **FACT\_PRESTASI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [FACT_PRESTASI] ([NPM] VARCHAR(20), [ID_KATEGORI_PRESTASI] INT, [ID_WAKTU_PRESTASI] INT, [ID_DETAIL_PRESTASI] INT, [POIN] INT, CONSTRAINT FK_PRES_MAHA FOREIGN KEY (NPM) REFERENCES DIM_MAHASISWA(NPM), CONSTRAINT FK_PRES_KATE FOREIGN KEY (ID_KATEGORI_PRESTASI) REFERENCES DIM_KATEGORI_PRESTASI(ID_KATEGORI_PRESTASI), CONSTRAINT FK_PRES_WAKTU FOREIGN KEY (ID_WAKTU_PRESTASI) REFERENCES DIM_WAKTU_PRESTASI(ID_WAKTU_PRESTASI), CONSTRAINT FK_PRES_DETAIL FOREIGN KEY (ID_DETAIL_PRESTASI) REFERENCES DIM_DETAIL_PRESTASI (ID_DETAIL_PRESTASI))
```

- f. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.76. Mapping Data Flow Task FACT\_PRESTASI – Full Refresh

6). *Sequence Container* **LOAD1**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD1**.

7). *Execute SQL Task* **UPDATE\_FAKULTAS\_MAHasiswa**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD1** dan beri nama **UPDATE\_FAKULTAS\_MAHasiswa**. Ikon *Execute SQL Task* ini berfungsi menjalankan *query* untuk mengubah data fakultas yang ada pada *database* **SIKMA\_DW**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW**). Lakukan konfigurasi dengan menuliskan sintaks berikut:

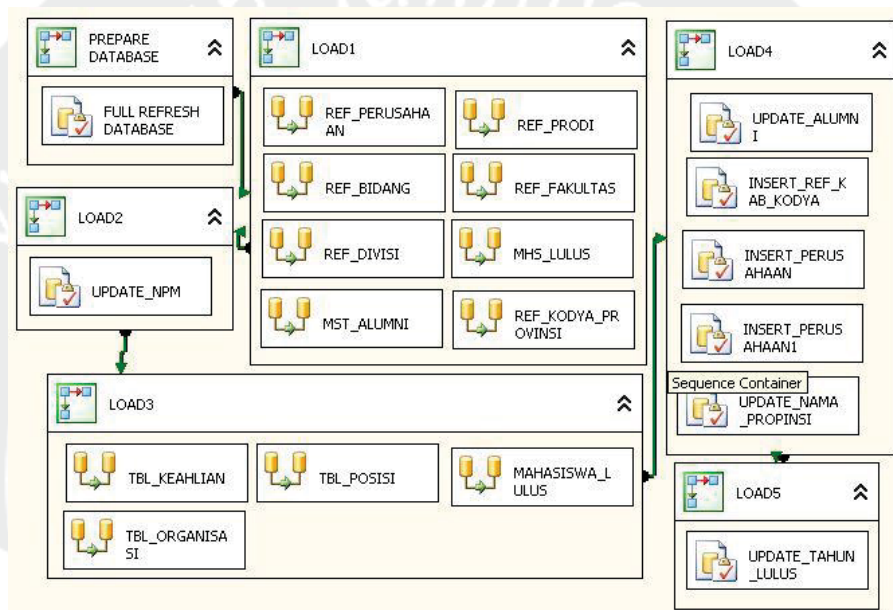
```
UPDATE DIM_MAHasiswa SET ID_FAKULTAS=5 WHERE ID_PRODI IN(01,02, 13);
UPDATE DIM_MAHasiswa SET ID_FAKULTAS=1 WHERE ID_PRODI IN(
03,04,11,12,15,20); UPDATE DIM_MAHasiswa SET ID_FAKULTAS=2 WHERE
ID_PRODI IN(05,21); UPDATE DIM_MAHasiswa SET ID_ FAKULTAS=3 WHERE
ID_PRODI =08; UPDATE DIM_MAHasiswa SET ID_ FAKULTAS=4 WHERE
ID_PRODI IN(06,07,14,24); UPDATE DIM_MAHasiswa SET ID_FAKULTAS=6
WHERE ID_PRODI IN(09,10); UPDATE DIM_MAHasiswa SET ID_AGAMA=6
WHERE ID_AGAMA IS NULL; UPDATE DIM_MAHasiswa SET ID_PRODI = 25
WHERE ID_PRODI IS NULL; UPDATE DIM_MAHasiswa SET ID_FAKULTAS=7
WHERE ID_FAKULTAS IS NULL;
```

### 5.1.3. *Control Flow Full Refresh* ALUMNI

Proses *full refresh* ALUMNI terbagi menjadi dua proses. Proses pertama adalah mengisi data dari *database* sumber menuju *staging area*. Proses selanjutnya adalah mengisi data dari *staging area* menuju *data warehouse*.

### 5.1.3.1. Loading Data dari Sumber Data ke ALUMNI\_STAGING

Pada proses *loading* data dari *database* sumber ke dalam *database* **SIKMA\_STAGING**, langkah pertama adalah menentukan *control flow items* yang sesuai dengan kebutuhan pada *tab control flow*. Untuk menggunakan *control flow items* yang telah disediakan, tarik dan letakkan ikon *control flow items* yang diperlukan ke area tengah. Kemudian proses *loading* data dari sumber data ke **SIKMA\_STAGING** pada *package* **ALUMNI\_STAGING.dtsx** akan terbentuk desain proses sebagai berikut :



**Gambar 5.77. Desain Control Flow pada ALUMNI\_STAGING – Full Refresh**

Untuk membuat desain *Control Flow* di atas, langkah-langkah yang dilakukan adalah sebagai berikut:

1). *Sequence Container* **Prepare Database**

Tarik dan letakkan ikon *sequence container* dan beri nama **PREPARE DATABASE**.

2). *Execute SQL Task* **FULL\_REFRESH\_DATABASE**

Tarik dan letakkan ikon *execute SQL task* ke dalam item **PREPARE DATABASE** *sequence container* yang telah dibuat dan beri nama **FULL\_REFRESH\_DATABASE**. Ikon *execute SQL task* berfungsi menjalankan *query* untuk menghapus semua isi tabel yang ada pada *database* **SIKMA\_STAGING**. Hal ini dilakukan agar isi tabel pada *database* **SIKMA\_STAGING** selalu baru. Pilih koneksi tersebut (**DATA-**



**WAREHOUSE. SIKMA\_STAGING**). Setelah proses tersebut, lakukan konfigurasi dengan menuliskan *sintaks* berikut:

```
DELETE FROM DIM_AGAMA; DELETE FROM FACT_BEASISWA; DELETE FROM DIM_SMA; DELETE FROM FACT_PRESTASI; DELETE FROM DIM_KATEGORI_PRESTASI; DELETE FROM DIM_JENIS_BEASISWA; DELETE FROM DIM_MAHASISWA; DELETE FROM DIM_PRODI; DELETE FROM DIM_FAKULTAS; DELETE FROM DIM_WAKTU; DELETE FROM DIM_KAB_KODYA; DELETE FROM DIM_KAB_SMA; DELETE FROM DIM_WAKTU_PRESTASI; DELETE FROM DIM_DETAIL_PRESTASI
```

3). *Sequence Container* **LOAD1**

Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD1**.

4). *Data Flow Task* **LOAD1**

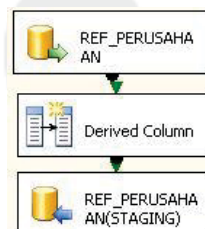
*Control flow item* yang digunakan untuk ekstrak, *transform*, dan *load* data adalah *data flow task*. Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* **LOAD1**. Karena ada 8 data sumber yang akan di-*load* maka dibuat 8 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **REF\_PERUSAHAAN**, **REF\_BIDANG**, **REF\_DIVISI**, **MST\_ALUMNI**, **REF\_PRODI**, **REF\_FAKULTAS**, **MHS\_LULUS**, dan **REF\_KODYA\_PROVINSI**.

Hubungkan **PREPARE DATABASE** *sequence container* dengan *sequence container* **LOAD1**.

5). *Data Flow Task* pada *Sequence Container* **LOAD1**

Langkah-langkah untuk membuat *data flow task* pada **LOAD1** sebagai berikut:

**1. REF\_PERUSAHAAN**



**Gambar 5.78.** *Data Flow Task* **REF\_PERUSAHAAN** – *Full Refresh*

- a. Data yang akan di-*load* bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : **DATA-WAREHOUSE.siatmax**

**Data Access Mode** : **Table or view**

Name of table or view : dbo].[REF\_PERUSAHAAN]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

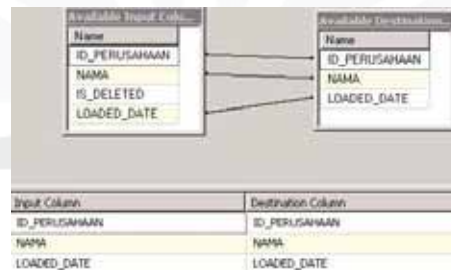
Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING

Use a Table or View : [dbo].[REF\_PERUSAHAAN]

Karena tabel **REF\_PERUSAHAAN** belum terbentuk pada *database*, maka buat tabel **REF\_PERUSAHAAN** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [REF_PERUSAHAAN]([ID_PERUSAHAAN] INT, [NAMA]
VARCHAR(50), [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.79. Mapping Data Flow Task REF\_PERUSAHAAN – Full Refresh

## 2. REF\_BIDANG



Gambar 5.80. Data Flow Task REF\_BIDANG – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax

Data Access Mode : Table or view

Name of table or view : dbo].[REF\_BIDANG]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

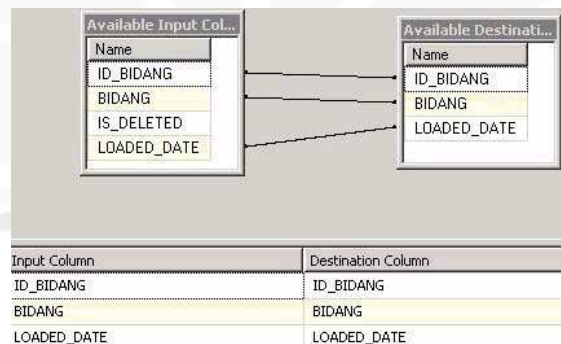
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_BIDANG]

Karena tabel **REF\_BIDANG** belum terbentuk pada *database*, maka buat tabel **REF\_BIDANG** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

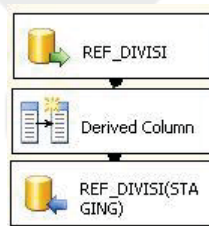
```
CREATE TABLE [REF_BIDANG]([ID_BIDANG] INT, [BIDANG]
VARCHAR(50), [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



**Gambar 5.81. Mapping Data Flow Task REF\_BIDANG – Full Refresh**

### 3. REF\_DIVISI



**Gambar 5.82. Data Flow Task REF\_DIVISI – Full Refresh**

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : dbo.[REF\_DIVISI]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

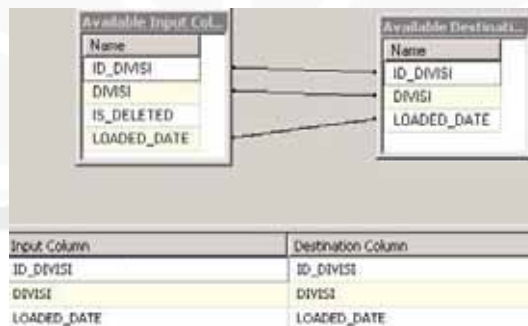
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_DIVISI]

Karena tabel **REF\_DIVISI** belum terbentuk pada *database*, maka buat tabel **REF\_DIVISI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

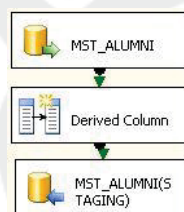
```
CREATE TABLE [REF_DIVISI]([ID_DIVISI] INT, [DIVISI] VARCHAR(50), [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.83. Mapping Data Flow Task REF\_DIVISI – Full Refresh

#### 4. MST\_ALUMNI



Gambar 5.84. Data Flow Task MST\_ALUMNI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[MST\_ALUMNI]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

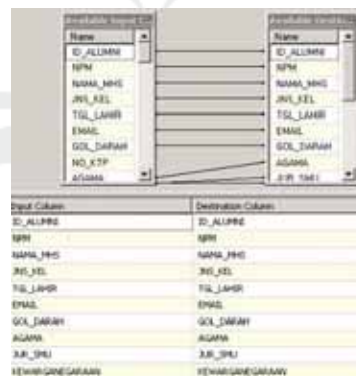
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[MST\_ALUMNI]

Karena tabel **MST\_ALUMNI** belum terbentuk pada *database*, maka buat tabel **MST\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [MST_ALUMNI]([ID_ALUMNI] INT, [NPM] VARCHAR (10),
[NAMA_MHS] VARCHAR(100), [JNS_KEL] VARCHAR(1), [TGL_LAHIR]
DATETIME, [EMAIL] VARCHAR(100), [GOL_DARAH] VARCHAR(2),
[AGAMA] VARCHAR(20), [JUR_SMU] NVARCHAR(10),
[KEWARGANEGARAAN] NVARCHAR(5), [ID_FAKULTAS] TINYINT,
[ID_PRODI] NVARCHAR(2), [TMP_LAHIR] NVARCHAR(50), [TAHUN_
MASUK] INT, [TAHUN_LULUS] INT, [GAJI_PERTAMA] FLOAT,
[LOADED_DATE] DATETIME)
```

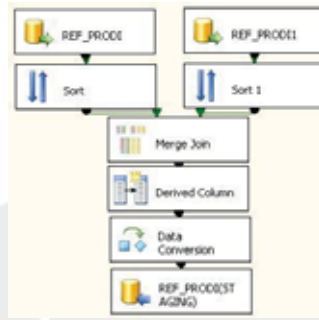
- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



**Gambar 5.85. Mapping Data Flow Task MST\_ALUMNI – Full Refresh**



## 5. REF\_PRODI



Gambar 5.86. *Data Flow Task REF\_PRODI – Full Refresh*

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **REF\_PRODI**

**Connection Manager** : DATA-WAREHOUSE.biadw

**Data Access Mode** : Table or view

**Name of table or view** : dbo].[prodi]

b) **REF\_PRODI1**

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : dbo].[REF\_PRODI]

b. Pada *Data Flow Task* **REF\_PRODI** dan **REF\_PRODI1** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Kemudian data digabungkan berdasarkan **id\_prodi** yang telah diurutkan untuk mengambil data **nama\_prodi**.

c. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.

d. Fungsi *data conversion* digunakan untuk mengkonversi **ID\_PRODI** menjadi data *varchar*.

e. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

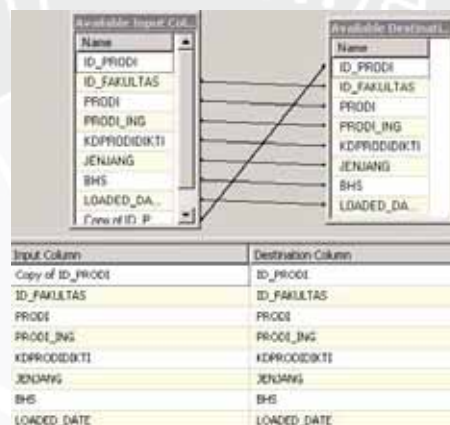
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_PRODI]

Karena tabel **REF\_PRODI** belum terbentuk pada *database*, maka buat tabel **REF\_PRODI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

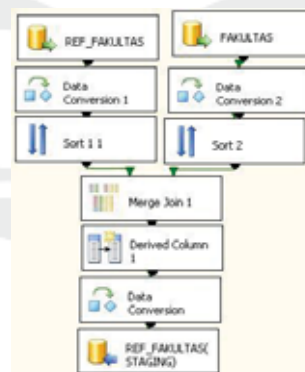
```
CREATE TABLE [REF_PRODI]([ID_PRODI] VARCHAR (2), [ID_FAKULTAS]
INT, [PRODI] VARCHAR(50), [PRODI_ING] VARCHAR (50), [KDPRODIKTI]
VARCHAR (5), [JENJANG] VARCHAR(2), [BHS] VARCHAR(1),
[LOADED_DATE] DATETIME)
```

- f. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.87. Mapping Data Flow Task REF\_PRODI – Full Refresh

## 6. REF\_FAKULTAS



Gambar 5.88. Data Flow Task REF\_FAKULTAS – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **REF\_FAKULTAS**

Connection Manager : DATA-WAREHOUSE.siatmax

Data Access Mode : Table or view

Name of table or view : dbo].[REF\_FAKULTAS]

b) **FAKULTAS**

**Connection Manager** : DATA-WAREHOUSE.biadw

**Data Access Mode** : Table or view

**Name of table or view** : dbo].[fakultas]

- b. Fungsi *data conversion* digunakan untuk mengkonversi **ID\_FAKULTAS** menjadi data *int* dan **KD\_FAKULTAS** menjadi data *int*.
- c. Pada *Data Flow Task* **REF\_FAKULTAS** dan **FAKULTAS** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Kemudian data digabungkan berdasarkan **ID\_FAKULTAS** yang telah diurutkan untuk mengambil data **NAMA\_FAKULTAS**.
- d. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- e. Fungsi *data conversion* digunakan untuk mengkonversi **FAKULTAS** menjadi data *varchar*.
- f. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *data conversion*.

Lakukan konfigurasi berikut:

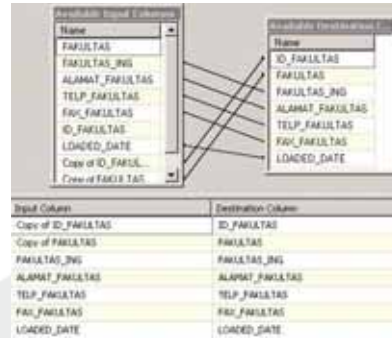
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_FAKULTAS]

Karena tabel **REF\_FAKULTAS** belum terbentuk pada *database*, maka buat tabel **REF\_FAKULTAS** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

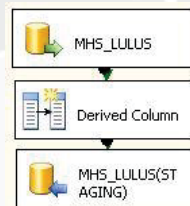
```
CREATE TABLE [REF_FAKULTAS]([ID_FAKULTAS] INT, [FAKULTAS]
VARCHAR(30), [FAKULTAS_ING] VARCHAR(30), [ALAMAT_FAKULTAS]
VARCHAR (80), [TELP_FAKULTAS] VARCHAR (12), [FAX_FAKULTAS]
VARCHAR(12), [LOADED_DATE] DATETIME)
```

- g. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.89. Mapping Data Flow Task REF\_FAKULTAS – Full Refresh

## 7. MHS\_LULUS



Gambar 5.90. Data Flow Task MHS\_LULUS – Full Refresh

- Data yang akan di-load bersumber dari server, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.biadw  
**Data Access Mode** : Table or view  
**Name of table or view** : dbo].[mhs\_lulus]

- Tambahkan kolom waktu pada saat data dimasukkan ke dalam database sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

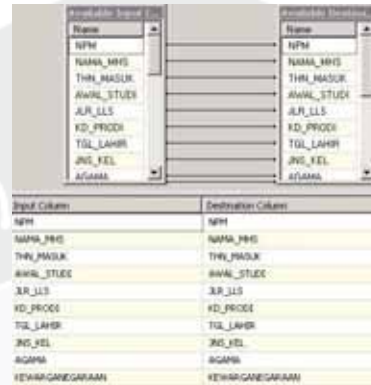
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[TBL\_MHS\_LULUS]

Karena tabel **TBL\_MHS\_LULUS** belum terbentuk pada database, maka buat tabel **TBL\_MHS\_LULUS** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [TBL_MHS_LULUS]([NPM] VARCHAR(10), [NAMA_ MHS]
VARCHAR(150), [THN_MASUK] NUMERIC(4,0), [AWAL_ STUDI]
DATETIME, [JLR_LLS] VARCHAR (25), [KD_PRODI] VARCHAR(5),
[TGL_LAHIR] DATETIME, [JNS_KEL] VARCHAR(2), [AGAMA]
VARCHAR(15), [KEWARGANEGARAAN] VARCHAR(5), [ID_ KAB_ORTU]
```

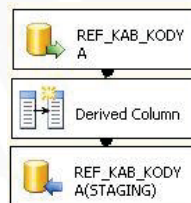
INT, [TGL\_WISUDA] DATETIME, [TGL\_YUD] DATETIME, [IPK\_LULUS] NUMERIC(4,2), [LAMA\_STUDI] NUMERIC(4,2), [LAMA\_TA] REAL, [LOADED\_DATE] DATETIME)

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.91. Mapping Data Flow Task MHS\_LULUS – Full Refresh

## 8. REF\_KODYA\_PROVINSI



Gambar 5.92. Data Flow Task REF\_KODYA\_PROVINSI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.biadw

**Data Access Mode** : Table or view

**Name of table or view** : dbo].[prop\_kab\_kodya]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

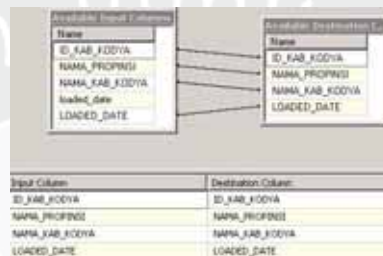
**Use a Table or View** : [dbo].[REF\_KAB\_KODYA]



Karena tabel **REF\_KAB\_KODYA** belum terbentuk pada *database*, maka buat tabel **REF\_KAB\_KODYA** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [REF_KAB_KODYA]([ID_KAB_KODYA] INT, [NAMA_PROVINSI] VARCHAR(150), [NAMA_KAB_KODYA] VARCHAR (150), [LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



**Gambar 5.93. Mapping Data Flow Task REF\_KODYA\_PROVINSI – Full Refresh**

6). *Sequence Container* **LOA2**

Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD2**.

7). *Execute SQL Task* **LOAD2**

Tarik dan letakkan ikon *execute SQL task* ke dalam item **LOAD2** dan beri nama **UPDATE\_NPM**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_STAGING**). Ikon *Execute SQL Task* **UPDATE\_NPM** berfungsi menjalankan *query* untuk mengubah data npm mahasiswa pada *database* **SIKMA\_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE MST_ALUMNI SET NPM=SUBSTRING(NPM,1,9) WHERE NPM IN(SELECT NPM FROM MST_ALUMNI);
```

8). *Sequence Container* **LOA3**

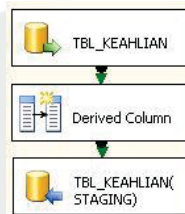
Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD3**.

7). *Data Flow Task* **LOAD3**

Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* **LOAD3**. Karena ada 4 data sumber yang akan di-load maka dibuat 4 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **TBL\_KEAHLIAN**, **TBL\_POSISI**, **MAHASISWA\_LULUS**, dan **TBL\_ORGANISASI**. Hubungkan **LOAD3** *sequence container* dengan *sequence container* **LOAD2**.

8). *Data Flow Task* pada *Sequence Container* **LOAD3**

Langkah-langkah untuk membuat *data flow task* pada **LOAD3** sebagai berikut:

1. **TBL\_KEAHLIAN**

Gambar 5.94. *Data Flow Task* **TBL\_KEAHLIAN** – *Full Refresh*

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : **DATA-WAREHOUSE.siatmax**  
**Data Access Mode** : **Table or view**  
**Name of table or view** : **dbo].[TBL\_KEAHLIAN]**

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

**Connection Manager** : **DATA-WAREHOUSE.SIKMA\_STAGING**  
**Use a Table or View** : **[dbo].[TBL\_KEAHLIAN]**

Karena tabel **TBL\_KEAHLIAN** belum terbentuk pada *database*, maka buat tabel **TBL\_KEAHLIAN** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```

CREATE TABLE [TBL_KEAHLIAN]([ID_KEAHLIAN] INT, [ID_ ALUMNI] INT,
[NAMA] VARCHAR(250), [LEVEL_KEAHLIAN] VARCHAR(100), [TAHUN]
INT, [INSERT_DATE] DATETIME, [IP_ ADDRESS] VARCHAR(30),
[USERNAME] VARCHAR(50), [LOADED_ DATE] DATETIME)

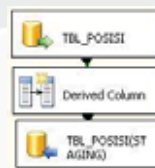
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.95. Mapping Data Flow Task TBL\_KEAHLIAN – Full Refresh

## 2. TBL\_POSISI



Gambar 5.96. Data Flow Task TBL\_POSISI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : dbo].[TBL\_POSISI]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[TBL\_POSISI]

Karena tabel **TBL\_POSISI** belum terbentuk pada *database*, maka buat tabel **TBL\_POSISI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

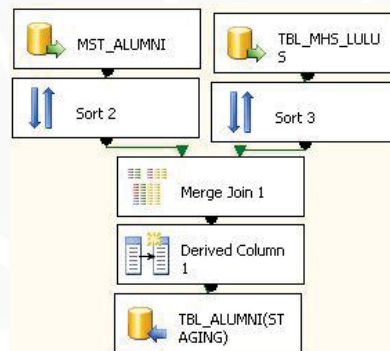
```
CREATE TABLE [TBL_POSISI]([ID_POSISI] INT, [ID_ALUMNI] INT,
[PERUSAHAAN] VARCHAR(100), [BIDANG] VARCHAR(100), [JABATAN]
VARCHAR(100), [DIVISI] VARCHAR(50), [THN_MASUK] INT,
[THN_KELUAR] INT, [DESKRIPSI] VARCHAR(250), [LOADED_ DATE]
DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.97. Mapping Data Flow Task TBL\_POSISI – Full Refresh

### 3. MAHASISWA\_LULUS



Gambar 5.98. Data Flow Task MAHASISWA\_LULUS – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:
  - a) **MST\_ALUMNI**  
 Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : dbo].[MST\_ALUMNI]
  - b) **TBL\_MHS\_LULUS**  
 Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : dbo].[TBL\_MHS\_LULUS]
- b. Pada *data flow task* **MST\_ALUMNI** dan **TBL\_MHS\_LULUS** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil

data **AWAL\_STUDI**, **THN\_MASUK**, **JLR\_LULUS**, dan **KD\_PRODI**.

- c. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- d. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[TBL\_ALUMNI]

Karena tabel **TBL\_ALUMNI** belum terbentuk pada *database*, maka buat tabel **TBL\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

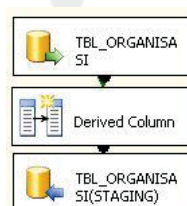
```
CREATE TABLE [TBL_ALUMNI]([ID_FAKULTAS] INT, [ID_PRODI]
VARCHAR(2), [TMP_LAHIR] VARCHAR(50), [THN_MASUK] INT,
[TAHUN_LULUS] INT, [GAJI_PERTAMA] FLOAT, [AWAL_STUDI]
DATETIME, [ID_KAB_KODYA] INT, [TGL_WISUDA] DATETIME, [TGL_YUD]
DATETIME, [IPK_LULUS] NUMERIC(4,2), [LAMA_STUDI] NUMERIC(4,2),
[LAMA_TA] REAL, [LAMA_MENUNGGU_KERJA] VARCHAR(30),
[LOADED_DATE] DATETIME)
```

- e. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.99. Mapping *Data Flow Task* MAHASISWA\_LULUS – *Full Refresh*

#### 4. TBL\_ORGANISASI



Gambar 5.100. *Data Flow Task* TBL\_ORGANISASI – *Full Refresh*



- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : dbo].[TBL\_ORGANISASI]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[TBL\_ORGANISASI]

Karena tabel **TBL\_ORGANISASI** belum terbentuk pada *database*, maka buat tabel **TBL\_ORGANISASI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [TBL_ORGANISASI]([ID_ORGANISASI] INT, [ID_ ALUMNI]
INT, [NAMA_ORGANISASI] VARCHAR(100), [THN_MULAI] INT,
[THN_SELESAI] INT, [JABATAN] VARCHAR(50), [INSERT_ DATE]
DATETIME, [IP_ADDRESS] VARCHAR(30), [USERNAME] VARCHAR(50),
[LOADED_DATE] DATETIME)
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



**Gambar 5.101. Mapping Data Flow Task TBL\_ORGANISASI – Full Refresh**

#### 9). Sequence Container **LOAD4**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD4**.

10). *Execute SQL Task* **LOAD4**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD4** sebanyak 5 kali dan beri nama **UPDATE\_ALUMNI, INSERT\_REF\_KAB\_KODYA, INSERT\_PERUSAHAAN, INSERT\_PERUSAHAAN1, dan UPDATE\_NAMA\_PROPINSI**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_STAGING**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

- a. Ikon *Execute SQL Task* **UPDATE\_ALUMNI** berfungsi untuk mengubah data alumni yang ada pada *database* **SIKMA\_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE TBL_ALUMNI SET ID_FAKULTAS=5 WHERE ID_PRODI IN(01,02,13);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=1 WHERE ID_PRODI IN(03,04,11,12,15,20);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=2 WHERE ID_PRODI IN(05,21);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=3 WHERE ID_PRODI =08;
UPDATE TBL_ALUMNI SET ID_FAKULTAS=4 WHERE ID_PRODI IN(06,07,14,24);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=6 WHERE ID_PRODI IN(09,10);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=7 WHERE ID_PRODI IN(50,51,52,53,54);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=8 WHERE ID_PRODI=60;
UPDATE TBL_ALUMNI SET JNS_KEL= ' ' WHERE JNS_KEL IS NULL;
UPDATE TBL_ALUMNI SET AGAMA='N/A' WHERE AGAMA IS NULL;
UPDATE TBL_ALUMNI SET TMP_LAHIR='N/A' WHERE TMP_LAHIR IS NULL;
```

- b. Ikon *Execute SQL Task* **INSERT\_REF\_KAB\_KODYA** berfungsi untuk menambahkan data kabupaten kodya yang ada pada *database* **SIKMA\_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT INTO REF_KAB_KODYA VALUES(533,NULL,NULL,NULL);
INSERT INTO REF_KAB_KODYA VALUES(475,NULL,NULL,NULL);
INSERT INTO REF_KAB_KODYA VALUES(130,NULL,NULL,NULL);
```

- c. Ikon *Execute SQL Task* **INSERT\_PERUSAHAAN** berfungsi untuk menambah data perusahaan yang ada pada *database* **SIKMA\_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT INTO TBL_POSISI(ID_POSISI,ID_ALUMNI,PERUSAHAAN)VALUES(88,10833,'Adaro');
INSERT INTO TBL_POSISI(ID_POSISI,ID_ALUMNI,PERUSAHAAN)VALUES(89,11374,'Adicipta Inovasi Tekno logi');
```

- d. Ikon *Execute SQL Task* **INSERT\_PERUSAHAAN1** berfungsi untuk menambah data perusahaan yang ada pada *database* **SIKMA\_STAGING**.

Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
TBL_POSISI(ID_POSISI,ID_ALUMNI,PERUSAHAAN)VALUES(421,32107,'PT.Halilin
tar Lintas Semesta'); INSERT INTO TBL_POSISI
(ID_POSISI,ID_ALUMNI,PERUSAHAAN)VALUES(422,33354,'PT. Harita Prima Abadi
Mineral'); INSERT INTO TBL_POSISI(ID_POSISI,
ID_ALUMNI,PERUSAHAAN)VALUES(423,32734,'PT. HM Sampoerna, Tbk');
```

- e. Ikon *Execute SQL Task* **UPDATE\_NAMA\_PROPINSI** berfungsi untuk mengubah data propinsi yang ada pada *database* **SIKMA\_STAGING**.

Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE REF_KAB_KODYA SET NAMA_PROPINSI='Nanggroe Aceh Darusalam'
WHERE ID_KAB_KODYA IN(SELECT ID_KAB_KODYA WHERE
NAMA_PROPINSI='NANGROE ACEH DARUSALAM'); UPDATE
REF_KAB_KODYA SET NAMA_PROPINSI='Bangka Belitung' WHERE
ID_KAB_KODYA IN(SELECT ID_KAB_KODYA WHERE
NAMA_PROPINSI='KEPULAUAN BANGKA BELITUNG');
```

#### 11). *Sequence Container* **LOAD5**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD5**.

#### 12). *Execute SQL Task* **LOAD5**

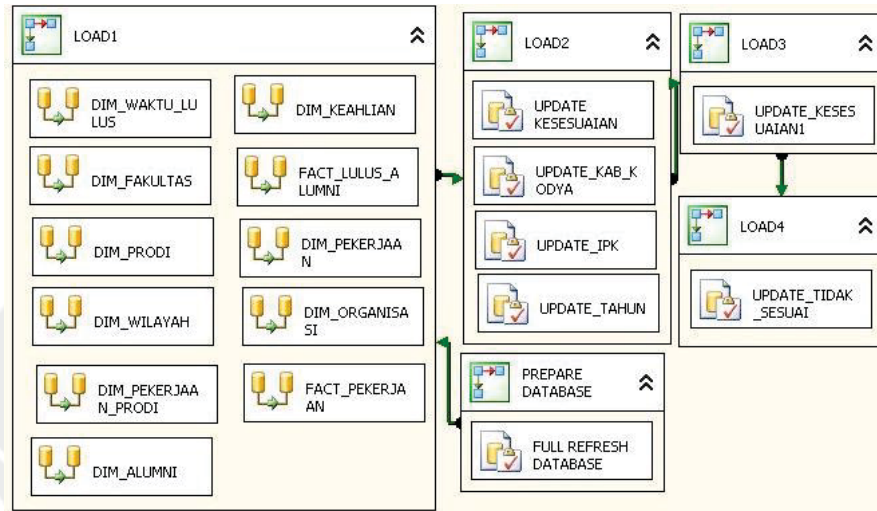
Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD5** dan beri nama **UPDATE\_TAHUN\_LULUS**. Pilih koneksi tersebut (**DATA-WAREHOUSE. SIKMA\_STAGING**). Ikon *Execute SQL Task* **UPDATE\_TAHUN\_LULUS** berfungsi untuk mengubah data tahun lulus yang ada pada *database* **SIKMA\_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE dbo.TBL_ALUMNI SET TAHUN_LULUS = SUBSTRING(CAST(DATE
PART([YEAR],TGL_WISUDA) AS VARCHAR(30)),1,4) WHERE(NPM IN
(SELECT NPM FROM dbo.TBL_ALUMNI AS TBL_ALUMNI_1))
```

### 5.1.3.2. Loading Data dari ALUMNI\_STAGING ke ALUMNI\_DW

Dalam tahap ini, data pada *database* **SIKMA\_STAGING** akan di-load ke dalam *data warehouse* yaitu **SIKMA\_DW\_ALUMNI**. Pada proses *loading* data dari *database* **SIKMA\_STAGING** ke dalam *database* **SIKMA\_DW\_ALUMNI**, langkah pertama adalah menentukan *control flow items*

yang sesuai dengan kebutuhan pada *tab control flow*. Proses *loading* data dari **SIKMA\_STAGING** ke **SIKMA\_DW\_ALUMNI** pada *package* **ALUMNI\_DW.dtsx** akan dibuat desain proses sebagai berikut :



**Gambar 5.102. Desain Control Flow pada ALUMNI\_DW – Full Refresh**

Untuk membuat desain *Control Flow* di atas, langkah-langkah yang dilakukan adalah sebagai berikut:

1). *Sequence Container* **Prepare Database**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **PREPARE DATABASE**.

2). *Execute SQL Task* **FULL\_REFRESH\_DATABASE**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **PREPARE DATABASE** *sequence container* yang telah dibuat dan beri nama **FULL\_REFRESH\_DATABASE**. Ikon *Execute SQL Task* ini berfungsi menjalankan *query* untuk menghapus semua isi tabel yang ada pada *database* **SIKMA\_DW\_ALUMNI**. Hal ini dilakukan agar isi tabel pada *database* **SIKMA\_DW\_ALUMNI** selalu baru. Buat koneksi ke *database* pada *server* **DATA-WAREHOUSE**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI**). Langkah-langkah pembuatan koneksi **DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI** dapat diamati pada lampiran tugas akhir ini. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
DELETE FROM FACT_LULUS_ALUMNI; DELETE FROM FACT_PKERJAAAN;
DELETE FROM DIM_WAKTU_LULUS; DELETE FROM DIM_PRODI_ALUMNI;
DELETE FROM DIM_FAKULTAS_ALUMNI; DELETE FROM DIM_WILAYAH_
```

ALUMNI; DELETE FROM DIM\_KEAHLIAN\_ALUMNI; DELETE FROM DIM\_ALUMNI; DELETE FROM DIM\_Pekerjaan\_ProDI; DELETE FROM DIM\_ORGANISASI\_ALUMNI; DELETE FROM DIM\_Pekerjaan\_AlumNI;

### 3). *Sequence Container* **LOAD1**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD1**.

### 4). *Data Flow Task* **LOAD1**

Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* tersebut. Karena ada 11 data sumber yang akan di-load maka dibuat 11 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **DIM\_WAKTU\_LULUS**, **DIM\_FAKULTAS**, **DIM\_PRODI**, **DIM\_WILAYAH**, **DIM\_Pekerjaan\_ProDI**, **DIM\_ALUMNI**, **DIM\_KEAHLIAN**, **FACT\_LULUS\_ALUMNI**, **DIM\_Pekerjaan**, **DIM\_ORGANISASI**, dan **FACT\_Pekerjaan**.

Hubungkan **PREPARE DATABASE** *sequence container* dengan *sequence container* **LOAD1**.

### 5). *Data Flow Task* pada *Sequence Container* **LOAD1**

Langkah-langkah untuk membuat *data flow task* **LOAD1** sebagai berikut:

#### 1. DIM\_WAKTU\_LULUS



Gambar 5.103. *Data Flow Task* DIM\_WAKTU\_LULUS – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_WAKTU\_WISUDA]

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[DIM\_WAKTU\_LULUS]



Karena tabel **DIM\_WAKTU\_LULUS** belum terbentuk pada *database*, maka buat tabel **DIM\_WAKTU\_LULUS** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_WAKTU_LULUS] ([ID_WAKTU_LULUS] INT,
[TGL_LULUS] DATETIME, [BULAN] INT, [TAHUN] INT, CONSTRAINT
PK_WAKTU_LULUS PRIMARY KEY(ID_WAKTU_LULUS))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.104. Mapping Data Flow Task DIM\_WAKTU\_LULUS – Full Refresh

## 2. DIM\_FAKULTAS



Gambar 5.105. Data Flow Task DIM\_FAKULTAS – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_FAKULTAS]

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

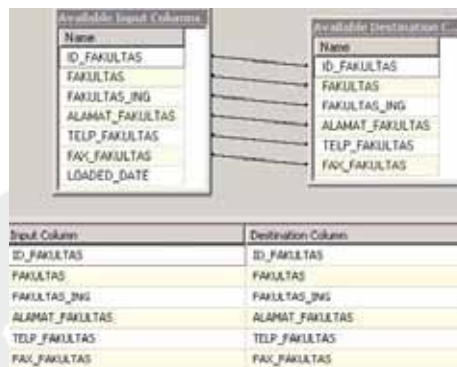
**Use a Table or View** : [dbo].[DIM\_FAKULTAS\_ALUMNI]

Karena tabel **DIM\_FAKULTAS\_ALUMNI** belum terbentuk pada *database*, maka buat tabel **DIM\_FAKULTAS\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_FAKULTAS_ALUMNI] ([ID_FAKULTAS] INT, [FAKULTAS]
VARCHAR(30), [FAKULTAS_ING] VARCHAR(30), [ALAMAT_FAKULTAS]
VARCHAR(80), [TELP_FAKULTAS] VARCHAR(12), [FAX_FAKULTAS]
VARCHAR(12), CONSTRAINT PK_FAKULTAS_ALUMNI PRIMARY
KEY(ID_FAKULTAS))
```

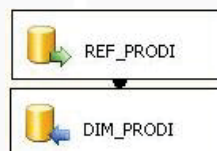


- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.106. Mapping Data Flow Task DIM\_FAKULTAS – Full Refresh

### 3. DIM\_PRODI



Gambar 5.107. Data Flow Task DIM\_PRODI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_PRODI]

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

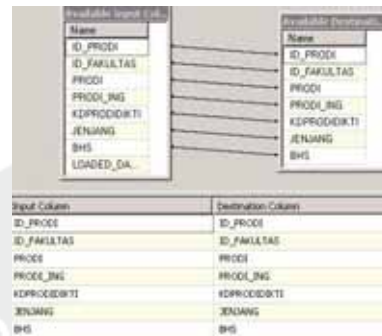
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[DIM\_PRODI\_ALUMNI]

Karena tabel **DIM\_PRODI\_ALUMNI** belum terbentuk pada *database*, maka buat tabel **DIM\_PRODI\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_PRODI_ALUMNI] ([ID_PRODI] VARCHAR(2), [ID_FAKULTAS] INT, [PRODI] VARCHAR(50), [PRODI_ING] VARCHAR(50), [KDPRODIDIKTI] VARCHAR(5), [JENJANG] VARCHAR(2), [BHS] VARCHAR(1), CONSTRAINT PK_PRODI_ALUMNI PRIMARY KEY(ID_PRODI), CONSTRAINT FK_PRODI_ALUMNI FOREIGN KEY(ID_FAKULTAS) REFERENCES DIM_FAKULTAS_ALUMNI(ID_FAKULTAS))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.108. Mapping Data Flow Task DIM\_PRODI – Full Refresh

#### 4. DIM\_WILAYAH



Gambar 5.109. Data Flow Task DIM\_WILAYAH – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_KAB\_KODYA]

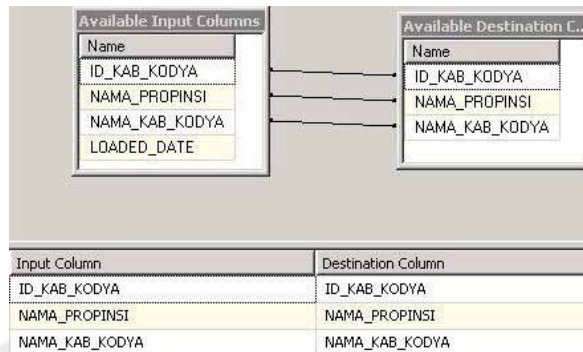
- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_WILAYAH\_ALUMNI]

Karena tabel **DIM\_WILAYAH\_ALUMNI** belum terbentuk pada *database*, maka buat tabel **DIM\_WILAYAH\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_WILAYAH_ALUMNI] ([ID_KAB_KODYA] INT,
[NAMA_PROPINSI] VARCHAR(150), [NAMA_KAB_KODYA] VARCHAR (150),
CONSTRAINT PK_WILAYAH_ALUMNI PRIMARY KEY(ID_KAB_KODYA))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.110. Mapping Data Flow Task DIM\_WILAYAH – Full Refresh

## 5. DIM\_PKERJAAN\_PRODI



Gambar 5.111. Data Flow Task DIM\_PKERJAAN\_PRODI – Full Refresh

- a. Data yang akan di-load bersumber dari server, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo],[REF\_JABATAN]

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

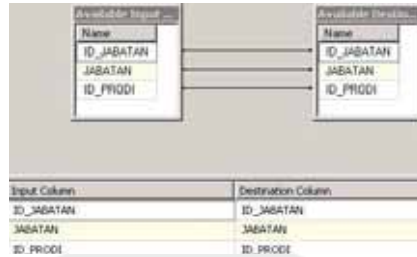
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo],[DIM\_PKERJAAN\_PRODI]

Karena tabel **DIM\_PKERJAAN\_PRODI** belum terbentuk pada *database*, maka buat tabel **DIM\_PKERJAAN\_PRODI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_PKERJAAN_PRODI] ([ID_JABATAN] INT, [JABATAN]
VARCHAR(100), [ID_PRODI] VARCHAR(2), CONSTRAINT
PK_PKERJAAN_PRODI PRIMARY KEY(ID_JABATAN), CONSTRAINT
FK_PRODI_PRODI FOREIGN KEY(ID_PRODI) REFERENCES
DIM_PRODI_ALUMNI(ID_PRODI))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* sebagai berikut:



Gambar 5.112. Mapping *Data Flow Task* DIM\_Pekerjaan\_Alumni – Full Refresh

## 6. DIM\_ALUMNI



Gambar 5.113. *Data Flow Task* DIM\_ALUMNI – Full Refresh

- a. Data yang akan di-load bersumber dari server, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[TBL\_ALUMNI]

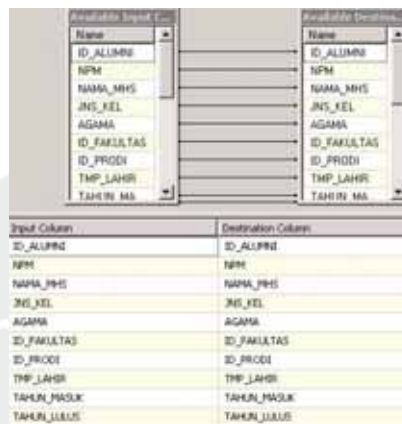
- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_ALUMNI]

Karena tabel **DIM\_ALUMNI** belum terbentuk pada database, maka buat tabel **DIM\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_ALUMNI] ([ID_ALUMNI] INT, [NPM] VARCHAR(10),
[NAMA_MHS] VARCHAR(100), [JNS_KEL] VARCHAR(1), [AGAMA]
VARCHAR(20), [ID_FAKULTAS] INT, [ID_PRODI] VARCHAR (2), [TMP_LAHIR]
VARCHAR(50), [TAHUN_MASUK] INT, [TAHUN_LULUS] INT, [GAJI_PERTAMA]
FLOAT, [AWAL_STUDI] DATETIME, [ID_ KAB_KODYA] INT, [TGL_WISUDA]
DATETIME, [TGL_YUD] DATETIME, [LAMA_STUDI] NUMERIC(4,2), [LAMA_TA]
REAL, [LAMA_ MENUNGGU_KERJA] VARCHAR(30), CONSTRAINT PK_ALUMNI
PRIMARY KEY (ID_ALUMNI), CONSTRAINT FK_ALUMNI_FAKUL FOREIGN
KEY(ID_ FAKULTAS)REFERENCES DIM_FAKULTAS_ALUMNI(ID_FAKULTAS),
CONS TRAINT FK_ALUMNI_PRODI FOREIGN KEY(ID_PRODI)REFERENCES
DIM_ PRODI_ALUMNI(ID_PRODI), CONSTRAINT FK_ALUMNI_KAB FOREIGN
KEY (ID_KAB_KODYA) REFERENCES
DIM_WILAYAH_ALUMNI(ID_KAB_KODYA))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.114. Mapping *Data Flow Task* DIM\_ALUMNI – Full Refresh

## 7. DIM\_KEAHLIAN



Gambar 5.115. *Data Flow Task* DIM\_KEAHLIAN – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[TBL\_KEAHLIAN]

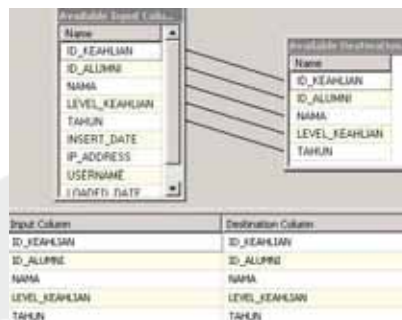
- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_KEAHLIAN\_ALUMNI]

Karena tabel **DIM\_KEAHLIAN\_ALUMNI** belum terbentuk pada *database*, maka buat tabel **DIM\_KEAHLIAN\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

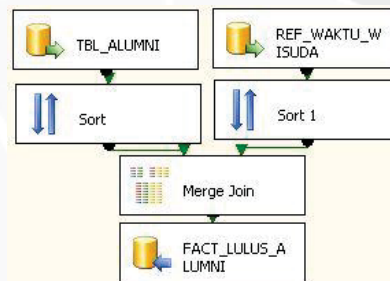
```
CREATE TABLE [DIM_KEAHLIAN_ALUMNI] ([ID_KEAHLIAN] INT,
[ID_ALUMNI] INT, [NAMA] VARCHAR(250), [LEVEL_KEAHLIAN]
VARCHAR(100), [TAHUN] INT, CONSTRAINT PK_KEAHLIAN PRIMARY KEY
(ID_KEAHLIAN), CONSTRAINT FK_KEAHLIAN_ALUMNI FOREIGN
KEY(ID_ALUMNI)REFERENCES DIM_ALUMNI(ID_ALUMNI))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.116. Mapping *Data Flow Task* DIM\_KEAHLIAN – Full Refresh

## 8. FACT\_LULUS\_ALUMNI



Gambar 5.117. *Data Flow Task* FACT\_LULUS\_ALUMNI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **TBL\_ALUMNI**

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[TBL\_ALUMNI]

b) **REF\_WAKTU\_WISUDA**

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_WAKTU\_WISUDA]

- b. Pada *data flow task* **TBL\_ALUMNI** dan **REF\_WAKTU\_WISUDA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **TGL\_WISUDA** yang telah diurutkan untuk mengambil data **ID\_WAKTU\_WISUDA**.
- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

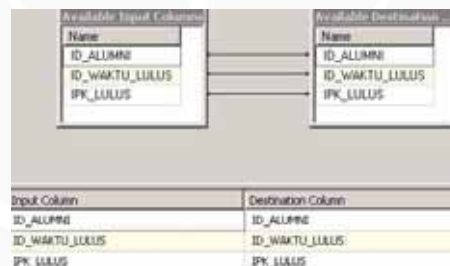


**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[FACT\_LULUS\_ALUMNI]

Karena tabel **FACT\_LULUS\_ALUMNI** belum terbentuk pada *database*, maka buat tabel **FACT\_LULUS\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [FACT_LULUS_ALUMNI] ([ID_ALUMNI] INT,
[ID_WAKTU_LULUS] INT, [IPK_LULUS] NUMERIC(4,2), CONSTRAINT
FK_LULUS_ALUMNI FOREIGN KEY (ID_ALUMNI) REFERENCES
DIM_ALUMNI(ID_ALUMNI), CONSTRAINT FK_LULUS_WAKTU FOREIGN
KEY(ID_WAKTU_LULUS) REFERENCES DIM_WAKTU_LULUS
(ID_WAKTU_LULUS))
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.118. Mapping *Data Flow Task* **FACT\_LULUS\_ALUMNI** – *Full Refresh*

## 9. DIM\_PKERJAAN



Gambar 5.119. *Data Flow Task* **DIM\_PKERJAAN**– *Full Refresh*

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[TBL\_POSISI]

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_PKERJAAN\_ALUMNI]

Karena tabel **DIM\_PKERJAAN\_ALUMNI** belum terbentuk pada *database*, maka buat tabel **DIM\_PKERJAAN\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [DIM_PKERJAAN_ALUMNI] ([ID_PKERJAAN] INT,
[ID_ALUMNI] INT, [PERUSAHAAN] VARCHAR(100), [BIDANG] NVARCHAR(100),
[JABATAN] VARCHAR(100), [DIVISI] VARCHAR(50), [THN_MASUK] INT,
[THN_KELUAR] INT, CONSTRAINT PK_PKERJAAN PRIMARY
KEY(ID_PKERJAAN), CONSTRAINT FK_PKERJAAN_ALUMNI FOREIGN
KEY(ID_ALUMNI) REFERENCES DIM_ALUMNI(ID_ALUMNI))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.120. Mapping *Data Flow Task* DIM\_PKERJAAN – Full Refresh

## 10. DIM\_ORGANISASI



Gambar 5.121. *Data Flow Task* DIM\_ORGANISASI – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[TBL\_ORGANISASI]

- b. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[DIM\_ORGANISASI\_ALUMNI]

Karena tabel **DIM\_ORGANISASI\_ALUMNI** belum terbentuk pada *database*, maka buat tabel **DIM\_ORGANISASI\_ALUMNI** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

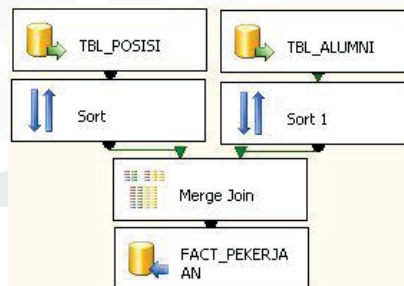
```
CREATE TABLE [DIM_ORGANISASI_ALUMNI] ([ID_ORGANISASI] INT,
[ID_ALUMNI] INT, [NAMA_ORGANISASI] VARCHAR(100), [THN_MULAI] INT,
[THN_SELESAI] INT, [JABATAN] VARCHAR(50), CONSTRAINT PK_ORGANISASI
PRIMARY KEY(ID_ORGANISASI), CONSTRAINT FK_ORGANISASI_ALUMNI
FOREIGN KEY(ID_ALUMNI) REFERENCES DIM_ALUMNI(ID_ALUMNI))
```

- c. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



Gambar 5.122. Mapping *Data Flow Task* DIM\_ORGANISASI – Full Refresh

## 11. FACT\_PKERJAAN



Gambar 5.123. *Data Flow Task* FACT\_PKERJAAN – Full Refresh

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

- a) TBL\_POSISI

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[TBL\_POSISI]

- b) TBL\_ALUMNI

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[TBL\_ALUMNI]

- b. Pada *data flow task* TBL\_POSISI dan TBL\_ALUMNI digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan

berdasarkan **ID\_ALUMNI** yang telah diurutkan untuk mengambil data **ID\_POSISI**.

- c. Mendefinisikan *data flow destination* dengan tarik dan letakkan *ADO.NET DESTINATION* dan lakukan konfigurasi berikut:

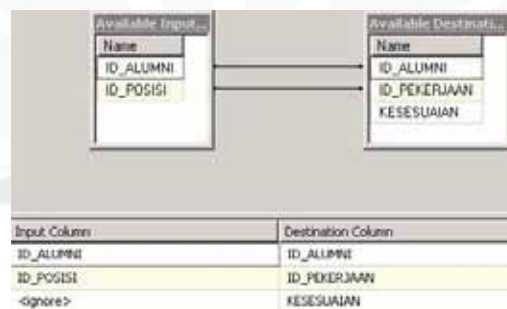
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[FACT\_Pekerjaan]

Karena tabel **FACT\_Pekerjaan** belum terbentuk pada *database*, maka buat tabel **FACT\_Pekerjaan** dengan menekan tombol *new* dan masukkan *query* sebagai berikut:

```
CREATE TABLE [FACT_Pekerjaan] ([ID_ALUMNI] INT, [ID_Pekerjaan] INT, [Kesesuaian] INT, CONSTRAINT FK_Peker_Alumni FOREIGN KEY(ID_ALUMNI)REFERENCES DIM_ALUMNI(ID_ALUMNI), CONSTRAINT FK_Peker_Peker FOREIGN KEY(ID_Pekerjaan)REFERENCES DIM_Pekerjaan_Alumni(ID_Pekerjaan))
```

- d. Lakukan pemetaan dengan menyesuaikan kolom input dan kolom output pada *data flow task* tersebut adalah sebagai berikut:



**Gambar 5.124. Mapping Data Flow Task FACT\_Pekerjaan – Full Refresh**

#### 6).Sequence Container **LOAD2**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD2**.

#### 7).Execute SQL Task pada **LOAD2**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD2** sebanyak 4 kali dan beri nama **UPDATE\_Kesesuaian**, **UPDATE\_KAB\_KODYA**, **UPDATE\_IPK**, dan **UPDATE\_TAHUN**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

- a. Ikon *Execute SQL Task* **UPDATE\_Kesesuaian** berfungsi menjalankan *query* untuk mengubah data kesesuaian pekerjaan yang ada pada *database*

**SIKMA\_DW\_ALUMNI.** Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE dbo.FACT_Pekerjaan SET KESESUAIAN=1 WHERE (ID_Pekerjaan
IN (SELECT D.ID_Pekerjaan FROM dbo.DIM_Pekerjaan_ Alumni AS D
INNER JOIN dbo.DIM_Alumni AS A ON D.ID_Alumni = A.ID_Alumni INNER
JOIN dbo.DIM_Prodi_Alumni AS PRO ON A.ID_Prodi = PRO.ID_Prodi
WHERE(D.Jabatan IN (SELECT Jabatan FROM
dbo.DIM_Pekerjaan_Prodi))));
```

- b. Ikon *Execute SQL Task* **UPDATE\_KAB\_KODYA** berfungsi untuk mengubah data id kabupaten kodya yang ada pada *database SIKMA\_DW\_ALUMNI.* Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE DIM_Alumni SET ID_KAB_KODYA=1 WHERE ID_Alumni IN(
SELECT ID_Alumni FROM DIM_Alumni WHERE ID_KAB_KODYA IS NULL);
```

- c. Ikon *Execute SQL Task* **UPDATE\_IPK** berfungsi untuk mengubah data IPK yang ada pada *database SIKMA\_DW\_ALUMNI.* Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE FACT_Lulus_Alumni SET IPK_Lulus=0 WHERE ID_Alumni IN(
SELECT ID_Alumni FROM FACT_Lulus_Alumni WHERE IPK_Lulus IS
NULL);
```

- d. Ikon *Execute SQL Task* **UPDATE\_TAHUN** berfungsi untuk menghapus data tarif selain id tarif 10 yang ada pada *database SIKMA\_DW\_ALUMNI.* Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE DIM_Pekerjaan_Alumni SET THN_Masuk=0 WHERE THN_Masuk
IS NULL; UPDATE DIM_Pekerjaan_Alumni SET THN_Keluar=0 WHERE
THN_Keluar IS NULL;
```

#### 8). *Sequence Container* **LOAD3**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD3.**

#### 9). *Execute SQL Task* **UPDATE\_KESESUAIAN1**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD3** dan beri nama **UPDATE\_KESESUAIAN1.** Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI.** Ikon *Execute SQL Task* **UPDATE\_TIDAK\_SESUIAI** berfungsi menjalankan *query* untuk mengubah

data pekerjaan yang tidak sesuai pada *database* **SIKMA\_DW\_ALUMNI**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE FACT_PKERJAAN SET KESESUAIAN=1 WHERE ID_PKERJAAN IN(
148,88,90,93,95,96,98,99,100,101,102,103,104,105,106,108,109,
110,111,112,114,115,117,118,119,122,123,124,125,126,127,129);
```

#### 10). *Sequence Container* **LOAD4**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD4**.

#### 11). *Execute SQL Task* **UPDATE\_TIDAK\_SESUAI**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD4** dan beri nama **UPDATE\_TIDAK\_SESUAI**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI**). Ikon *Execute SQL Task* **UPDATE\_KESESUAIAN1** berfungsi menjalankan *query* untuk mengubah data kesesuaian pekerjaan yang ada pada *database* **SIKMA\_DW\_ALUMNI**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE FACT_PKERJAAN SET KESESUAIAN=0 WHERE KESESUAIAN IS NULL
```

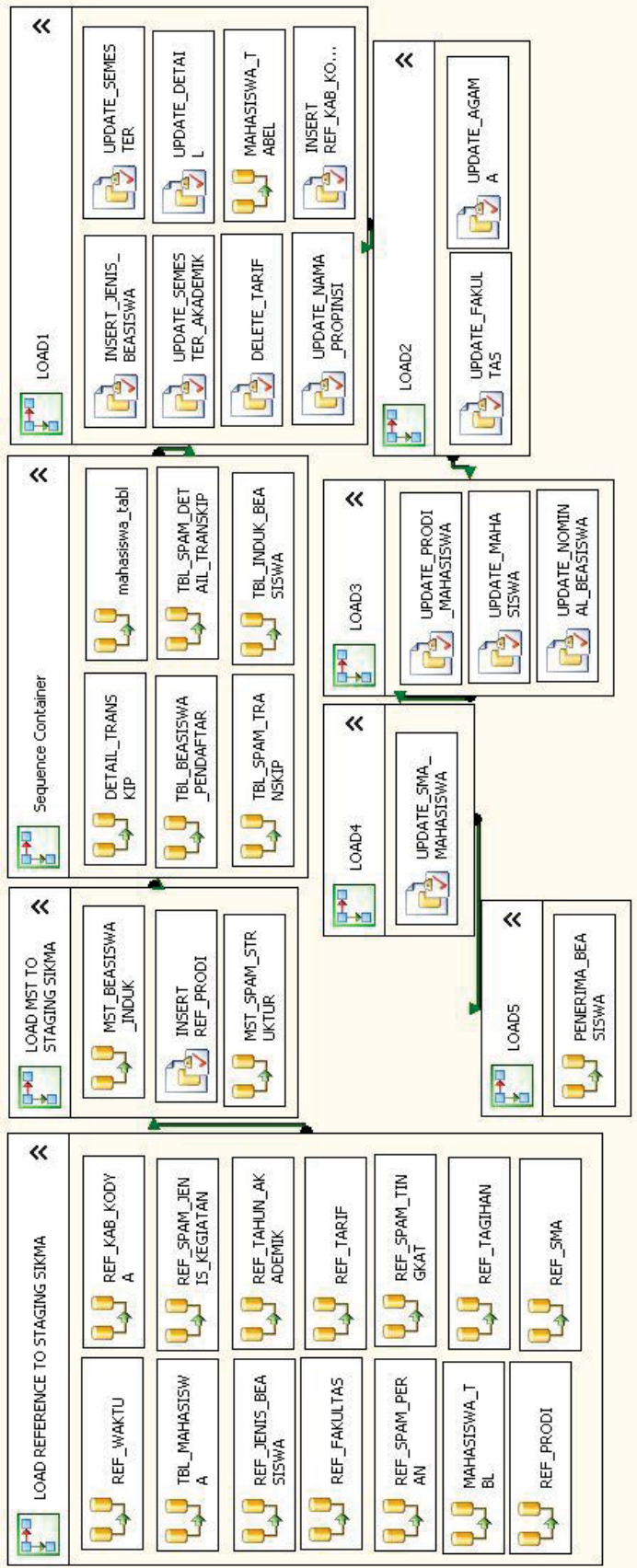
### 5.1.4. *Control Flow Incremental Load* **SIKMA**

Proses *incremental load* **SIKMA** terbagi menjadi dua proses. Proses pertama adalah mengisi data dari *database* sumber menuju *staging area*. Proses selanjutnya adalah mengisi data dari *staging area* menuju *data warehouse*.

#### 5.1.4.1. *Loading Data dari Sumber Data ke* **SIKMA\_STAGING**

Pada proses *loading* data dari *database* sumber ke dalam *database* **SIKMA\_STAGING**, langkah pertama adalah menentukan *control flow items* yang sesuai dengan kebutuhan pada *tab control flow*. Untuk menggunakan *control flow items* yang telah disediakan, tarik dan letakkan ikon *control flow items* yang diperlukan ke area tengah. Kemudian proses *loading* data dari sumber data ke **SIKMA\_STAGING** pada *package* *Staging\_incremental\_sikma.dtsx* akan terbentuk desain proses sebagai berikut :





Gambar 5.125. Desain Control Flow pada Staging\_incremental\_sikma – Incremental Load

Untuk membuat desain *Control Flow* di atas, langkah-langkah yang dilakukan adalah sebagai berikut:

1). *Sequence Container* **LOAD REFERENCE TO STAGING SIKMA**

Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD REFERENCE TO STAGING SIKMA**.

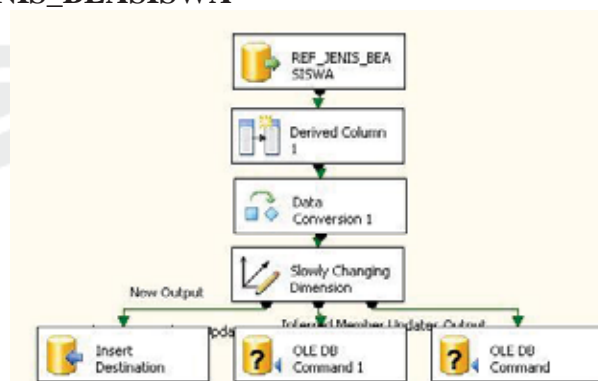
2). *Data Flow Task* **LOAD REFERENCE TO STAGING SIKMA**

*Control flow item* yang digunakan untuk ekstrak, *transform*, dan *load* data adalah *data flow task*. Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* **LOAD REFERENCE TO STAGING SIKMA**. Karena ada 14 data sumber yang akan di-*load* maka dibuat 14 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **REF\_JENIS\_BEASISWA**, **REF\_PRODI**, **REF\_FAKULTAS**, **REF\_SMA**, **REF\_WAKTU**, **REF\_TARIF**, **MAHASIWA\_TBL**, **REF\_SPAM\_JENIS\_KEGIATAN**, **REF\_SPAM\_TINGKAT**, **REF\_SPAM\_PERAN**, **REF\_KAB\_KODYA**, **REF\_TAHUN\_AKADEMIK**, **REF\_TAGIHAN**, dan **TBL\_MAHASIWA**.

3). *Data Flow Task* pada *Sequence Container* **LOAD REFERENCE TO STAGING**

Langkah-langkah untuk membuat *data flow task* pada **LOAD REFERENCE TO STAGING** sebagai berikut:

**1. REF\_JENIS\_BEASISWA**



Gambar 5.126. *Data Flow Task* REF\_JENIS\_BEASISWA – *Incremental Load*

a. Data yang akan di-*load* bersumber dari *server*, maka tarik dan letakkan ikon *OLE DB SOURCE* dan lakukan konfigurasi sebagai berikut:

Connection Manager : DATA-WAREHOUSE.siatmax  
 Data Access Mode : Table or view  
 Name of table or view : dbo],[REF\_JENIS\_BEASISWA]

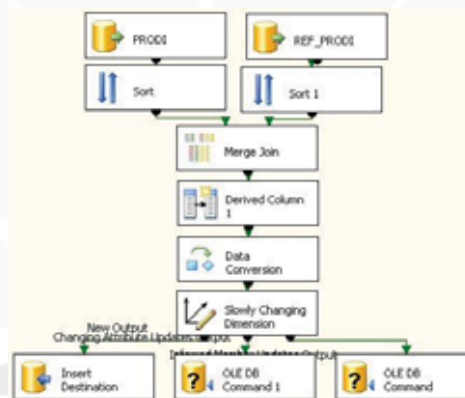
- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.
- c. Fungsi *data conversion* digunakan untuk mengkonversi **JENIS\_BEASISWA** menjadi data *varchar*.
- d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_JENIS\_BEASISWA]

Pilih **ID\_JENIS\_BEASISWA** sebagai business key. Pilih tipe *changing attribute* untuk kolom **JENIS\_BEASISWA**.

## 2. REF\_PRODI



Gambar 5.127. Data Flow Task REF\_PRODI – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

### a) REF\_PRODI

**Connection Manager** :DATA-WAREHOUSE.siatmax

**Data Access Mode** :Table or view

**Name of table or view** :[dbo].[REF\_PRODI]

### b) PRODI

**Connection Manager** : DATA-WAREHOUSE.biadw

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[prodi]

- b. Pada *Data Flow Task* **REF\_PRODI** dan **REF\_PRODI1** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Kemudian data digabungkan berdasarkan **id\_prodi** yang telah diurutkan untuk mengambil data **nama\_prodi**.

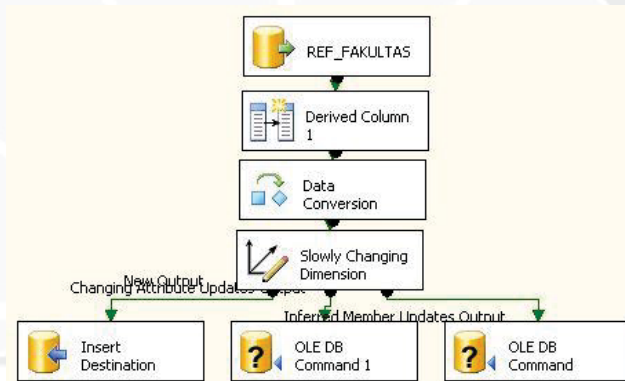
- c. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.
- d. Fungsi *data conversion* digunakan untuk mengkonversi **ID\_PRODI** menjadi data *varchar*.
- e. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*.  
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_PRODI]

Pilih **ID\_PRODI** dan **ID\_FAKULTAS** sebagai business key. Pilih tipe changing attribute untuk kolom **BHS**, **JENJANG**, **KDPRODIDIKTI**, **PRODI**, dan **PRODI\_ING**.

### 3. REF\_FAKULTAS



**Gambar 5.128. Data Flow Task REF\_FAKULTAS – Incremental Load**

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[REF\_FAKULTAS]

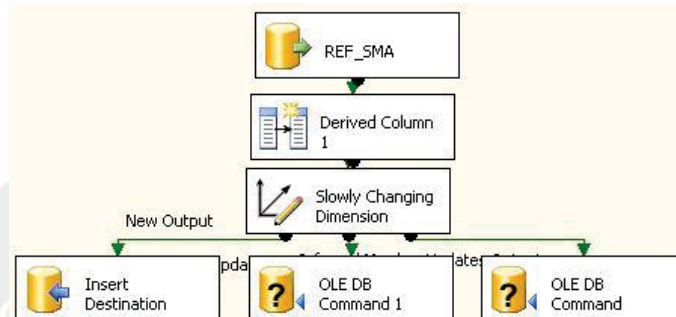
- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.
- c. Fungsi *data conversion* digunakan untuk mengkonversi **ID\_PRODI** menjadi data *varchar*.
- d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*.  
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_FAKULTAS]

Pilih **ID\_FAKULTAS** sebagai business key. Pilih tipe changing attribute untuk kolom **ALAMAT\_FAKULTAS**, **FAKULTAS**, **FAKULTAS\_ING**, **FAX\_FAKULTAS**, dan **TELP\_FAKULTAS**.

#### 4. REF\_SMA



**Gambar 5.129. Data Flow Task REF\_SMA – Incremental Load**

- Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.biadw

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[sma]

- Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.

- Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

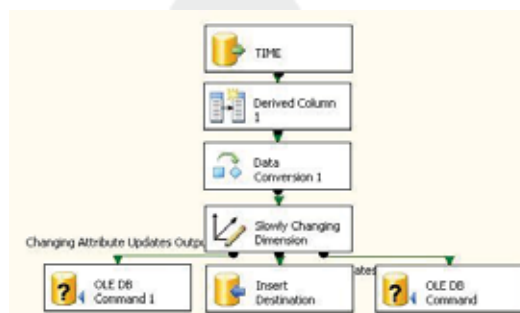
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_SMA]

Pilih **ID\_SMA** sebagai business key. Pilih tipe changing attribute untuk kolom **ALAMAT**, **ID\_KAB\_SMA**, **ID\_SMA**, dan **NAMA\_SMA**.

#### 5. REF\_WAKTU



**Gambar 5.130. Data Flow Task REF\_WAKTU – Incremental Load**

- Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:



**Connection Manager** : DATA-WAREHOUSE.biadw

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[time]

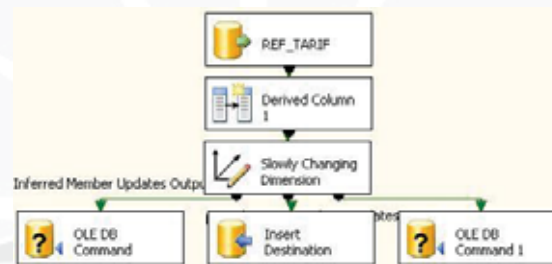
- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.
- c. Fungsi *data conversion* digunakan untuk mengkonversi **SEMESTER** menjadi data *varchar*.
- d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data converion*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_WAKTU]

Pilih **ID\_WAKTU** sebagai business key. Pilih tipe changing attribute untuk kolom **ID\_TAHUN\_AKADEMIK**, **NO\_SEMESTER**, dan **SEMESTER**.

## 6. REF\_TARIF



**Gambar 5.130.** Data Flow Task REF\_TARIF – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:
- Connection Manager** : DATA-WAREHOUSE.sikeuDB
- Data Access Mode** : Table or view
- Name of table or view** : [dbo].[trnsTarif]
- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*.
  - c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

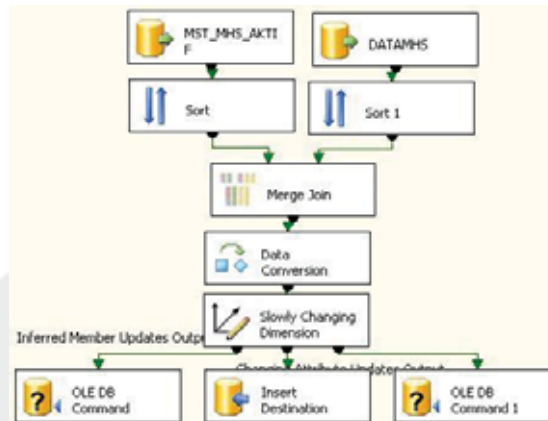
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_TARIF]

Pilih **THN\_MASUK**, **SEMESTER**, dan **MASUK** sebagai business key. Pilih tipe changing attribute untuk kolom **BIAYA**, **ISCURRENT**, dan **TAGIHAN**.



## 7. MAHASISWA\_TBL



**Gambar 5.132.** *Data Flow Task MAHASISWA\_TBL – Incremental Load*

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**a) DATAMHS**

**Connection Manager** : DATA-WAREHOUSE.siatma

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[datamhs1]

**b) MST\_MHS\_AKTIF**

**Connection Manager** : DATA-WAREHOUSE.siatma

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[MST\_MHS\_AKTIF]

b. Pada *data flow task* **DATAMHS** dan **MST\_MHS\_AKTIF** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **NAMA\_MHS**, **TMP\_LAHIR**, **JNS\_KEL**, dan **AGAMA**.

c. Fungsi *data conversion* digunakan untuk mengkonversi **NPM** menjadi data *varchar*.

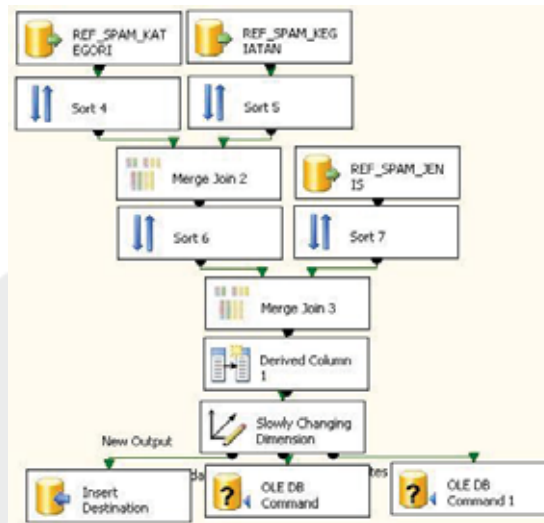
d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[MAHASISWA\_TBL]

Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **AGAMA**, **JNS\_KEL**, **NAMA\_MHS**, **TGL\_LAHIR**, dan **THN\_MASUK**.

## 8. REF\_SPAM\_JENIS\_KEGIATAN



Gambar 5.133. Data Flow Task REF\_SPAM\_JENIS\_KEGIATAN – Incremental Load

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **REF\_SPAM\_KEGIATAN**

Connection Manager : DATA-WAREHOUSE.siatmax  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[REF\_SPAM\_KEGIATAN]

b) **REF\_SPAM\_KATEGORI**

Connection Manager : DATA-WAREHOUSE.siatmax  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[REF\_SPAM\_KATEGORI]

c) **REF\_SPAM\_JENIS**

Connection Manager : DATA-WAREHOUSE.siatmax  
 Data Access Mode : Table or view  
 Name of the table or view : [dbo].[REF\_SPAM\_JENIS]

b. Pada *data flow task* **REF\_SPAM\_KEGIATAN** dan **REF\_SPAM\_KATEGORI** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE\_KATEGORI** yang telah diurutkan untuk mengambil data **KETERANGAN\_KATEGORI**.

c. Pada *data flow task* **REF\_SPAM\_JENIS** dan **MERGE JOIN** digabungkan dengan fungsi *merge join*. Masing-masing harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan

**KODE\_KEGIATAN** yang telah diurutkan untuk mengambil data **KETERANGAN\_KEGIATAN**.

- d. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- e. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

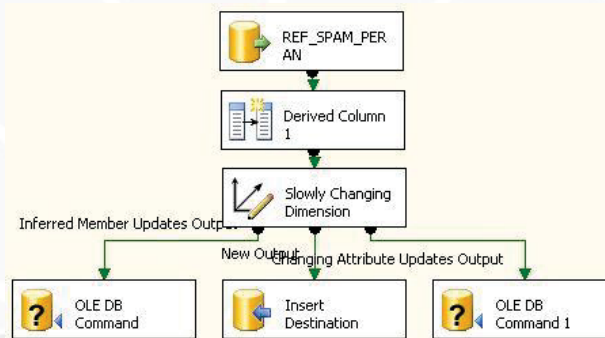
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_SPAM\_JENIS\_KEGIATAN]

Pilih **KODE\_JENIS**, **KODE\_KATEGORI**, dan **KODE\_KEGIATAN** sebagai business key. Pilih tipe changing attribute untuk kolom **KETERANGAN\_JENIS**, **KETERANGAN\_KATEGORI**, dan **KETERANGAN\_KEGIATAN**.

## 9. REF\_SPAM\_PERAN



**Gambar 5.134. Data Flow Task REF\_SPAM\_PERAN – Incremental Load**

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_SPAM\_PERAN]

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

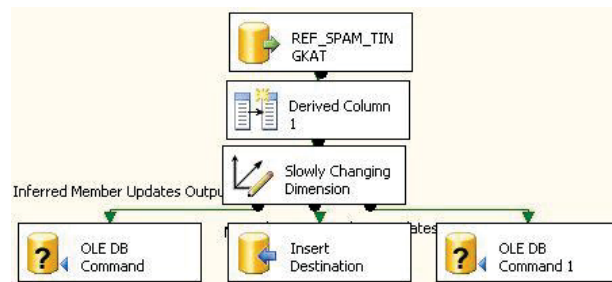
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_SPAM\_PERAN]

Pilih **KODE\_PERAN** sebagai business key. Pilih tipe changing attribute untuk kolom **KETERANGAN\_PERAN**.

## 10. REF\_SPAM\_TINGKAT



Gambar 5.135. Data Flow Task REF\_SPAM\_TINGKAT – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_SPAM\_TINGKAT]

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

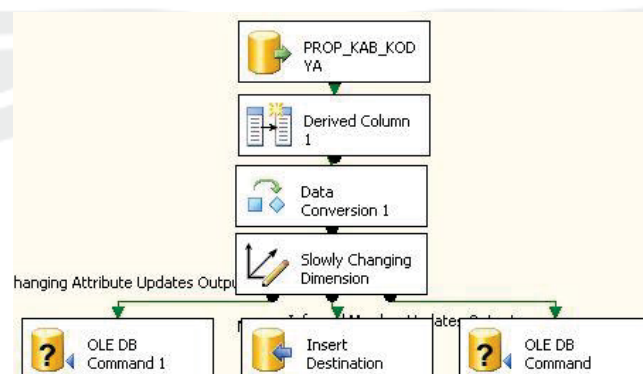
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_SPAM\_TINGKAT]

Pilih **KODE\_TINGKAT** sebagai business key. Pilih tipe changing attribute untuk kolom **KETERANGAN\_TINGKAT**.

## 11. REF\_KAB\_KODYA



Gambar 5.136. Data Flow Task REF\_KAB\_KODYA – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.biadw

**Data Access Mode** : Table or view

- Name of the table or view : [dbo].[prop\_kab\_kodya]
- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
  - c. Fungsi *data conversion* digunakan untuk mengkonversi **ID\_KAB\_KODYA** menjadi data *integer*.
  - d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*. Lakukan konfigurasi berikut:
 

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING

Use a Table or View : [dbo].[REF\_KAB\_KODYA]

Pilih **ID\_KAB\_KODYA** sebagai business key. Pilih tipe changing attribute untuk kolom **NAMA\_KAB\_KODYA** dan **NAMA\_PROPINSI**.

## 12. REF\_TAHUN\_AKADEMIK



Gambar 5.137. Data Flow Task REF\_TAHUN\_AKADEMIK – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:
  - a) **TBL\_TAHUN\_AKADEMIK**

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

Name of table or view : [dbo].[TBL\_TAHUN\_AKADEMIK]
  - b) **TBL\_SEMESTER\_AKADEMIK**

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

Name of table or view : [dbo].[TBL\_SEMESTER\_AKADEMIK]
- b. Pada *data flow task* **TBL\_TAHUN\_AKADEMIK** dan **TBL\_SEMESTER\_AKADEMIK** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus di urutkan menggunakan



fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID\_TAHUN\_AKADEMIK** yang telah diurutkan untuk mengambil data **TAHUN\_AKADEMIK**.

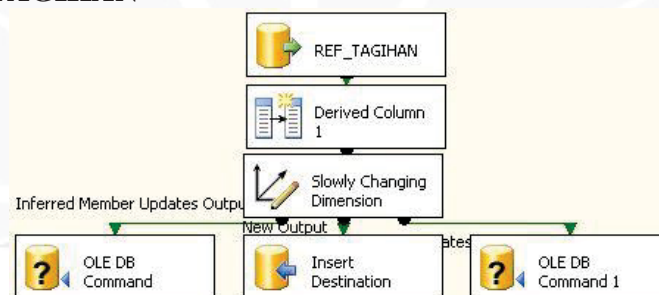
- c. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- d. Fungsi *data conversion* digunakan untuk mengkonversi **ID\_TAHUN\_AKADEMIK** menjadi data *integer*.
- e. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*.  
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_TAHUN\_AKADEMIK]

Pilih **ID\_TAHUN\_AKADEMIK** sebagai business key. Pilih tipe changing attribute untuk kolom **TAHUN\_AKADEMIK**.

### 13. REF\_TAGIHAN



**Gambar 5.138. Data Flow Task REF\_TAGIHAN – Incremental Load**

- a. Pilih *ADO.NET Source* karena data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.sikeuDB

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[refTagihan]

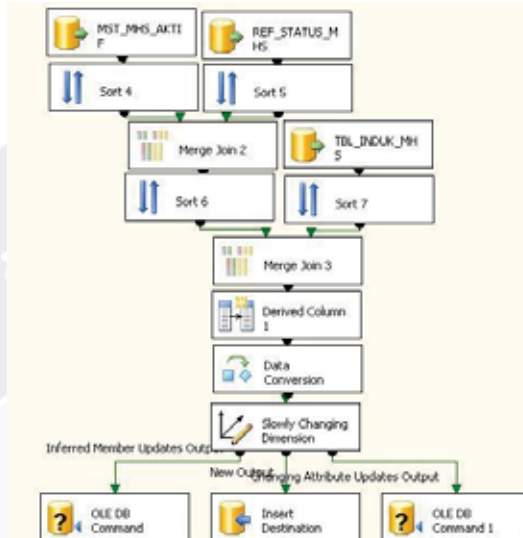
- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.  
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_TAGIHAN]

Pilih **IDNYA** sebagai business key. Pilih tipe changing attribute untuk kolom **KETERANGAN**, **NAMA**, **PR\_DENDA**, **REK\_DENDA**, **REKENING**, dan **REKENING\_BARU**.

#### 14. TBL\_MAHASISWA



Gambar 5.139. Data Flow Task TBL\_MAHASISWA – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

##### a) TBL\_INDUK\_MHS

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

Name of the table or view : [dbo].[tbl\_induk\_mhs\_3]

##### b) MST\_MHS\_AKTIF

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

Name of the table or view : [dbo].[MST\_MHS\_AKTIF]

##### c) REF\_STATUS\_MAHASISWA

Connection Manager : DATA-WAREHOUSE.siatma

Data Access Mode : Table or view

Name of the table or view : [dbo].[REF\_STATUS\_MAHASISWA]

- b. Pada *data flow task* **MST\_MHS\_AKTIF** dan **REF\_STATUS\_MAHASISWA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KD\_STATUS\_MHS** yang telah diurutkan untuk mengambil data **STATUS\_MHS**.

- c. Pada *data flow task* **TBL\_INDUK\_MHS** dan **MERGE JOIN** digabungkan dengan fungsi *merge join*. Masing-masing harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **ID\_FAKULTAS**, **ID\_PRODI**, dan **Nama\_MHS**.
- d. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- e. Fungsi *data conversion* digunakan untuk mengkonversi **ID\_FAKULTAS** dan **ID\_PRODI** menjadi data *varchar*.
- f. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*.  
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[TBL\_MAHASISWA]

Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **AGAMA**, **ID\_FAKULTAS**, **ID\_PRODI**, **JENIS\_KELAMIN**, **NAMA\_MAHASISWA**, **STATUS\_MHS**, **TANGGAL\_LAHIR** dan **TEMPAT\_LAHIR**.

4). **Sequence Container LOAD MASTER TO STAGING SIKMA**

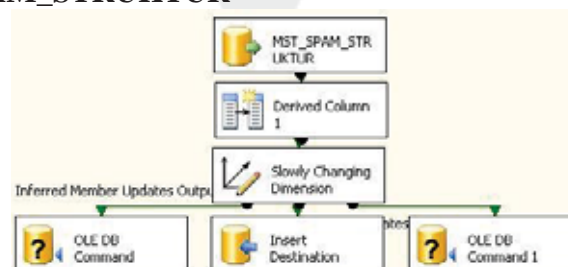
Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD MASTER TO STAGING SIKMA**.

5). **Data Flow Task LOAD MASTER TO STAGING SIKMA**

Tarik dan letakkan ikon *data flow task* ke dalam item **LOAD MASTER TO STAGING SIKMA** *sequence container* yang telah dibuat dan beri nama **MST\_SPAM\_STRUKTUR**. Lakukan hal yang sama untuk membuat *data flow task* **MST\_BEASISWA\_INDUK**.

6). **Data Flow Task pada Sequence Container LOAD MASTER TO STAGING SIKMA**

**1. MST\_SPAM\_STRUKTUR**



Gambar 5.140. Data Flow Task MST\_SPAM\_STRUKTUR – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[MST\_SPAM\_STRUKTUR]

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[MST\_SPAM\_STRUKTUR]

Pilih **NO\_URUT** sebagai business key. Pilih tipe changing attribute untuk kolom

**KODE\_JENIS, KODE\_KATEGORI, KODE\_KEGIATAN, KODE\_PERAN, KODE\_TINGKAT,** dan **MASTER\_POIN**.

## 2. MST\_BEASISWA\_INDUK



Gambar 5.141. Data Flow Task MST\_BEASISWA\_INDUK – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

- a) REF\_BEASISWA\_KATEGORI

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[REF\_BEASISWA\_KATEGORI]

- b) MST\_BEASISWA\_INDUK

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[MST\_BEASISWA\_INDUK]

- b. Pada *data flow task* **REF\_BEASISWA\_KATEGORI** dan **MST\_BEASISWA\_INDUK** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NO\_KATEGORI** yang telah diurutkan untuk mengambil data **KETERANGAN KATEGORI**
- c. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- d. Fungsi *data conversion* digunakan untuk mengkonversi **NO\_BEASISWA** menjadi data *integer*.
- e. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[MST\_BEASISWA\_INDUK]

Pilih **NO\_BEASISWA** sebagai business key. Pilih tipe changing attribute untuk kolom **ID\_SPONSOR**, **JUMLAH\_TERMIN**, **NAMA\_BEASISWA**, **PERIODE**, **QUOTA**, dan **STATUS**.

7). *Execute SQL Task* **LOAD MASTER TO STAGING SIKMA**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD MASTER TO STAGING SIKMA** dan beri nama **INSERT REF\_PRODI**. Ikon *Execute SQL Task* ini berfungsi menjalankan *query* untuk menambah data **ref\_prodi** dan **ref\_fakultas** yang ada pada *database* **SIKMA\_STAGING**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_STAGING**).

Mendefinisikan *SQL Statement* pada item **LOAD MASTER TO STAGING SIKMA**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT REF_PRODI VALUES(25,7,NULL,NULL,NULL,NULL, NULL,NULL); INSERT
INTO REF_FAKULTAS VALUES(7,NULL,NULL,NULL,NULL,NULL, NULL);
```

Sintaks di atas digunakan untuk menambah data **ref\_prodi** dan **ref\_fakultas** yang ada pada *database* **SIKMA\_STAGING**.

8). *Sequence Container* **LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA**



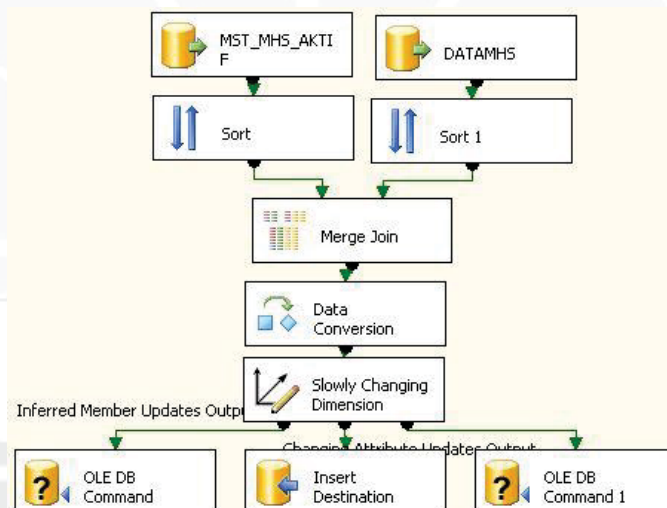
Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA**.

9). *Data Flow Task* **LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA**

Tarik dan letakkan ikon *data flow task* ke dalam item **LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA** *sequence container* yang telah dibuat dan beri nama **TBL\_SPAM\_TRANSKIP**. Lakukan hal yang sama untuk membuat *data flow task* **mahasiswa\_tabl**, **TBL\_SPAM\_DETAIL\_TRANSKIP**, **TBL\_PENERIMA\_BEASISWA**, **TBL\_INDUK\_BEASISWA**, dan **DETAIL\_TRANSKIP**.

10). *Data Flow Task* pada *Sequence Container* **LOAD TRANSAKSIONAL TABEL TO STAGING SIKMA**

1. mahasiswa\_tabl



Gambar 5.142. *Data Flow Task* mahasiswa\_tabl – Incremental Load

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **MST\_MHS\_AKTIF**

Connection Manager : DATA-WAREHOUSE.siatma  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[MST\_MHS\_AKTIF]

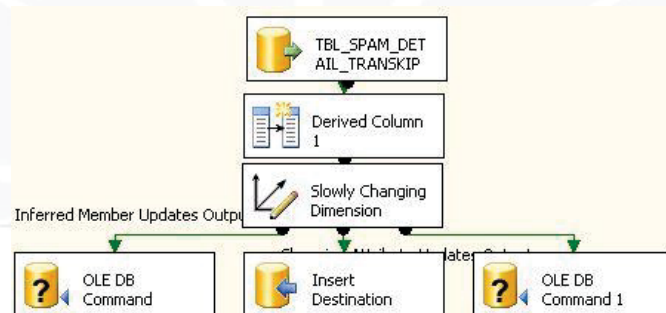
b) **DATAMHS**

Connection Manager : DATA-WAREHOUSE.siatma  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[datamhs1]

- b. Pada *data flow task* **MST\_MHS\_AKTIF** dan **DATAMHS** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **THN\_MASUK** dan **TGL\_LAHIR**
- c. Fungsi *data conversion* digunakan untuk mengkonversi **NPM** menjadi data *varchar*.
- d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[MAHASISWA\_TBL]  
 Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **AGAMA**, **JNS\_KEL**, **NAMA\_MHS**, **TGL\_LAHIR**, dan **THN\_MASUK**.

## 2. TBL\_SPAM\_DETAIL\_TRANSKIP



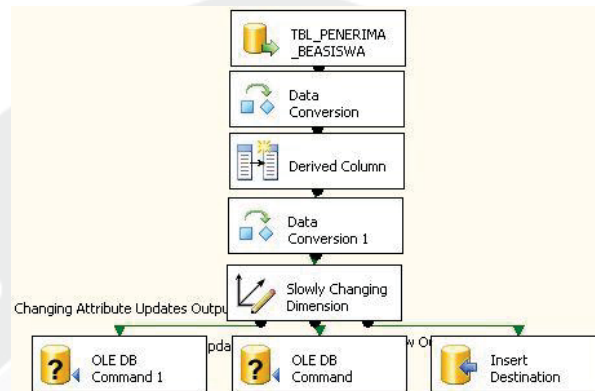
Gambar 5.143. *Data Flow Task* TBL\_SPAM\_DETAIL\_TRANSKIP – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:
- Connection Manager** : DATA-WAREHOUSE.siatmax  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[TBL\_SPAM\_DETAIL\_TRANSKIP]
- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *derived column*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[TBL\_SPAM\_DETAIL\_TRANSKIP]

Pilih **NO\_URUT\_DETAIL** sebagai business key. Pilih tipe changing attribute untuk kolom **ID\_TAHUN\_AKADEMIK**, **ID\_TRANSKIP\_SPAM**, **KODE\_JENIS**, **KODE\_KATEGORI**, **KODE\_KEGIATAN**, **KODE\_PERAN**, **KODE\_TINGKAT**, **NO\_SEMESTER**, dan **POIN**.

### 3. TBL\_PENERIMA\_BEASISWA



**Gambar 5.144. Data Flow Task TBL\_PENERIMA\_BEASISWA – Incremental Load**

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[TBL\_PENERIMA\_BEASISWA]

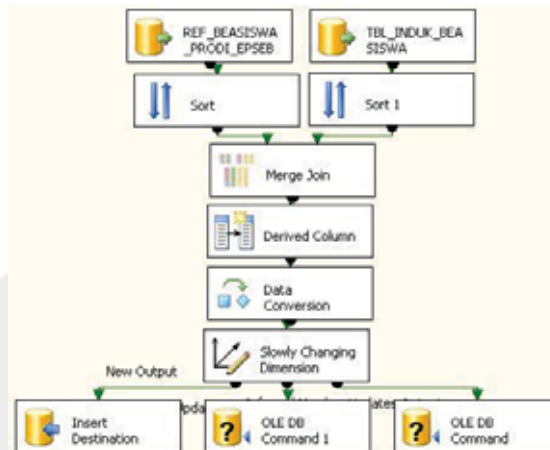
- b. Fungsi *data conversion* digunakan untuk mengkonversi **NAMA\_MHS** menjadi data *varchar*.  
 c. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.  
 d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[TBL\_PENERIMA\_BEASISWA]

Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **JENIS\_BEASISWA**, **NAMA\_MHS**, **PRODI**, dan **SEMESTER**.

#### 4. TBL\_INDUK\_BEASISWA



Gambar 5.145. Data Flow Task TBL\_INDUK\_BEASISWA – Incremental Load

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**a) REF\_BEASISWA\_PRODI\_EPSBED**

**Connection Manager** : DATA-WAREHOUSE.siatmax  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[REF\_BEASISWA\_PRODI\_EPSBED]

**b) TBL\_INDUK\_BEASISWA**

**Connection Manager** : DATA-WAREHOUSE.siatmax  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[TBL\_INDUK\_BEASISWA]

b. Pada *data flow task* **REF\_BEASISWA\_PRODI\_EPSBED** dan **TBL\_INDUK\_BEASISWA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus di urutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID\_PRODI** yang telah diurutkan untuk mengambil data **ID\_PRODI\_EPSBED**

c. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.

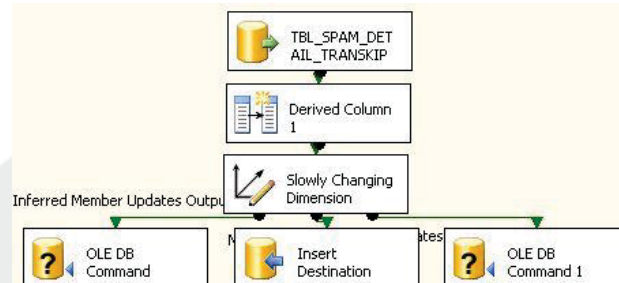
d. Fungsi *data conversion* digunakan untuk mengkonversi **NPM** menjadi data *varchar*.

e. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[TBL\_INDUK\_BEASISWA]

Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **CABANG**, **COUNTER**, **ID\_JENIS\_BEASISWA**, **ID\_JENIS\_KELAMIN**, **ID\_JENJANG**, **ID\_PKERJAAN**, **IPK**, **TAHUN**, **JML\_TANGGUNGAN**, dan **SEMESTER**.

## 5. DETAIL\_TRANSKIP



**Gambar 5.146. Data Flow Task DETAIL\_TRANSKIP – Incremental Load**

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[TBL\_SPAM\_DETAIL\_TRANSKIP]

- b. Tambahkan kolom waktu dimana data dimasukkan ke dalam *database* sebagai kelengkapan historis data dengan fungsi *derived column*.
- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[DETAIL]

Pilih **NO\_URUT\_DETAIL** sebagai business key. Pilih tipe changing attribute untuk kolom **ID\_TAHUN\_AKADEMIK**, **ID\_TRANSKIP\_SPAM**, **KODE\_JENIS**, **KODE\_KATEGORI**, **KODE\_KEGIATAN**, **KODE\_PERAN**, **KODE\_TINGKAT**, **NO\_SEMESTER**, **PENYELENGGARA**, dan **POIN**.

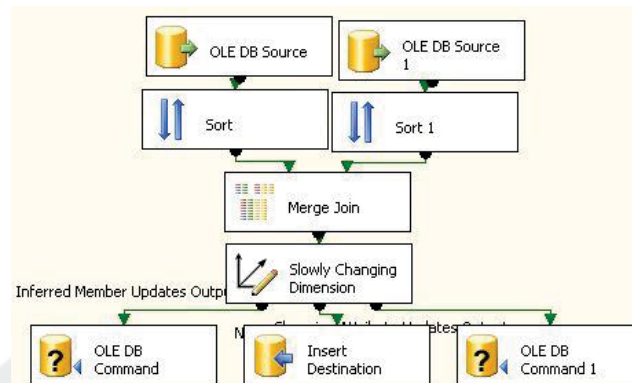
### 11). Sequence Container **LOAD1**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD1**.

### 12). Data Flow Task **LOAD1**

Tarik dan letakkan ikon *Data Flow Task* ke dalam item **LOAD1 Sequence Container** yang telah dibuat sebelumnya dan beri nama **MAHASISWA\_TABEL**.





Gambar 5.147. Data Flow Task MAHASISWA\_TABEL – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) MAHASISWA\_TABL

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[Mahasiswa\_tabl]

b) MAHASISWA

**Connection Manager** : DATA-WAREHOUSE.biadw  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[mahasiswa]

- b. Pada *data flow task* MAHASISWA\_TABL dan MAHASISWA digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus di urutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **ID\_KAB\_ORTU** dan **ID\_SMA**.

- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *merge join*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[MAHASISWA\_TABEL]

Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **AGAMA**, **ID\_KAB\_MHS**, **JNS\_KEL**, **NAMA\_MHS**, **TGL\_LAHIR**, **THN\_MASUK**, dan **TMP\_LAHIR**.

13). *Execute SQL Task* pada **LOAD1**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD1** Sequence Container sebanyak 7 kali dan beri nama **INSERT\_JENIS\_BEASISWA**,

**UPDATE\_NAMA\_PROPINSI**, **UPDATE\_SEMESTER\_AKADEMIK**, **DELETE\_TARIF**, **INSERT\_REF\_KAB\_KODYA**, **UPDATE\_SEMESTER**, dan **UPDATE\_DETAIL**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_STAGING**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

- a. Ikon *Execute SQL Task* **INSERT\_JENIS\_BEASISWA** berfungsi menjalankan *query* untuk menambah data jenis beasiswa yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT INTO REF_JENIS_BEASISWA VALUES(3,'SPP 50% Tetap', NULL); INSERT
INTO REF_JENIS_BEASISWA VALUES(4,'SPP Tetap', NULL); INSERT INTO
REF_JENIS_BEASISWA VALUES(5,'Mandiri', NULL); INSERT INTO
REF_JENIS_BEASISWA VALUES (6,'Djarum', NULL);INSERT INTO
REF_JENIS_BEASISWA VALUES(7,'BRI',NULL); INSERT INTO REF_JENIS_
BEASISWA VALUES(8,'Toyota',NULL); INSERT INTO REF_JENIS_BEASISWA
VALUES(9,'Vivere',NULL); INSERT INTO REF_JENIS_BEASISWA VALUES
(10,'SWA',NULL); INSERT INTO REF_JENIS_BEASISWA
VALUES(11,'YPKAAR',NULL)
```

- b. Ikon *Execute SQL Task* **UPDATE\_NAMA\_PROPINSI** berfungsi untuk mengubah data nama propinsi yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE REF_KAB_KODYA SET NAMA_PROPINSI='Nanggroe Aceh Darusalam'
WHERE ID_KAB_KODYA IN(SELECT ID_KAB_KODYA WHERE
NAMA_PROPINSI='NANGROE ACEH DARUSALAM'); UPDATE
REF_KAB_KODYA SET NAMA_PROPINSI='Bangka Belitung' WHERE
ID_KAB_KODYA IN(SELECT ID_KAB_KODYA WHERE NAMA_PROPINSI=
'KEPULAUAN BANGKA BELITUNG');
```

- c. Ikon *Execute SQL Task* **UPDATE\_SEMESTER\_AKADEMIK** berfungsi untuk mengubah data semester yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE REF_WAKTU_PRESTASI SET SEMESTER=1 WHERE DateKey IN(
SELECT DateKey FROM REF_WAKTU_ PRESTASI WHERE QuarterOfYear IN(1,2));
UPDATE REF_WAKTU_PRESTASI SET SEMESTER=2 WHERE DateKey
```

```
IN(SELECT DateKey FROM REF_WAKTU_PRESTASI WHERE QuarterOfYear
IN(3,4));UPDATE REF_WAKTU_PRESTASI SET SEMESTER _AKADEMIK=2
WHERE DateKey IN(SELECT DateKey FROM REF_WAKTU_ PRESTASI WHERE
Quarter OfYear IN(1,2));UPDATE REF_WAKTU_PRES TASI SET
SEMESTER_AKADEMIK=1 WHERE DateKey IN(SELECT DateKey FROM
REF_WAKTU_PRESTASI WHERE Quarter OfYear IN(3,4));
```

- d. Ikon *Execute SQL Task* **DELETE\_TARIF** berfungsi untuk menghapus data tarif selain id tarif 10 yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
DELETE FROM REF_TARIF WHERE tagihan !=10
```

- e. Ikon *Execute SQL Task* **INSERT REF\_KAB\_KODYA** berfungsi untuk menambah data kabupaten dan kodya yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT INTO REF_KAB_KODYA VALUES(533,NULL,NULL, NULL);INSERT INTO
REF_KAB_KODYA VALUES(475,NULL, NULL,NULL);INSERT INTO
REF_KAB_KODYA VALUES(130, NULL,NULL,NULL);INSERT INTO
REF_KAB_KODYA VALUES(474,NULL,NULL,NULL);
```

- f. Ikon *Execute SQL Task* **UPDATE\_SEMESTER** berfungsi untuk mengubah data semester yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE REF_WAKTU SET SEMESTER='Ganjil' WHERE ID_WAKTU IN(SELECT
ID_WAKTU WHERE SEMESTER= 'Gasal')
```

- g. Ikon *Execute SQL Task* **UPDATE\_DETAIL** berfungsi untuk mengubah data detail transkrip spama yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE TBL_SPAM_DETAIL_TRANSKIP SET NAMA_KEGIATAN= 'LDKPM_
UAJY' WHERE NO_URUT_DETAIL IN (SELECT NO_URUT_DETAIL FROM
TBL_SPAM_DETAIL_TRANSKIP WHERE NAMA_KEGIATAN= 'LDPKM');UPDATE
TBL_SPAM_DETAIL_TRANSKIP SET NAMA_KEGIATAN= 'INISIASI_UAJY'
WHERE NO_URUT_DETAIL IN(SELECT NO_URUT_DETAIL FROM TBL_SPAM_
DETAIL_TRANSKIP WHERE NAMA_KEGIATAN= 'INISIASI ');UPDATE TBL_
SPAM_DETAIL_TRANSKIP SET NAMA_KEGIATAN= 'SEMI NAR INTEGRITAS
ENTREPRENEURSHIP HMPSM UAJY' WHERE NO_URUT_DETAIL IN(SELECT
NO_URUT_DETAIL FROM TBL_SPAM_DETAIL_TRANS KIP WHERE NAMA_
KEGIATAN LIKE'SEMINAR INTEGRITAS ENT%');UPDATE TBL_SPAM_
```

```
DETAIL_TRANSKIP SET DESKRIPSI= 'GREEN_ACTION' WHERE NO_URUT_
DETAIL IN(SELECT NO_URUT_DETAIL FROM TBL_SPAM_DETAIL_ TRANSKIP
WHERE NAMA_KEGIATAN LIKE'GREEN ACT%');
```

14). *Sequence Container* **LOAD2**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD2**.

15). *Execute SQL Task* **LOAD2**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD2** sebanyak 2 kali dan beri nama **UPDATE\_AGAMA** dan **UPDATE\_FAKULTAS**. Pilih koneksi tersebut (**DATA-WAREHOUSE. SIKMA\_STAGING**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

- a. Ikon *Execute SQL Task* **UPDATE\_AGAMA** berfungsi untuk mengubah data agama yang ada pada *database* **SIKMA\_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE MAHASISWA_TABEL SET AGAMA='BUDDHISM' WHERE NPM
IN(SELECT NPM FROM MAHASISWA_TABEL WHERE AGAMA=
'Budha');UPDATE MAHASISWA_TABEL SET AGAMA='CHRISTIAN' WHERE
NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE AGAMA=
'Protestan');UPDATE MAHASISWA_TABEL SET AGAMA='CATHOLIK' WHERE
NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE AGAMA=
'Katolik');UPDATE MAHASISWA_TABEL SET AGAMA='MOHAMMEDANISM'
WHERE NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE
AGAMA='Islam');UPDATE MAHASISWA_TABEL SET AGAMA='HINDUISM'
WHERE NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE
AGAMA='Hindu');UPDATE MAHASISWA_TABEL SET AGAMA='N/A' WHERE
NPM IN(SELECT NPM FROM MAHASISWA_TABEL WHERE AGAMA IS NULL);
```

- b. Ikon *Execute SQL Task* **UPDATE\_FAKULTAS** berfungsi untuk mengubah data fakultas yang ada pada *database* **SIKMA\_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE TBL_MAHASISWA SET ID_FAKULTAS=5 WHERE ID_PRODI
IN(01,02,13);UPDATE TBL_MAHASISWA SET ID_FAKULTAS=1 WHERE
ID_PRODI IN(03,04,11,12,15, 20);UPDATE TBL_MAHASISWA SET
ID_FAKULTAS=2 WHERE ID_PRODI IN(05,21);UPDATE TBL_MAHASISWA SET
ID_FAKULTAS=3 WHERE ID_PRODI =08;UPDATE TBL_MAHASISWA SET
ID_FAKULTAS=4 WHERE ID_PRODI IN(06,07,14,24);UPDATE
TBL_MAHASISWA SET ID_FAKULTAS=6 WHERE ID_PRODI IN(09,10);
```

16). *Sequence Container* **LOAD3**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD3**.

17). *Execute SQL Task* **LOAD3**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD2** sebanyak 3 kali dan beri nama **UPDATE\_NOMINAL\_BEASISWA**, **UPDATE\_PRODI\_MAHASISWA**, dan **UPDATE\_MAHASISWA**. Pilih koneksi tersebut (**DATA-WAREHOUSE. SIKMA\_STAGING**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

- a. Ikon *Execute SQL Task* **UPDATE\_NOMINAL\_BEASISWA** berfungsi untuk mengubah data nominal beasiswa yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE PENERIMA_BEASISWA SET NOMINAL=BIAYA_SPP WHERE
JENIS_BEASISWA='SPP Tetap' and NPM IN(SELECT NPM FROM
PENERIMA_BEASISWA WHERE JENIS_ BEASISWA='SPP Tetap');UPDATE
PENERIMA_BEASISWA SET NOMINAL=(0.5*BIAYA_SPP) WHERE
JENIS_BEASISWA ='SPP 50% Tetap' and NPM IN(SELECT NPM FROM
PENERIMA_BEASISWA WHERE JENIS_BEASISWA='SPP 50% Tetap');
```

- b. Ikon *Execute SQL Task* **UPDATE\_PRODI\_MAHASISWA** berfungsi untuk mengubah data prodi mahasiswa yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE TBL_MAHASISWA SET ID_PRODI = 25 WHERE ID_PRODI IS NULL;
UPDATE TBL_MAHASISWA SET ID_FAKULTAS=7 WHERE ID_FAKULTAS IS
NULL;
```

- c. Ikon *Execute SQL Task* **UPDATE\_MAHASISWA** berfungsi untuk mengubah data mahasiswa yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE MAHASISWA_TABEL SET NAMA_MHS='N/A' WHERE (NPM IS NULL)
OR (NAMA_MHS IS NULL); UPDATE MAHASISWA_TABEL SET TMP_
LAHIR='N/A' WHERE (TMP_LAHIR IS NULL); UPDATE MAHASISWA_TABEL SET
JNS_KEL='-1' WHERE (JNS_KEL IS NULL); UPDATE MAHASISWA_TABEL SET
ID_KAB_MHS=1 WHERE (ID_KAB_MHS IS NULL);
```

18). *Sequence Container* **LOAD4**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD4**.



19). *Execute SQL Task* **LOAD4**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD4** dan beri nama dengan **UPDATE\_SMA\_MAHASISWA**. Ikon *Execute SQL Task* ini berfungsi untuk mengubah data sma yang ada pada database **SIKMA\_STAGING**. Pilih koneksi pada *Execute SQL Task* tersebut (**DATA-WAREHOUSE.SIKMA\_STAGING**). Lakukan konfigurasi dengan menuliskan sintaks berikut:

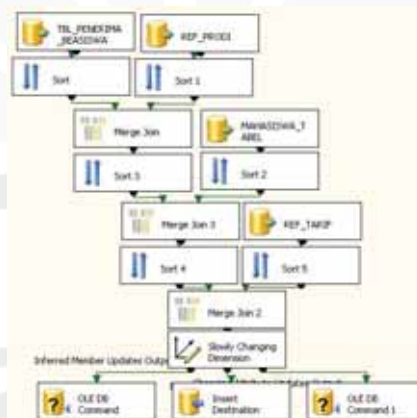
```
UPDATE MAHASISWA_TABEL SET ID_SMA=1 WHERE ID_SMA IS NULL;UPDATE
MAHASISWA_TABEL SET ID_SMA=1 WHERE ID_SMA=1;
```

20). *Sequence Container* **LOAD5**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD5**.

21). *Data Flow Task* pada **LOAD5**

Tarik dan letakkan ikon *data flow task* ke dalam item **LOAD5** dan beri nama dengan **PENERIMA\_BEASISWA**.



**Gambar 5.148.** *Data Flow Task* PENERIMA\_BEASISWA – *Incremental Load*

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **TBL\_PENERIMA\_BEASISWA**

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[TBL\_PENERIMA\_BEASISWA]

b) **REF\_PRODI**

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[REF\_PRODI]



c) **MAHASISWA\_TABEL**

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[MAHASISWA\_TABEL]

d) **REF\_TARIF**

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[REF\_TARIF]

b. Pada *data flow task* **TBL\_PENERIMA\_BEASISWA** dan **REF\_PRODI** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **PRODI** yang telah diurutkan untuk mengambil data **ID\_PRODI**.

c. Pada *data flow task* **MAHASISWA\_TABEL** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **THN\_MASUK**.

d. Pada *data flow task* **REF\_TARIF** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID\_PRODI** dan **THN\_MASUK** yang telah diurutkan untuk mengambil data **biaya**.

e. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan lakukan konfigurasi berikut:

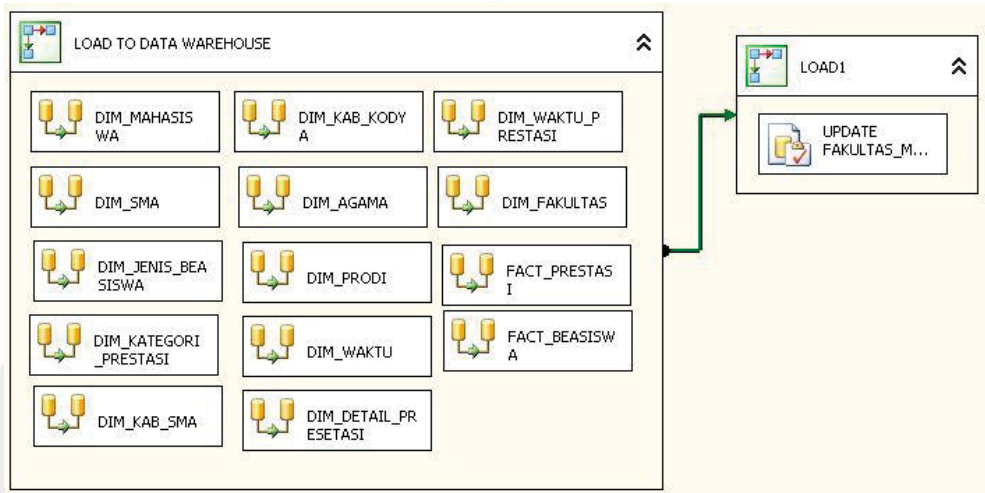
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[PENERIMA\_BEASISWA]

Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **JENIS\_BEASISWA, NAMA\_MHS, NOMINAL, PRODI, SEMESTER, dan TAHUN**.

#### 5.1.4.2. Loading Data dari SIKMA\_STAGING ke SIKMA\_DW

Dalam tahap ini, data pada *database* **SIKMA\_STAGING** akan di-load ke dalam *data warehouse* yaitu **SIKMA\_DW**. Pada proses *loading* data dari *database* **SIKMA\_STAGING** ke dalam *database* **SIKMA\_DW**, langkah

pertama adalah menentukan *control flow items* yang sesuai dengan kebutuhan pada *tab Control Flow*. Proses *loading* data dari **SIKMA\_STAGING** ke **SIKMA\_DW** pada *package* Warehouse\_sikma.dtsx akan dibuat desain proses sebagai berikut :



**Gambar 5.149. Desain Control Flow pada Warehouse\_sikma – Incremental Load**

Untuk membuat desain *Control Flow* di atas, langkah-langkah yang dilakukan adalah sebagai berikut:

1). *Sequence Container* **LOAD TO DATA WAREHOUSE**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD TO DATA WAREHOUSE**.

2). *Data Flow Task* **LOAD TO DATA WAREHOUSE**

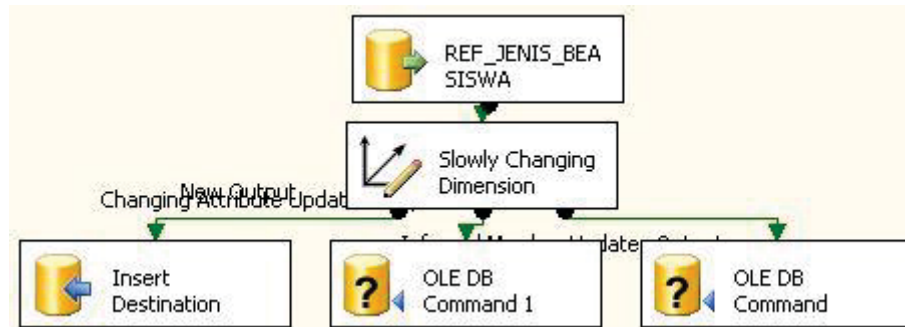
Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* tersebut. Karena ada 14 data sumber yang akan di-load maka dibuat 14 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **DIM\_KATEGORI\_PRESTASI, DIM\_WAKTU\_PRESTASI, DIM\_MAHASISWA, DIM\_DETAIL\_PRESTASI, DIM\_SMA, DIM\_PRODI, DIM\_FAKULTAS, DIM\_JENIS\_BEASISWA, DIM\_AGAMA, DIM\_KAB\_SMA, FACT\_PRESTASI, DIM\_WAKTU, FACT\_BEASISWA, dan DIM\_KAB\_KODYA.**

Hubungkan **PREPARE DATABASE** *sequence container* dengan *sequence container* **LOAD TO DATA WAREHOUSE**.

3). *Data Flow Task* pada *Sequence Container* **LOAD TO DATA WAREHOUSE**

Langkah-langkah untuk membuat *data flow task* **LOAD TO DATA WAREHOUSE** sebagai berikut:

## 1. DIM\_JENIS\_BEASISWA



Gambar 5.150. Data Flow Task DIM\_JENIS\_BEASISWA – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_JENIS\_BEASISWA]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

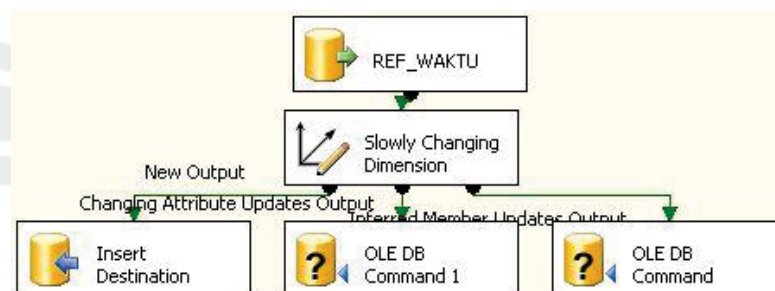
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[DIM\_JENIS\_BEASISWA]

Pilih **ID\_JENIS\_BEASISWA** sebagai business key. Pilih tipe changing attribute untuk kolom **JENIS\_BEASISWA**.

## 2. DIM\_WAKTU



Gambar 5.151. Data Flow Task DIM\_WAKTU – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_WAKTU]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

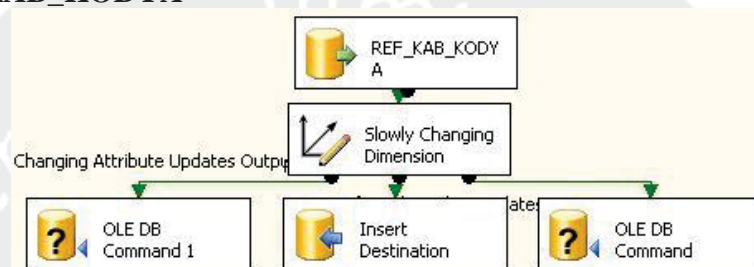
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[DIM\_WAKTU]

Pilih **ID\_WAKTU** sebagai business key. Pilih tipe changing attribute untuk kolom **NO\_SEMESTER**, **SEMESTER**, dan **TAHUN\_AKADEMIK**.

### 3. DIM\_KAB\_KODYA



Gambar 5.152. Data Flow Task DIM\_KAB\_KODYA – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_KAB\_KODYA]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

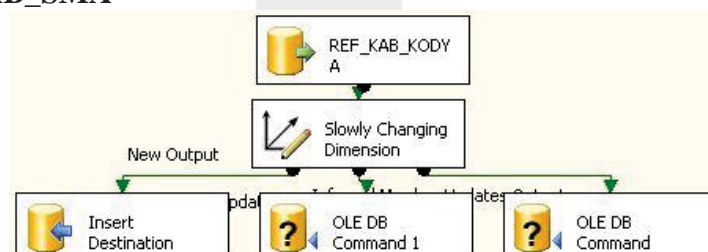
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[DIM\_KAB\_KODYA]

Pilih **ID\_KAB\_KODYA** sebagai business key. Pilih tipe changing attribute untuk kolom **NAMA\_KAB\_KODYA** dan **NAMA\_PROPINSI**.

### 4. DIM\_KAB\_SMA



Gambar 5.152. Data Flow Task DIM\_KAB\_KODYA – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_KAB\_KODYA]

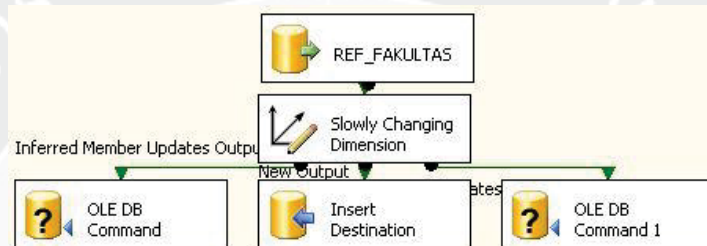
- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_KAB\_SMA]

Pilih **ID\_KAB\_KODYA** sebagai business key. Pilih tipe changing attribute untuk kolom **NAMA\_KAB\_KODYA** dan **NAMA\_PROPINSI**.

## 5. DIM\_FAKULTAS



Gambar 5.154. Data Flow Task DIM\_FAKULTAS – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_FAKULTAS]

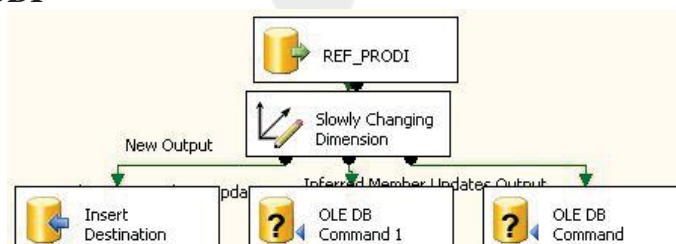
- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_FAKULTAS]

Pilih **ID\_FAKULTAS** sebagai business key. Pilih tipe changing attribute untuk kolom **ALAMAT\_FAKULTAS**, **FAKULTAS**, **FAKULTAS\_ING**, **FAX\_FAKULTAS**, dan **TELP\_FAKULTAS**.

## 6. DIM\_PRODI



Gambar 5.155. Data Flow Task DIM\_PRODI – Incremental Load



- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

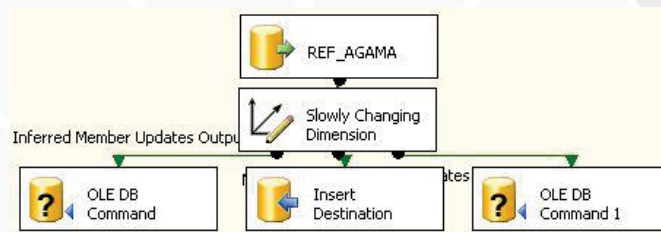
**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_PRODI]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_PRODI]

Pilih **ID\_FAKULTAS** dan **ID\_PRODI** sebagai business key. Pilih tipe changing attribute untuk kolom **BHS**, **JENJANG**, **KDPRODIDIKTI**, **PRODI**, dan **PRODI\_ING**.

## 7. DIM\_AGAMA



Gambar 5.156. Data Flow Task DIM\_AGAMA – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

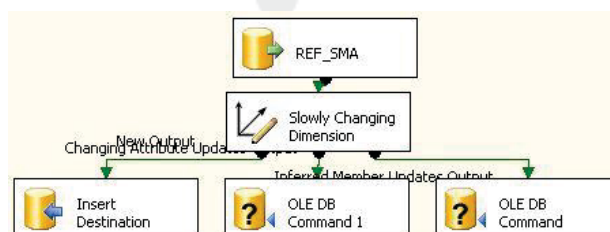
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_AGAMA]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_AGAMA]

Pilih **ID\_AGAMA** sebagai business key. Pilih tipe changing attribute untuk kolom **AGAMA** dan **AGAMA\_MHS**.

## 8. DIM\_SMA



Gambar 5.157. Data Flow Task DIM\_SMA – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of the table or view** : [dbo].[REF\_SMA]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

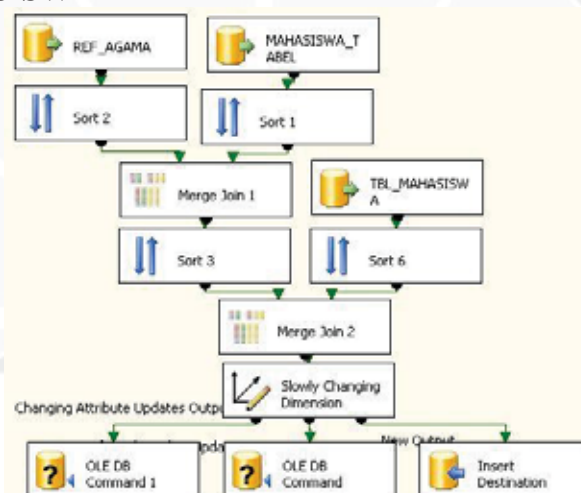
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW

**Use a Table or View** : [dbo].[DIM\_SMA]

Pilih **ID\_SMA** sebagai business key. Pilih tipe changing attribute untuk kolom **ALAMAT**, **ID\_KAB\_KODYA\_SMA** dan **NAMA\_SMA**.

## 9. DIM\_MAHASISWA



Gambar 5.157. Data Flow Task DIM\_MAHASISWA – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

- a) MAHASISWA\_TABEL

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[MAHASISWA\_TABEL]

- b) REF\_AGAMA

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[REF\_AGAMA]

- c) TBL\_MAHASISWA

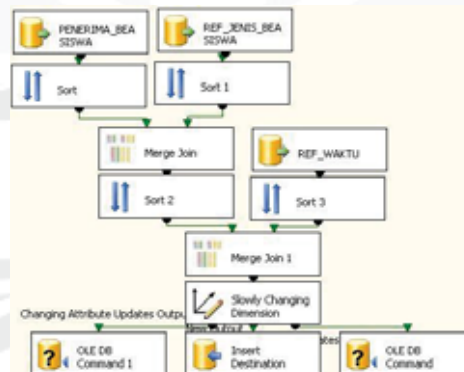
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[TBL\_MAHASISWA]

- b. Pada *data flow task* **MAHASISWA\_TABEL** dan **REF\_AGAMA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **AGAMA** yang telah diurutkan untuk mengambil data **ID\_AGAMA**.
- c. Pada *data flow task* **TBL\_MAHASISWA** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **ID\_FAKULTAS** dan **ID\_PRODI**.
- d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan lakukan konfigurasi berikut:
- Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_MAHASISWA]  
 Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **JNS\_KEL**, **NAMA\_MHS**, **TGL\_LAHIR**, **THN\_MASUK**, dan **TMP\_LAHIR**.

## 10. FACT\_BEASISWA



Gambar 5.159. Data Flow Task FACT\_BEASISWA – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:
- a) PENERIMA\_BEASISWA
- Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[PENERIMA\_BEASISWA]
- b) JENIS\_BEASISWA
- Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[REF\_JENIS\_BEASISWA]

## c) WAKTU

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[REF\_WAKTU]

b. Pada *data flow task* **PENERIMA\_BEASISWA** dan **JENIS\_BEASISWA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **JENIS\_BEASISWA** yang telah diurutkan untuk mengambil data **ID\_JENIS\_BEASISWA**.

c. Pada *data flow task* **WAKTU** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **TAHUN** dan **SEMESTER** yang telah diurutkan untuk mengambil data **ID\_WAKTU**.

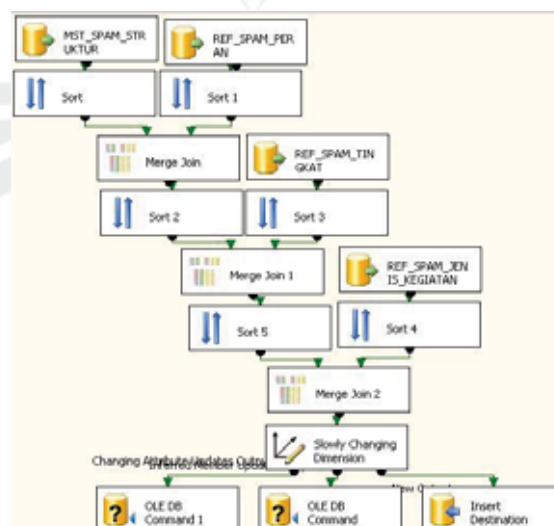
d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[FACT\_BEASISWA]

Pilih **ID\_JENIS\_BEASISWA**, **ID\_WAKTU**, **NPM** sebagai business key.

Pilih tipe changing attribute untuk kolom **NOMINAL**.

## 11. DIM\_KATEGORI\_PPRESTASI



Gambar 5.160. *Data Flow Task* DIM\_KATEGORI\_PPRESTASI– Incremental Load

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

## a) MST\_SPAM\_STRUKTUR

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo],[MST\_SPAM\_STRUKTUR]

## b) REF\_PERAN

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo],[REF\_SPAM\_PERAN]

## c) REF\_TINGKAT

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo],[REF\_SPAM\_TINGKAT]

## d) REF\_JENIS\_KEGIATAN

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo],[REF\_SPAM\_JENIS\_KEGIATAN]

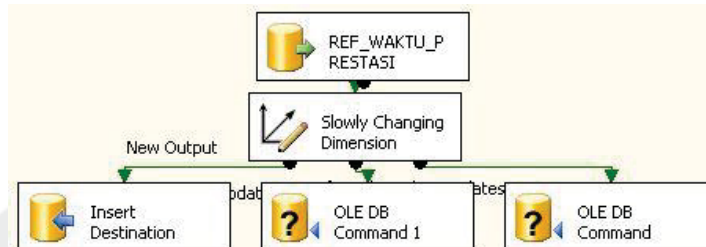
- b. Pada *data flow task* **MST\_SPAM\_STRUKTUR** dan **REF\_PERAN** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE\_PERAN** yang telah diurutkan untuk mengambil data **KETERANGAN\_PERAN**.
- c. Pada *data flow task* **REF\_TINGKAT** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE\_TINGKAT** yang telah diurutkan untuk mengambil data **KETERANGAN\_TINGKAT**.
- d. Pada *data flow task* **REF\_JENIS\_KEGIATAN** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE\_KEGIATAN**, **KODE\_JENIS** dan **KODE\_KATEGORI** yang telah diurutkan untuk mengambil data **KETERANGAN\_KATEGORI**, **KETERANGAN\_JENIS**, dan **KETERANGAN\_KEGIATAN**.
- e. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo],[DIM\_KATEGORI\_PRESTASI]



Pilih **NO\_URUT** sebagai business key. Pilih tipe changing attribute untuk kolom **JENIS, KATEGORI, KEGIATAN, PERAN, dan TINGKAT**.

## 12. DIM\_WAKTU\_PRESTASI



Gambar 5.161. Data Flow Task DIM\_WAKTU\_PRESTASI – Incremental Load

- Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

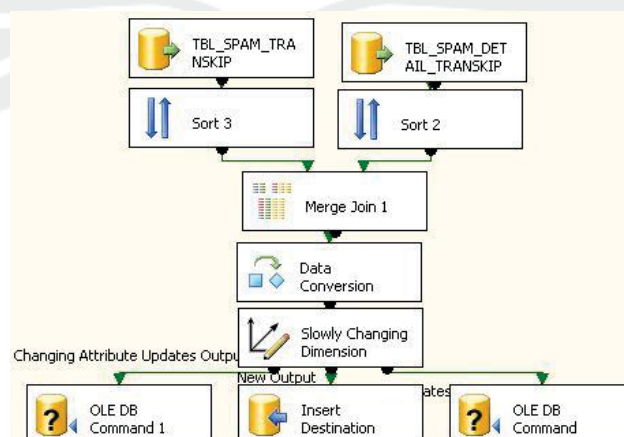
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_WAKTU\_PRESTASI]

- Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_WAKTU\_PRESTASI]

Pilih **ID\_WAKTU\_PRESTASI** sebagai business key. Pilih tipe changing attribute untuk kolom **SEMESTER\_AKADEMIK, TAHUN, dan TANGGAL**.

## 13. DIM\_DETAIL\_PRESTASI



Gambar 5.162. Data Flow Task DIM\_DETAIL\_PRESTASI – Incremental Load

- Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **DETAIL\_TRANSKIP**

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[TBL\_SPAM\_DETAIL\_TRANSKIP]

b) **TRANSKIP**

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[TBL\_SPAM\_TRANSKIP]

b. Pada *data flow task* **DETAIL\_TRANSKIP** dan **TRANSKIP** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID\_TRANSKIP\_SPAM** yang telah diurutkan untuk mengambil data **NAMA\_KEGIATAN**.

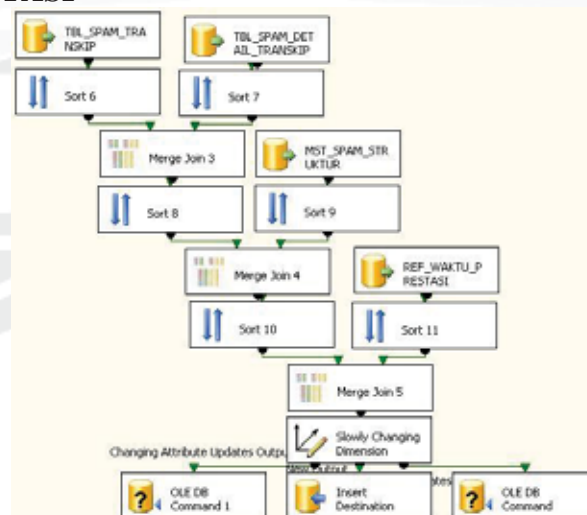
c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA\_DW

Use a Table or View : [dbo].[DIM\_DETAIL\_PRESTASI]

Pilih **ID\_DETAIL\_PRESTASI** sebagai business key. Pilih tipe changing attribute untuk kolom **PENYELENGGARA** dan **TEMPAT**.

#### 14. FACT\_PRESTASI



Gambar 5.163. *Data Flow Task* **FACT\_PRESTASI – Incremental Load**

a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **TRANSKIP\_PRESTASI**

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[TBL\_SPAM\_TRANSKIP]

**b) DETAIL\_TRANSKIP**

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[TBL\_SPAM\_DETAIL\_TRANSKIP]

**c) SPAM\_STRUKTUR**

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[MST\_SPAM\_STRUKTUR]

**d) WAKTU\_PRESTASI**

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : [dbo].[REF\_WAKTU\_PRESTASI]

b. Pada *data flow task* **TRANSKIP\_PRESTASI** dan **DETAIL\_TRANSKIP** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID\_TRANSKIP\_SPAM** yang telah diurutkan untuk mengambil data **TOTAL\_POIN**.

c. Pada *data flow task* **SPAM\_STRUKTUR** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **KODE\_KATEGORI**, **KODE\_KEGIATAN**, **KODE\_JENIS**, **KODE\_TINGKAT**, **KODE\_PERAN**, dan **POIN** yang telah diurutkan untuk mengambil data **NO\_URUT**.

d. Pada *data flow task* **WAKTU\_PRESTASI** dan hasil merge sebelumnya digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **TGL\_MULAI**, **ID\_TAHUN\_AKADEMIK**, dan **NO\_SEMESTER** yang telah diurutkan untuk mengambil data **ID\_WAKTU**.

e. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA\_DW  
 Use a Table or View : [dbo].[FACT\_PRESTASI]

Pilih **ID\_DETAIL\_PRESTASI**, **ID\_KATEGORI\_PRESTASI**, **ID\_WAKTU\_PRESTASI**, dan **NPM** sebagai business key. Pilih tipe *changing attribute* untuk kolom **POIN**.

#### 4). *Sequence Container* **LOAD1**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD1**.

#### 5). *Execute SQL Task* **UPDATE\_FAKULTAS\_MAHASISWA**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD1** dan beri nama **UPDATE\_FAKULTAS\_MAHASISWA**. Ikon *Execute SQL Task* ini berfungsi menjalankan *query* untuk mengubah data fakultas yang ada pada *database* **SIKMA\_DW**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW**). Lakukan konfigurasi dengan menuliskan sintaks berikut:

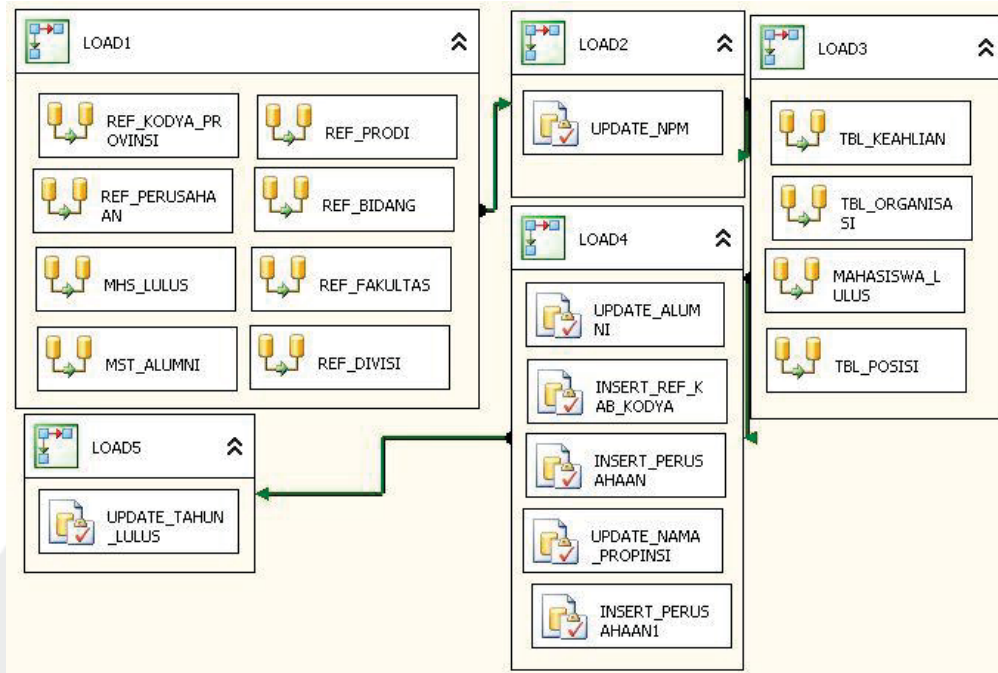
```
UPDATE DIM_MAHASISWA SET ID_FAKULTAS=5 WHERE ID_PRODI IN(01,02, 13);
UPDATE DIM_MAHASISWA SET ID_FAKULTAS=1 WHERE ID_PRODI IN(
03,04,11,12,15,20); UPDATE DIM_MAHASISWA SET ID_FAKULTAS=2 WHERE
ID_PRODI IN(05,21); UPDATE DIM_MAHASISWA SET ID_ FAKULTAS=3 WHERE
ID_PRODI =08; UPDATE DIM_MAHASISWA SET ID_ FAKULTAS=4 WHERE
ID_PRODI IN(06,07,14,24); UPDATE DIM_MAHASISWA SET ID_FAKULTAS=6
WHERE ID_PRODI IN(09,10); UPDATE DIM_MAHASISWA SET ID_AGAMA=6
WHERE ID_AGAMA IS NULL; UPDATE DIM_MAHASISWA SET ID_PRODI = 25
WHERE ID_PRODI IS NULL; UPDATE DIM_MAHASISWA SET ID_FAKULTAS=7
WHERE ID_FAKULTAS IS NULL;
```

#### 5.1.5. *Control Flow Incremental Load* **ALUMNI**

Proses *incremental load* **ALUMNI** terbagi menjadi dua proses. Proses pertama adalah mengisi data dari *database* sumber menuju *staging area*. Proses selanjutnya adalah mengisi data dari *staging area* menuju *data warehouse*.

##### 5.1.5.1. *Loading Data dari Sumber Data ke* **ALUMNI\_STAGING**

Pada proses *loading* data dari *database* sumber ke dalam *database* **SIKMA\_STAGING**, langkah pertama adalah menentukan *control flow items* yang sesuai dengan kebutuhan pada *tab control flow*. Untuk menggunakan *control flow items* yang telah disediakan, tarik dan letakkan ikon *control flow items* yang diperlukan ke area tengah. Kemudian proses *loading* data dari sumber data ke **SIKMA\_STAGING** pada *package* **ALUMNI\_STAGING.dtsx** akan terbentuk desain proses sebagai berikut :



**Gambar 5.164. Desain Control Flow pada ALUMNI\_STAGING – Incremental Load**

Untuk membuat desain *Control Flow* di atas, langkah-langkah yang dilakukan adalah sebagai berikut:

1). *Sequence Container* **LOAD1**

Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD1**.

2). *Data Flow Task* **LOAD1**

*Control flow item* yang digunakan untuk ekstrak, *transform*, dan *load* data adalah *data flow task*. Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* **LOAD1**. Karena ada 8 data sumber yang akan di-load maka dibuat 8 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **REF\_PERUSAHAAN**, **REF\_BIDANG**, **REF\_DIVISI**, **MST\_ALUMNI**, **REF\_PRODI**, **REF\_FAKULTAS**, **MHS\_LULUS**, dan **REF\_KODYA\_PROVINSI**.

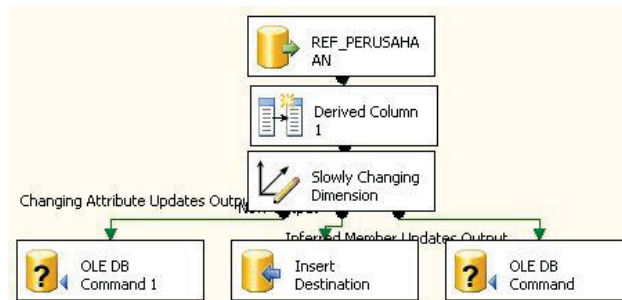
Hubungkan **PREPARE DATABASE** *sequence container* dengan *sequence container* **LOAD1**.

3). *Data Flow Task* pada *Sequence Container* **LOAD1**

Langkah-langkah untuk membuat *data flow task* pada **LOAD1** sebagai berikut:



## 1. REF\_PERUSAHAAN



Gambar 5.165. Data Flow Task REF\_PERUSAHAAN – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax  
**Data Access Mode** : Table or view  
**Name of table or view** : dbo].[REF\_PERUSAHAAN]

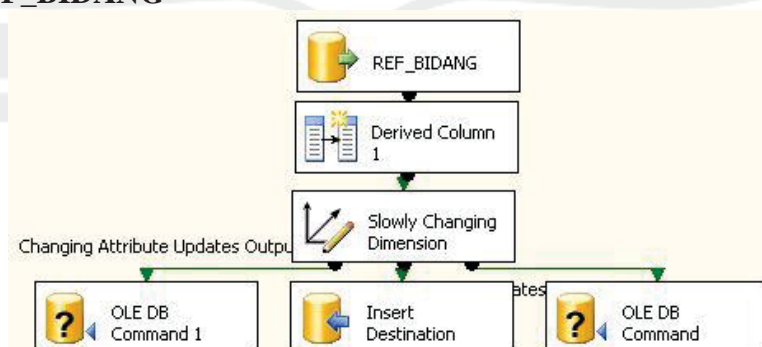
- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.

- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[REF\_PERUSAHAAN]

Pilih **ID\_PERUSAHAAN** sebagai business key. Pilih tipe changing attribute untuk kolom **NAMA**.

## 2. REF\_BIDANG



Gambar 5.166. Data Flow Task REF\_BIDANG – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax  
**Data Access Mode** : Table or view  
**Name of table or view** : dbo].[REF\_BIDANG]

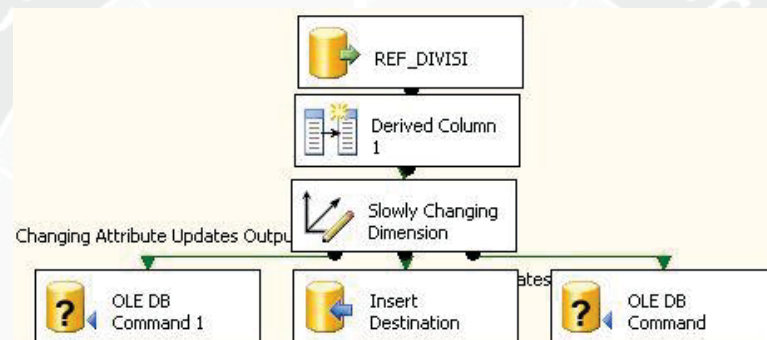
- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_BIDANG]

Pilih **ID\_BIDANG** sebagai business key. Pilih tipe changing attribute untuk kolom **BIDANG**.

### 3. REF\_DIVISI



**Gambar 5.167.** Data Flow Task REF\_DIVISI – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[REF\_DIVISI]

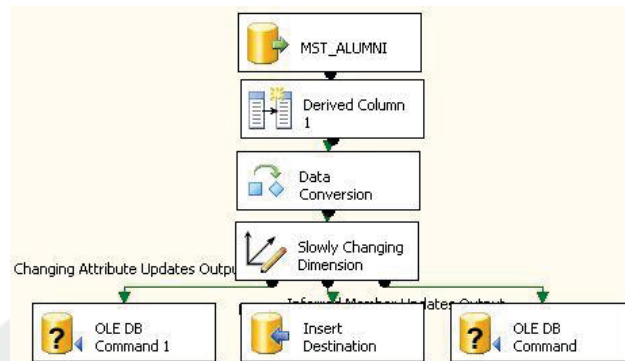
- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_DIVISI]

Pilih **ID\_DIVISI** sebagai business key. Pilih tipe changing attribute untuk kolom **DIVISI**.

#### 4. MST\_ALUMNI



Gambar 5.168. Data Flow Task MST\_ALUMNI – Incremental Load

- a. Data yang akan di-load bersumber dari server, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

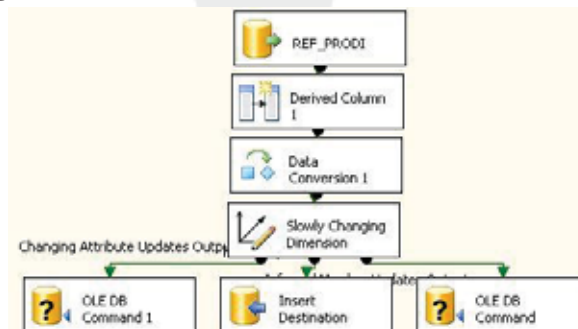
**Connection Manager** : DATA-WAREHOUSE.siatmax  
**Data Access Mode** : Table or view  
**Name of table or view** : dbo].[MST\_ALUMNI]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam database sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Fungsi *data conversion* digunakan untuk mengkonversi **TMP\_LAHIR** menjadi data *varchar*.
- d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[MST\_ALUMNI]

Pilih **ID\_ALUMNI** sebagai business key. Pilih tipe changing attribute untuk kolom **AGAMA**, **GAJI\_PERTAMA**, **ID\_FAKULTAS**, **ID\_PRODI**, **NAMA\_MHS**, dan **TAHUN\_LULUS**.

#### 5. REF\_PRODI



Gambar 5.169. Data Flow Task REF\_PRODI – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **REF\_PRODI**

Connection Manager : DATA-WAREHOUSE.biadw  
 Data Access Mode : Table or view  
 Name of table or view : dbo].[prodi]

b) **REF\_PRODI1**

Connection Manager : DATA-WAREHOUSE.siatmax  
 Data Access Mode : Table or view  
 Name of table or view : dbo].[REF\_PRODI]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Fungsi *data conversion* digunakan untuk mengkonversi **ID\_PRODI** menjadi data *varchar*.
- d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:

Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Use a Table or View : [dbo].[REF\_PRODI]

Pilih **ID\_PRODI** sebagai business key. Pilih tipe changing attribute untuk kolom **BHS**, **ID\_FAKULTAS**, **JENJANG**, **KDPRODIDIKTI**, **PRODI**, dan **PRODI\_ING**.

## 6. REF\_FAKULTAS



Gambar 5.170. Data Flow Task REF\_FAKULTAS – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) **REF\_FAKULTAS**

Connection Manager : DATA-WAREHOUSE.siatmax  
 Data Access Mode : Table or view  
 Name of table or view : dbo].[REF\_FAKULTAS]

### b) FAKULTAS

**Connection Manager** : DATA-WAREHOUSE.biadw  
**Data Access Mode** : Table or view  
**Name of table or view** : dbo].[fakultas]

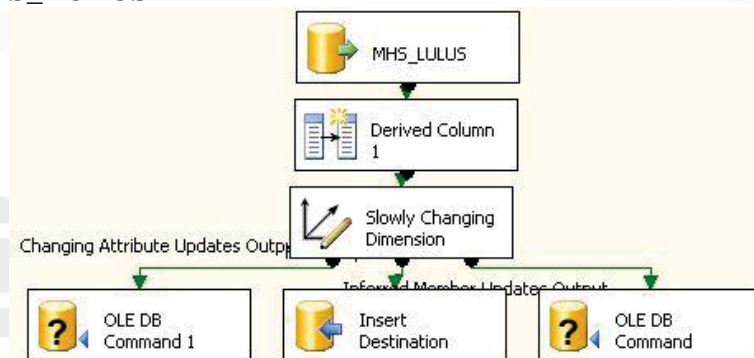
- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Fungsi *data conversion* digunakan untuk mengkonversi **ID\_FAKULTAS** menjadi data *int* dan **KD\_FAKULTAS** menjadi data *int*.
- d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *data conversion*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[REF\_FAKULTAS]

Pilih **ID\_FAKULTAS** sebagai business key. Pilih tipe changing attribute untuk kolom **ALAMAT\_FAKULTAS**, **FAKULTAS**, **FAKULTAS\_ING**, **FAX\_FAKULTAS**, dan **TELP\_FAKULTAS**.

### 7. MHS\_LULUS



Gambar 5.171. Data Flow Task MHS\_LULUS – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.biadw  
**Data Access Mode** : Table or view  
**Name of table or view** : dbo).[mhs\_lulus]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.



- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[TBL\_MHS\_LULUS]

Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **AGAMA**, **AWAL\_STUDI**, **IPK\_LULUS**, **JNS\_KEL**, **LAMA\_STUDI**, dan **LAMA\_TA**.

## 8. REF\_KODYA\_PROVINSI



Gambar 5.172. Data Flow Task REF\_KODYA\_PROVINSI – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.biadw

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[prop\_kab\_kodya]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[REF\_KAB\_KODYA]

Pilih **ID\_KAB\_KODYA** sebagai business key. Pilih tipe changing attribute untuk kolom **NAMA\_KAB\_KODYA** dan **NAMA\_PROPINSI**.

## 4). Sequence Container LOA2

Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD2**.

8). *Execute SQL Task* **LOAD2**

Tarik dan letakkan ikon *execute SQL task* ke dalam item **LOAD2** dan beri nama **UPDATE\_NPM**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_STAGING**). Ikon *Execute SQL Task* **UPDATE\_NPM** berfungsi menjalankan *query* untuk mengubah data npm mahasiswa pada *database* **SIKMA\_STAGING**. Lakukan konfigurasi dengan menuliskan sintaks berikut: `UPDATE MST_ALUMNI SET NPM=SUBSTRING(NPM,1,9) WHERE NPM IN(SELECT NPM FROM MST_ALUMNI);`

9). *Sequence Container* **LOAD3**

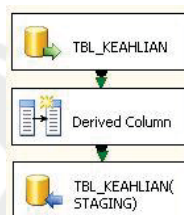
Tarik dan letakkan ikon *sequence container* dan beri nama **LOAD3**.

5). *Data Flow Task* **LOAD3**

Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* **LOAD3**. Karena ada 4 data sumber yang akan di-load maka dibuat 4 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **TBL\_KEAHLIAN**, **TBL\_POSISI**, **MAHASISWA\_LULUS**, dan **TBL\_ORGANISASI**. Hubungkan **LOAD3** *sequence container* dengan *sequence container* **LOAD2**.

6). *Data Flow Task* pada *Sequence Container* **LOAD3**

Langkah-langkah untuk membuat *data flow task* pada **LOAD3** sebagai berikut:

1. **TBL\_KEAHLIAN**

**Gambar 5.173.** *Data Flow Task* **TBL\_KEAHLIAN** – *Incremental Load*

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : **DATA-WAREHOUSE.siatmax**  
**Data Access Mode** : **Table or view**  
**Name of table or view** : **dbo].[TBL\_KEAHLIAN]**

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.

- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

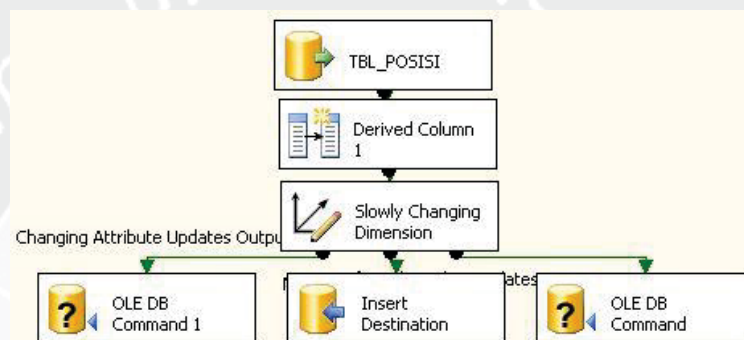
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[TBL\_KEAHLIAN]

Pilih **ID\_ALUMNI** dan **ID\_KEAHLIAN** sebagai business key. Pilih tipe changing attribute untuk kolom **LEVEL\_KEAHLIAN**, **NAMA**, dan **TAHUN**.

## 2. TBL\_POSISI



**Gambar 5.174. Data Flow Task TBL\_POSISI – Incremental Load**

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax

**Data Access Mode** : Table or view

**Name of table or view** : [dbo].[TBL\_POSISI]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.

- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

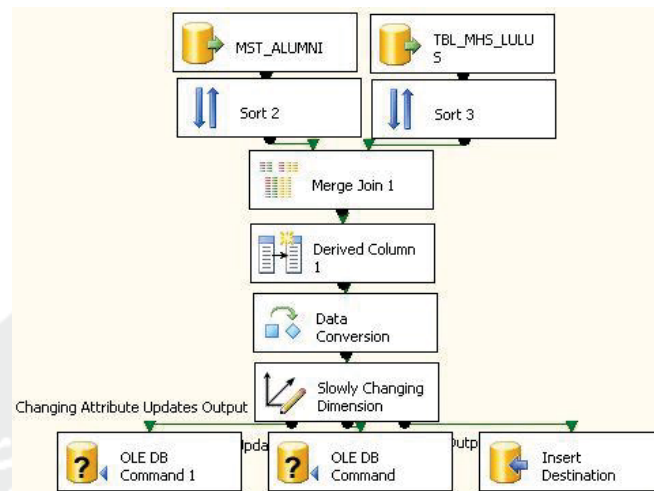
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING

**Use a Table or View** : [dbo].[TBL\_POSISI]

Pilih **ID\_ALUMNI** dan **ID\_POSISI** sebagai business key. Pilih tipe changing attribute untuk kolom **BIDANG**, **DIVISI**, **JABATAN**, **PERUSAHAAN**, **THN\_MASUK** dan **THN\_KELUAR**.

### 3. MAHASISWA\_LULUS



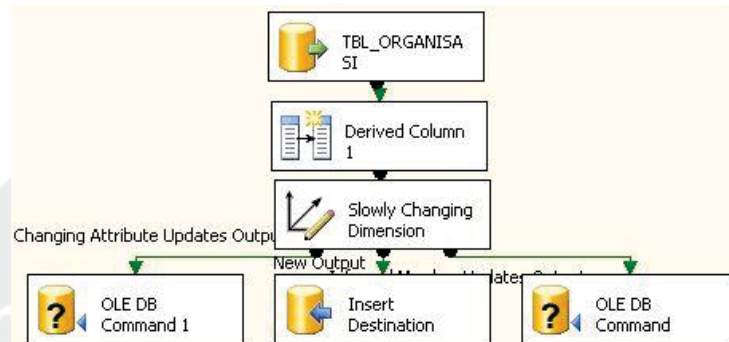
Gambar 5.175. *Data Flow Task MAHASISWA\_LULUS – Incremental Load*

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:
  - a) **MST\_ALUMNI**  
 Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : dbo].[MST\_ALUMNI]
  - b) **TBL\_MHS\_LULUS**  
 Connection Manager : DATA-WAREHOUSE.SIKMA\_STAGING  
 Data Access Mode : Table or view  
 Name of table or view : dbo].[TBL\_MHS\_LULUS]
- b. Pada *data flow task* **MST\_ALUMNI** dan **TBL\_MHS\_LULUS** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **NPM** yang telah diurutkan untuk mengambil data **AWAL\_STUDI**, **THN\_MASUK**, **JLR\_LULUS**, dan **KD\_PRODI**.
- c. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- d. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*. Lakukan konfigurasi berikut:
 

|                     |                                |
|---------------------|--------------------------------|
| Connection Manager  | : DATA-WAREHOUSE.SIKMA_STAGING |
| Use a Table or View | : [dbo].[TBL_ALUMNI]           |

Pilih **NPM** sebagai business key. Pilih tipe changing attribute untuk kolom **AGAMA**, **AWAL\_STUDI**, **IPK\_LULUS**, **JNS\_KEL**, **KD\_PRODI**, **LAMA\_STUDI**, **LAMA\_TA**, dan **NAMA\_MHS**.

#### 4. TBL\_ORGANISASI



Gambar 5.176. Data Flow Task TBL\_ORGANISASI – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.siatmax  
**Data Access Mode** : Table or view  
**Name of table or view** : dbo].[TBL\_ORGANISASI]

- b. Tambahkan kolom waktu pada saat data dimasukkan ke dalam *database* sebagai kelengkapan historis dengan *derived column*. Langkah-langkah pembuatan *derived column* dapat dilihat pada lampiran tugas akhir ini.
- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan hubungkan dengan *derived column*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Use a Table or View** : [dbo].[TBL\_ORGANISASI]

Pilih **ID\_ALUMNI** dan **ID\_ORGANISASI** sebagai business key. Pilih tipe changing attribute untuk kolom **JABATAN**, **NAMA\_ORGANISASI**, **THN\_MULAI** dan **THN\_SELESAI**.

#### 13). Sequence Container **LOAD4**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD4**.

#### 14). Execute SQL Task **LOAD4**

Tarik dan letakkan ikon Execute SQL Task ke dalam item **LOAD4** sebanyak 5 kali dan beri nama **UPDATE\_ALUMNI**, **INSERT\_REF\_KAB\_KODYA**, **INSERT\_PERUSAHAAN**, **INSERT\_PERUSAHAAN1**, dan **UPDATE\_**



**NAMA\_PROPINSI.** Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_STAGING**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

- a. Ikon *Execute SQL Task* **UPDATE\_ALUMNI** berfungsi untuk mengubah data alumni yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE TBL_ALUMNI SET ID_FAKULTAS=5 WHERE ID_PRODI IN(01,02,13);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=1 WHERE ID_PRODI IN(03,04,11,12,15,20);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=2 WHERE ID_PRODI IN(05,21);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=3 WHERE ID_PRODI =08;
UPDATE TBL_ALUMNI SET ID_FAKULTAS=4 WHERE ID_PRODI IN(06,07,14,24);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=6 WHERE ID_PRODI IN(09,10);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=7 WHERE ID_PRODI IN(50,51,52,53,54);
UPDATE TBL_ALUMNI SET ID_FAKULTAS=8 WHERE ID_PRODI=60;
UPDATE TBL_ALUMNI SET JNS_KEL= '-' WHERE JNS_KEL IS NULL;
UPDATE TBL_ALUMNI SET AGAMA='N/A' WHERE AGAMA IS NULL;
UPDATE TBL_ALUMNI SET TMP_LAHIR='N/A' WHERE TMP_LAHIR IS NULL;
```

- b. Ikon *Execute SQL Task* **INSERT\_REF\_KAB\_KODYA** berfungsi untuk menambahkan data kabupaten kodya yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT INTO REF_KAB_KODYA VALUES(533,NULL,NULL,NULL);
INSERT INTO REF_KAB_KODYA VALUES(475,NULL,NULL,NULL);
INSERT INTO REF_KAB_KODYA VALUES(130,NULL,NULL,NULL);
```

- c. Ikon *Execute SQL Task* **INSERT\_PERUSAHAAN** berfungsi untuk menambah data perusahaan yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
INSERT INTO TBL_POSISI(ID_POSISI,ID_ALUMNI,PERUSAHAAN)VALUES(88,10833,'Adaro');
INSERT INTO TBL_POSISI(ID_POSISI,ID_ALUMNI,PERUSAHAAN)VALUES(89,11374,'Adicipta Inovasi Tekno logi');
```

- d. Ikon *Execute SQL Task* **INSERT\_PERUSAHAAN1** berfungsi untuk menambah data perusahaan yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
TBL_POSISI(ID_POSISI,ID_ALUMNI,PERUSAHAAN)VALUES(421,32107,'PT.Halilintar Lintas Semesta');
INSERT INTO TBL_POSISI(ID_POSISI,ID_ALUMNI,PERUSAHAAN)VALUES(422,33354,'PT. Harita Prima Abadi
```

```
Mineral'); INSERT INTO TBL_POSISI(ID_POSISI,
ID_ALUMNI,PERUSAHAAN)VALUES(423,32734,'PT. HM Sampoerna, Tbk');
```

- e. Ikon *Execute SQL Task* **UPDATE\_NAMA\_PROPINSI** berfungsi untuk mengubah data propinsi yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE REF_KAB_KODYA SET NAMA_PROPINSI='Nanggroe Aceh Darusalam'
WHERE ID_KAB_KODYA IN(SELECT ID_KAB_KODYA WHERE
NAMA_PROPINSI='NANGROE ACEH DARUSALAM'); UPDATE
REF_KAB_KODYA SET NAMA_PROPINSI='Bangka Belitung' WHERE
ID_KAB_KODYA IN(SELECT ID_KAB_KODYA WHERE
NAMA_PROPINSI='KEPULAUAN BANGKA BELITUNG');
```

15). *Sequence Container* **LOAD5**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD5**.

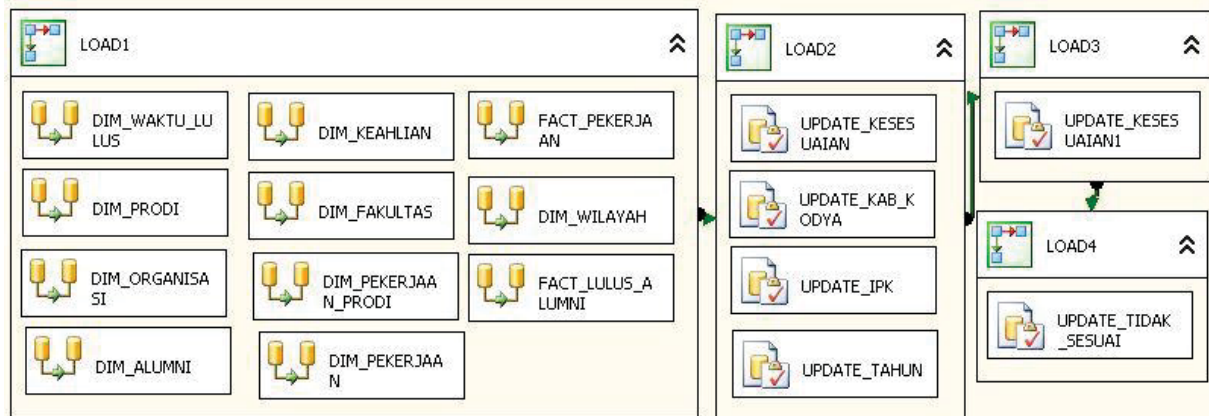
16). *Execute SQL Task* **LOAD5**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD5** dan beri nama **UPDATE\_TAHUN\_LULUS**. Pilih koneksi tersebut (**DATA-WAREHOUSE. SIKMA\_STAGING**). Ikon *Execute SQL Task* **UPDATE\_TAHUN\_LULUS** berfungsi untuk mengubah data tahun lulus yang ada pada *database SIKMA\_STAGING*. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE dbo.TBL_ALUMNI SET TAHUN_LULUS = SUBSTRING(CAST(DATE
PART([YEAR],TGL_WISUDA) AS VARCHAR(30)),1,4) WHERE(NPM IN
(SELECT NPM FROM dbo.TBL_ALUMNI AS TBL_ALUMNI_1))
```

### 5.1.5.2. Loading Data dari ALUMNI\_STAGING ke ALUMNI\_DW

Dalam tahap ini, data pada *database SIKMA\_STAGING* akan di-load ke dalam *data warehouse* yaitu **SIKMA\_DW\_ALUMNI**. Pada proses *loading* data dari *database SIKMA\_STAGING* ke dalam *database SIKMA\_DW\_ALUMNI*, langkah pertama adalah menentukan *control flow items* yang sesuai dengan kebutuhan pada *tab control flow*. Proses *loading* data dari **SIKMA\_STAGING** ke **SIKMA\_DW\_ALUMNI** pada *package ALUMNI\_DW.dtsx* akan dibuat desain proses sebagai berikut :



**Gambar 5.177. Desain Control Flow pada ALUMNI\_DW – Incremental Load**

Untuk membuat desain *Control Flow* di atas, langkah-langkah yang dilakukan adalah sebagai berikut:

1). *Sequence Container* **PREPARE DATABASE**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **PREPARE DATABASE**.

2). *Execute SQL Task* **FULL\_REFRESH\_DATABASE**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **PREPARE DATABASE** *sequence container* yang telah dibuat dan beri nama **FULL\_REFRESH\_DATABASE**. Ikon *Execute SQL Task* ini berfungsi menjalankan *query* untuk menghapus semua isi tabel yang ada pada *database* **SIKMA\_DW\_ALUMNI**. Hal ini dilakukan agar isi tabel pada *database* **SIKMA\_DW\_ALUMNI** selalu baru. Buat koneksi ke *database* pada *server* **DATA-WAREHOUSE**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI**). Langkah-langkah pembuatan koneksi **DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI** dapat diamati pada lampiran tugas akhir ini. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
DELETE FROM FACT_LULUS_ALUMNI; DELETE FROM FACT_PEMERJAAN;
DELETE FROM DIM_WAKTU_LULUS; DELETE FROM DIM_PRODI_ALUMNI;
DELETE FROM DIM_FAKULTAS_ALUMNI; DELETE FROM DIM_WILAYAH_ALUMNI;
DELETE FROM DIM_KEAHLIAN_ALUMNI; DELETE FROM DIM_ALUMNI;
DELETE FROM DIM_PEMERJAAN_PRODI; DELETE FROM DIM_ORGANISASI_ALUMNI;
DELETE FROM DIM_PEMERJAAN_ALUMNI;
```

3). *Sequence Container* **LOAD1**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD1**.

#### 4). *Data Flow Task* **LOAD1**

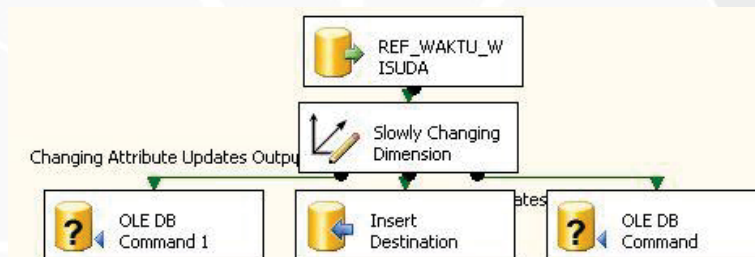
Tarik dan letakkan ikon *data flow task* ke dalam *sequence container* tersebut. Karena ada 11 data sumber yang akan di-load maka dibuat 11 buah *data flow task*. Beri nama masing-masing item *data flow task* tersebut **DIM\_WAKTU\_LULUS**, **DIM\_FAKULTAS**, **DIM\_PRODI**, **DIM\_WILAYAH**, **DIM\_PEKERJAAN\_PRODI**, **DIM\_ALUMNI**, **DIM\_KEAHLIAN**, **FACT\_LULUS\_ALUMNI**, **DIM\_PEKERJAAN**, **DIM\_ORGANISASI**, dan **FACT\_PEKERJAAN**.

Hubungkan **PREPARE DATABASE** *sequence container* dengan *sequence container* **LOAD1**.

#### 5). *Data Flow Task* pada *Sequence Container* **LOAD1**

Langkah-langkah untuk membuat *data flow task* **LOAD1** sebagai berikut:

##### 1. **DIM\_WAKTU\_LULUS**



Gambar 5.178. *Data Flow Task* **DIM\_WAKTU\_LULUS** – *Incremental Load*

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_WAKTU\_WISUDA]

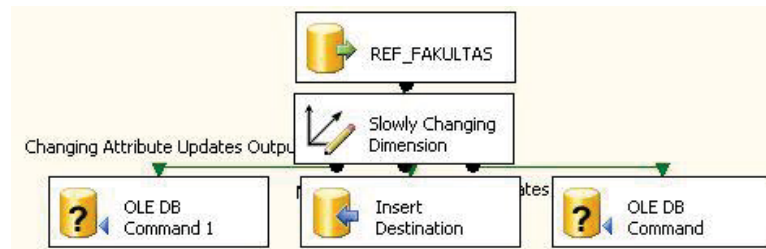
- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_WAKTU\_LULUS]

Pilih **ID\_WAKTU\_LULUS** sebagai business key. Pilih tipe changing attribute untuk kolom **TAHUN** dan **TGL\_LULUS**.

##### 2. **DIM\_FAKULTAS**



Gambar 5.179. Data Flow Task DIM\_FAKULTAS – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_FAKULTAS]

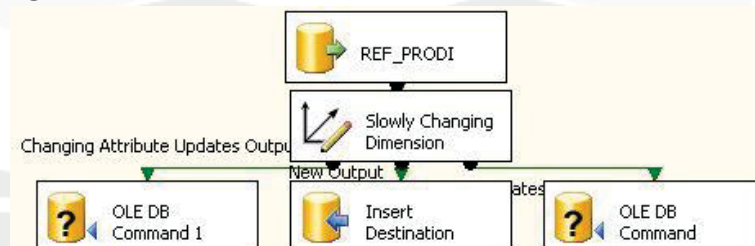
- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_FAKULTAS\_ALUMNI]

Pilih **ID\_FAKULTAS** sebagai business key. Pilih tipe changing attribute untuk kolom **ALAMAT\_FAKULTAS**, **FAKULTAS**, **FAKULTAS\_ING**, **FAX\_FAKULTAS** dan **TELP\_FAKULTAS**.

### 3. DIM\_PRODI



Gambar 5.180. Data Flow Task DIM\_PRODI – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_PRODI]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

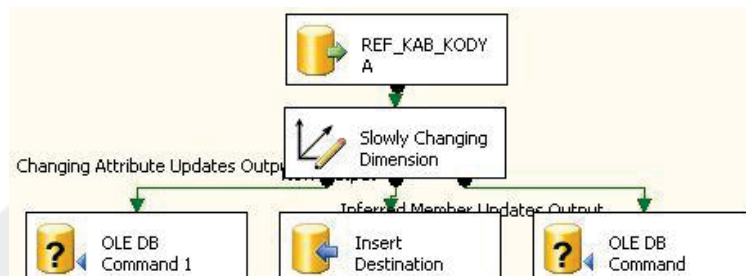
Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_PRODI\_ALUMNI]



Pilih **ID\_PRODI** dan **ID\_FAKULTAS** sebagai business key. Pilih tipe changing attribute untuk kolom **BHS**, **JENJANG**, **KDPRODIDIKTI**, **PRODI** dan **PRODI\_ING**.

#### 4. DIM\_WILAYAH



Gambar 5.181. Data Flow Task DIM\_WILAYAH – Incremental Load

- Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

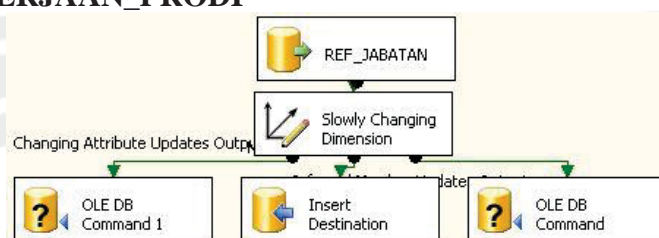
**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_KAB\_KODYA]

- Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_WILAYAH\_ALUMNI]

Pilih **ID\_KAB\_KODYA** sebagai business key. Pilih tipe changing attribute untuk kolom **NAMA\_KAB\_KODYA** dan **NAMA\_PROPINSI**.

#### 5. DIM\_PKERJAAN\_PRODI



Gambar 5.182. Data Flow Task DIM\_PKERJAAN\_PRODI – Incremental Load

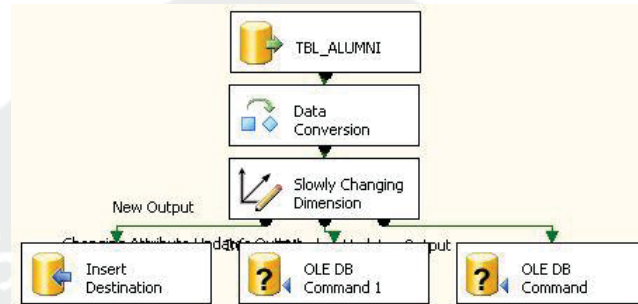
- Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_JABATAN]

- Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_Pekerjaan\_Prodi]  
 Pilih **ID\_JABATAN** dan **ID\_PRODI** sebagai business key. Pilih tipe changing attribute untuk kolom **JABATAN**.

## 6. DIM\_ALUMNI



**Gambar 5.183. Data Flow Task DIM\_ALUMNI – Incremental Load**

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

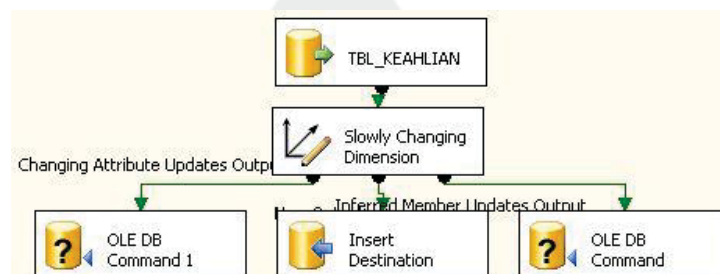
**ADO.NET Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[TBL\_ALUMNI]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*. Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_ALUMNI]

Pilih **ID\_ALUMNI** sebagai business key. Pilih tipe changing attribute untuk kolom **AWAL\_STUDI**, **GAJI\_PERTAMA**, **ID\_FAKULTAS**, **ID\_KAB\_KODYA**, **ID\_PRODI**, **JNS\_KEL**, **NPM**, dan **NAMA\_MHS**.

## 7. DIM\_KEAHLIAN



**Gambar 5.184. Data Flow Task DIM\_KEAHLIAN – Incremental Load**

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[TBL\_KEAHLIAN]

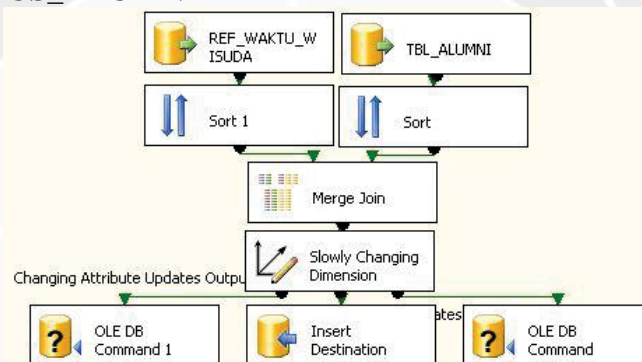
- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_KEAHLIAN\_ALUMNI]

Pilih **ID\_ALUMNI** dan **ID\_KEAHLIAN** sebagai business key. Pilih tipe changing attribute untuk kolom **LEVEL\_KEAHLIAN**, **NAMA**, dan **TAHUN**.

## 8. FACT\_LULUS\_ALUMNI



**Gambar 5.185.** Data Flow Task FACT\_LULUS\_ALUMNI – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

### a) TBL\_ALUMNI

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[TBL\_ALUMNI]

### b) REF\_WAKTU\_WISUDA

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of the table or view** : [dbo].[REF\_WAKTU\_WISUDA]

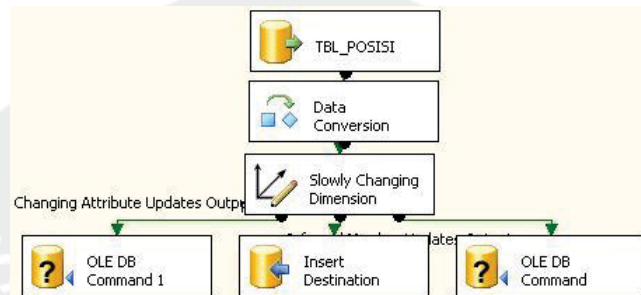
- b. Pada *data flow task* **TBL\_ALUMNI** dan **REF\_WAKTU\_WISUDA** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **TGL\_WISUDA** yang telah diurutkan untuk mengambil data **ID\_WAKTU\_WISUDA**.

- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan dihubungkan dengan *ADO.NET.SOURCE*.

Lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[FACT\_LULUS\_ALUMNI]  
 Pilih **ID\_ALUMNI** dan **ID\_WAKTU\_LULUS** sebagai business key. Pilih tipe changing attribute untuk kolom **IPK\_LULUS**.

## 9. DIM\_PKERJAAN



**Gambar 5.186. Data Flow Task DIM\_PKERJAAN – Incremental Load**

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

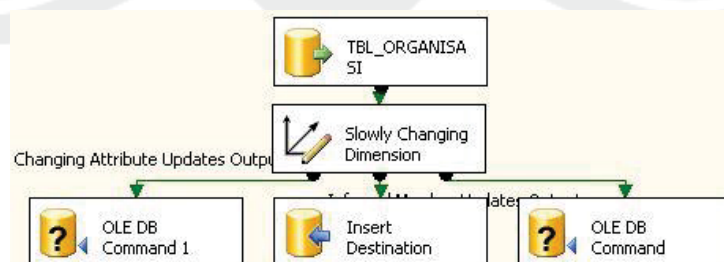
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[TBL\_POSISI]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_PKERJAAN\_ALUMNI]

Pilih **ID\_ALUMNI** dan **ID\_POSISI** sebagai business key. Pilih tipe changing attribute untuk kolom **BIDANG, DIVISI, JABATAN, PERUSAHAAN, THN\_KELUAR,** dan **THN\_MASUK**.

## 10. DIM\_ORGANISASI



**Gambar 5.187. Data Flow Task DIM\_ORGANISASI – Incremental Load**

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

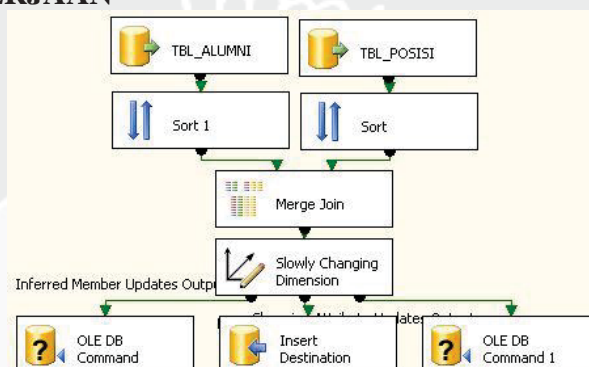
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[TBL\_ORGANISASI]

- b. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[DIM\_ORGANISASI\_ALUMNI]

Pilih **ID\_ALUMNI** dan **ID\_ORGANISASI** sebagai business key. Pilih tipe changing attribute untuk kolom **JABATAN**, **NAMA\_ORGANISASI**, **THN\_MULAI** dan **THN\_SELESAI**.

## 11. FACT\_Pekerjaan



Gambar 5.188. Data Flow Task FACT\_Pekerjaan – Incremental Load

- a. Data yang akan di-load bersumber dari *server*, maka tarik dan letakkan ikon *ADO.NET SOURCE* dan lakukan konfigurasi sebagai berikut:

a) TBL\_POSISI  
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[TBL\_POSISI]

b) TBL\_ALUMNI  
**Connection Manager** : DATA-WAREHOUSE.SIKMA\_STAGING  
**Data Access Mode** : Table or view  
**Name of table or view** : [dbo].[TBL\_ALUMNI]

- b. Pada *data flow task* **TBL\_POSISI** dan **TBL\_ALUMNI** digabungkan dengan fungsi *merge join*. Masing-masing *ADO.NET Source* harus diurutkan menggunakan fungsi *sort* terlebih dahulu. Data digabungkan berdasarkan **ID\_ALUMNI** yang telah diurutkan untuk mengambil data **ID\_POSISI**.

- c. Mendefinisikan *slowly changing dimension* dengan tarik dan letakkan *slowly changing dimension* dan lakukan konfigurasi berikut:

**Connection Manager** : DATA-WAREHOUSE.SIKMA\_DW  
**Use a Table or View** : [dbo].[FACT\_Pekerjaan]

Pilih **ID\_ALUMNI** dan **ID\_Pekerjaan** sebagai business key. Pilih tipe changing attribute untuk kolom **KESESUAIAN**.



12). *Sequence Container* **LOAD2**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD2**.

13). *Execute SQL Task* pada **LOAD2**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD2** sebanyak 4 kali dan beri nama **UPDATE\_KESESUAIAN**, **UPDATE\_KAB\_KODYA**, **UPDATE\_IPK**, dan **UPDATE\_TAHUN**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI**). Masing-masing *Execute SQL Task* mempunyai fungsi sebagai berikut:

- a. Ikon *Execute SQL Task* **UPDATE\_KESESUAIAN** berfungsi menjalankan *query* untuk mengubah data kesesuaian pekerjaan yang ada pada *database* **SIKMA\_DW\_ALUMNI**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE dbo.FACT_PEKERJAAN SET KESESUAIAN=1 WHERE (ID_
PEKERJAAN IN (SELECT D.ID_PEKERJAAN FROM dbo.DIM_PEKERJAAN_
ALUMNI AS D INNER JOIN dbo.DIM_ALUMNI AS A ON D.ID_ALUMNI =
A.ID_ALUMNI INNER JOIN dbo.DIM_PRODI_ALUMNI AS PRO ON A.ID_PRODI
= PRO.ID_PRODI WHERE(D.JABATAN IN (SELECT JABATAN FROM
dbo.DIM_PEKERJAAN_PRODI))));
```

- b. Ikon *Execute SQL Task* **UPDATE\_KAB\_KODYA** berfungsi untuk mengubah data id kabupaten kodya yang ada pada *database* **SIKMA\_DW\_ALUMNI**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE DIM_ALUMNI SET ID_KAB_KODYA=1 WHERE ID_ALUMNI IN(
SELECT ID_ALUMNI FROM DIM_ALUMNI WHERE ID_KAB_KODYA IS NULL);
```

- c. Ikon *Execute SQL Task* **UPDATE\_IPK** berfungsi untuk mengubah data IPK yang ada pada *database* **SIKMA\_DW\_ALUMNI**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE FACT_LULUS_ALUMNI SET IPK_LULUS=0 WHERE ID_ALUMNI IN(
SELECT ID_ALUMNI FROM FACT_LULUS_ALUMNI WHERE IPK_LULUS IS
NULL);
```

- d. Ikon *Execute SQL Task* **UPDATE\_TAHUN** berfungsi untuk menghapus data tarif selain id tarif 10 yang ada pada *database* **SIKMA\_DW\_ALUMNI**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE DIM_Pekerjaan_Alumni SET THN_MASUK=0 WHERE
THN_MASUK IS NULL; UPDATE DIM_Pekerjaan_Alumni SET
THN_KELUAR=0 WHERE THN_KELUAR IS NULL;
```

14). *Sequence Container* **LOAD3**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD3**.

15). *Execute SQL Task* **UPDATE\_KESESUAIAN1**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD3** dan beri nama **UPDATE\_KESESUAIAN1**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI**). Ikon *Execute SQL Task* **UPDATE\_TIDAK\_SESUI** berfungsi menjalankan *query* untuk mengubah data pekerjaan yang tidak sesuai pada *database* **SIKMA\_DW\_ALUMNI**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE FACT_Pekerjaan SET KESESUAIAN=1 WHERE ID_Pekerjaan IN(
148,88,90,93,95,96,98,99,100,101,102,103,104,105,106,108,109,
110,111,112,114,115,117,118,119,122,123,124,125,126,127,129);
```

16). *Sequence Container* **LOAD4**

Tarik dan letakkan ikon *Sequence Container* dan beri nama **LOAD4**.

17). *Execute SQL Task* **UPDATE\_TIDAK\_SESUI**

Tarik dan letakkan ikon *Execute SQL Task* ke dalam item **LOAD4** dan beri nama **UPDATE\_TIDAK\_SESUI**. Pilih koneksi tersebut (**DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI**). Ikon *Execute SQL Task* **UPDATE\_KESESUAIAN1** berfungsi menjalankan *query* untuk mengubah data kesesuaian pekerjaan yang ada pada *database* **SIKMA\_DW\_ALUMNI**. Lakukan konfigurasi dengan menuliskan sintaks berikut:

```
UPDATE FACT_Pekerjaan SET KESESUAIAN=0 WHERE KESESUAIAN IS NULL
```

## 5.2. Implementasi Cube

Setelah pembuatan *data warehouse* selesai dibangun, agar data dalam *data warehouse* dapat digunakan untuk mendukung dalam pengambilan keputusan, data-data tersebut harus dianalisis terlebih dahulu. Untuk melakukan analisis maka dibangunlah sebuah *cube*. *Cube* tersebut dapat dipergunakan untuk membuat *report* yang akan digunakan untuk membantu dalam pengambilan keputusan strategis perusahaan. *Tools* yang digunakan dalam pembuatan *cube* untuk

kepentingan analisis data adalah *tools* yang disediakan oleh SQL Server 2008 R2 yaitu SQL Server Analysis Service.

### 5.2.1. SIKMA\_SPAMA

Langkah-langkah dalam pembangunan *cube* **SIKMA\_SPAMA** adalah mendefinisikan *data source*, mendefinisikan *data source view*, mendefinisikan *cube*, menambah atau mengubah *measure*, atribut, dan dimensi, mendefinisikan kalkulasi pada *cube*, dan kemudian men-*deploy* proyek.

#### 5.2.1.1. Mendefinisikan *Data Source* SIKMA\_SPAMA

*Data Source* merepresentasikan koneksi ke data yang akan digunakan dalam analisis. Langkah-langkah untuk menciptakan *data source* adalah sebagai berikut :

1. Klik kanan **Data Source** pada **Solution Explorer** dan pilih **New Data Source**.
2. Mendefinisikan koneksi yang akan digunakan. Pada **Data Connection** pilih **DATA-WAREHOUSE.SIKMA\_DW**. Hal ini berarti koneksi tersebut akan menghubungkan dengan *database* **DATA-WAREHOUSE** pada *server* **SIKMA\_DW**.
3. Pada *dialog box* **Data Source Wizard**, pilih **Use a specific Windows user name and password**. Masukkan user name dan password.
4. Beri nama *data source* yang dibuat dengan **spama**. Maka terbentuklah koneksi ke *database* **spama**.

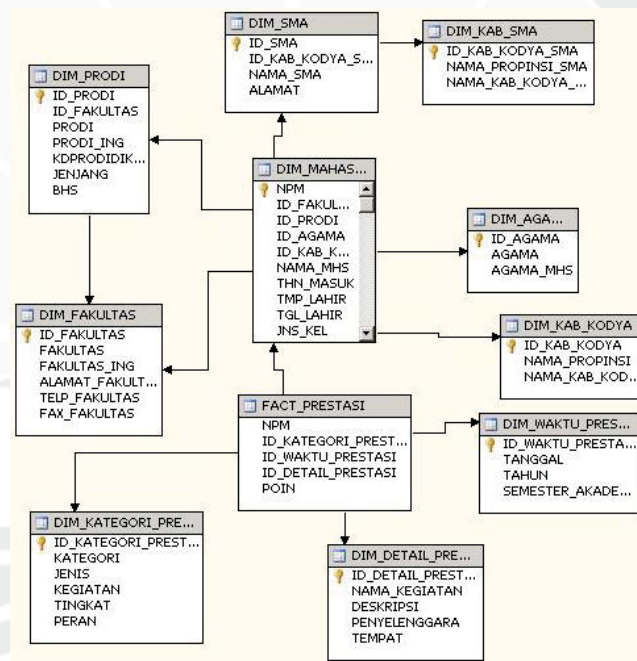
#### 5.2.1.2. Mendefinisikan *Data Source View* SIKMA\_SPAMA

*Data Source View* digunakan untuk fitur-fitur tingkat tinggi seperti *caching metadata*, menambah *relationship*, menciptakan kalkulasi dan mengeset kunci logika (*logical key*). Langkah-langkah untuk menciptakan *Data Source View* adalah sebagai berikut:

1. Klik kanan **Data Source View** pada **Solution Explorer** dan pilih **New Data Source View**
2. Mendefinisikan *Data Source* yang akan digunakan. Pada **Relational Data Source** pilih **spama** (*Data Source* yang sudah dibuat sebelumnya).

3. Tentukan objek dari *relational database* yang akan digunakan dalam *data source view*. Tabel yang tersedia adalah tabel yang terdapat dalam *data warehouse* (*database SIKMA\_DW*). Pilih tabel dimensi **DIM\_KATEGORI\_PRESTASI**, **DIM\_WAKTU\_PRESTASI**, **DIM\_MAHASISWA**, **DIM\_DETAIL\_PRESTASI**, **DIM\_KAB\_SMA**, **DIM\_WAKTU**, **DIM\_SMA**, **DIM\_FAKULTAS**, **DIM\_PRODI**, **DIM\_AGAMA**, dan **DIM\_KAB\_KODYA** dan tabel fakta **FACT\_PRESTASI** karena seluruh tabel tersebut akan digunakan dalam analisis.
4. Beri nama *data source view* yang dibuat dengan **spama**.

Setelah *Data Source View* terbentuk, dapat diamati keterkaitan antar tabel pada *Data Source Pane* (**spama.dsv [Design]**).



Gambar 5.189. Keterkaitan Antar Tabel SIKMA

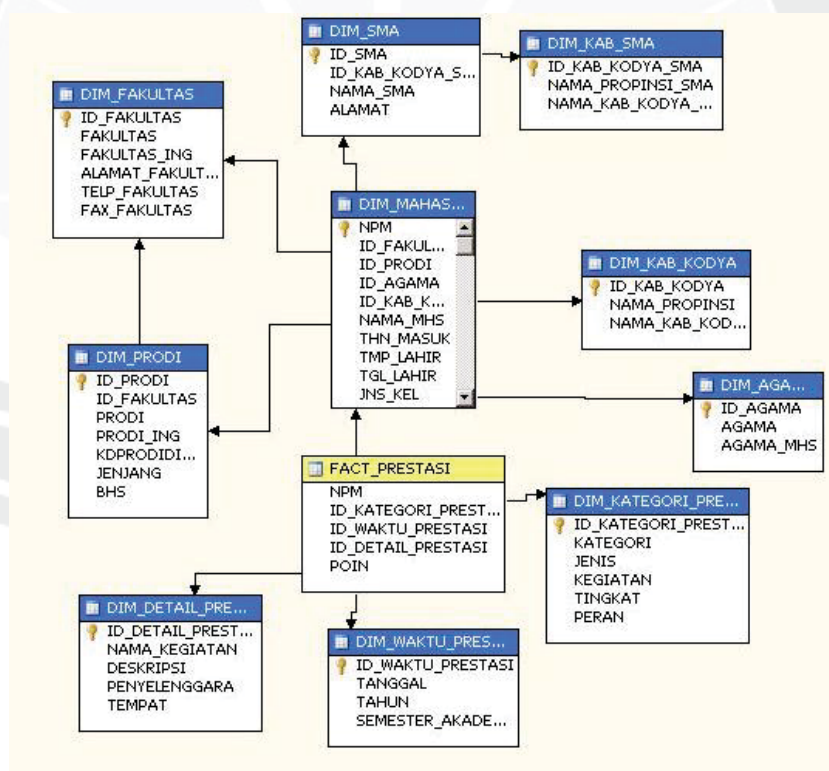
### 5.2.1.3. Mendefinisikan Cube SIKMA\_SPAMA

Langkah-langkah untuk menciptakan *cube* adalah sebagai berikut:

1. Klik kanan **Cubes** pada **Solution Explorer** dan pilih **New Cube**.
2. Memilih metode yang akan digunakan untuk membangun *cube*. Karena *cube* akan dibangun menggunakan *Data Source* yang sudah didefinisikan sebelumnya, Pilih **Use existing tables**.

3. Mendefinisikan measure group yang akan digunakan. Pilih **spama** (*Data Source View* yang sudah dibuat sebelumnya). Centang **FACT\_PRESTASI** sebagai measure group yang dipilih.
4. Menentukan measure yang digunakan. Centang **FACT PRESTASI**, **POIN**, dan **FACT PRESTASI Count**.
5. Menentukan *dimension* yang digunakan. Centang **DIM\_MAHASISWA**, **DIM\_DETAIL\_PRESTASI**, **DIM\_KATEGORI\_PRESTASI**, dan **DIM\_WAKTU\_PRESTASI**.
6. Beri nama *cube* yang dibuat dengan **spama**.

Setelah *cube* selesai dibuat, akan muncul *cube designer* dengan nama sesuai dengan nama *cube* yaitu **spama.Cube [Design]**. **Dimension** pada *Solution Explorer* secara otomatis terisi dengan dimensi-dimensi yang ada dalam *cube*. *SQL Server* akan mengubah warna-warna tabel pada *Data Source view* menjadi biru untuk tabel dimensi dan kuning untuk tabel fakta.



Gambar 5.190. Hasil Pembuatan *Cube*



#### 5.2.1.4. Memodifikasi Dimensi, Measure, Atribut, dan Hirarki SIKMA\_SPAMA

Tahap ini diperlukan untuk mengubah *cube* sesuai dengan kebutuhan analisis dan mengubah data agar lebih mudah digunakan dalam analisis. Modifikasi akan dilakukan pada setiap dimensi dan beberapa fakta yang ada pada *cube*.

##### 5.2.1.4.1. Modifikasi pada Dimensi DIM\_DETAIL\_PRESTASI

Modifikasi yang dilakukan pada **DIM\_DETAIL\_PRESTASI** adalah sebagai berikut:

###### 1. Mengubah Hirarki

Hirarki **DIM\_DETAIL\_PRESTASI** dengan tingkatan hirarki sebagai berikut:

ID DETAIL PRESTASI → NAMA KEGIATAN



Gambar 5.191. Hirarki pada Dimensi Detail Prestasi

##### 5.2.1.4.2. Modifikasi pada Dimensi DIM\_WAKTU\_PRESTASI

Modifikasi yang dilakukan pada **DIM\_WAKTU\_PRESTASI** adalah sebagai berikut:

###### 1. Mengubah Hirarki

Hirarki **DIM\_WAKTU\_PRESTASI** dengan tingkatan hirarki sebagai berikut:

TAHUN → SEMESTER AKADEMIK → TANGGAL



Gambar 5.192. Hirarki pada Dimensi Waktu Prestasi

### 5.2.1.4.3. Modifikasi pada Dimensi DIM\_KATEGORI\_PRESTASI

Modifikasi yang dilakukan pada **DIM\_KATEGORI\_PRESTASI** adalah sebagai berikut:

#### 1. Mengubah Hirarki

Hirarki **DIM\_KATEGORI\_PRESTASI** dengan tingkatan hirarki sebagai berikut:

KATEGORI → KEGIATAN → JENIS → TINGKAT → PERAN



Gambar 5.193. Hirarki pada Dimensi Kategori Prestasi

### 5.2.1.4.4. Modifikasi pada Dimensi DIM\_MAHASISWA

Modifikasi yang dilakukan pada **DIM\_MAHASISWA** adalah sebagai berikut:

#### 1. Mengubah Hirarki

Hirarki **DIM\_MAHASISWA** dengan tingkatan hirarki sebagai berikut:

##### a. Hirarki Mahasiswa

NPM → NAMA MHS → THN MASUK → TMP LAHIR → TGL LAHIR  
→ NAMA PROPINSI → NAMA KAB KODYA

##### b. Hirarki Fakultas

FAKULTAS → PRODI

##### c. Hirarki Daerah SMA

NAMA PROPINSI SMA → NAMA KAB KODYA SMA



Gambar 5.194. Hirarki pada Dimensi Mahasiswa

### 5.2.1.5. Mendefinisikan Kalkulasi pada Cube

Sesuai dengan kebutuhan analisis, maka perlu didefinisikan beberapa *calculated member* pada *cube*. *Calculated member* merupakan anggota *measure* yang didefinisikan berdasarkan pada kombinasi data *cube*, operator aritmatika, angka-angka, dan fungsi. Sebagai contoh dapat diciptakan sebuah *calculated member* untuk menghitung penjumlahan dua *measure* pada *cube*. Didefinisi pada *calculated member* disimpan pada *cube*, tetapi nilai-nilainya dihitung pada waktu *query*. Ada satu kalkulasi yang dibuat yaitu kalkulasi untuk menghitung rata-rata poin spama. Berikut akan dijelaskan tahap-tahap pembuatan kalkulasi:

1. Akses *Cube Designer* spama dan klik tab **Calculations**.
2. Pada *toolbar* di calculation, pilih **new Calculated Member** untuk membuat *calculated member* baru.
3. Akan muncul panel **Calculation Expression**, lakukan modifikasi berikut:
  - a. **Rata-rata poin spama**

Kalkulasi ini dibuat untuk menghitung rata-rata poin spama setiap kali kegiatan yang dilakukan oleh mahasiswa yaitu dengan membagi *measure* spama dengan *measure* jumlah baris prestasi yang tidak sama dengan kosong. Spesifikasi pada *measure* ini:

**Name** : [Rata2poinSpama]  
**Parent Hierarchy** : Measure  
**Expression** : **IIF**([Measures].[FACT PRESTASI Count]=0,  
**null**,[Measures].[POIN]/[Measures].[FACT  
**PRESTASI Count**)  
**Format String** : "#,#"   
**Visible** : True  
**Associated Measure** : FACT PRESTASI

The screenshot shows the 'Calculation Expression' panel with the following settings:

- Name:** [Rata2Poin Spama]
- Parent Properties:** Parent hierarchy: Measures
- Expression:** IIF ([Measures].[FACT PRESTASI Count]=0, null, [Measures].[POIN] / [Measures].[FACT PRESTASI Count])
- Additional Properties:** Format string: '#,#', Visible: True, Associated measure group: FACT PRESTASI

**Gambar 5.195. Calculated Member Rata-Rata Poin Spama**

### 5.2.2. SIKMA\_BEASISWA

Langkah-langkah dalam pembangunan *cube* SIKMA\_BEASISWA adalah mendefinisikan *data source*, mendefinisikan *data source view*, mendefinisikan *cube*, menambah atau mengubah *measure*, atribut, dan dimensi, mendefinisikan kalkulasi pada *cube*, dan kemudian men-*deploy* proyek.

#### 5.2.2.1. Mendefinisikan *Data Source* SIKMA\_BEASISWA

*Data Source* merepresentasikan koneksi ke data yang akan digunakan dalam analisis. Langkah-langkah untuk menciptakan *Data Source* adalah sebagai berikut :

1. Klik kanan **Data Source** pada **Solution Explorer** dan pilih **New Data Source**.
2. Mendefinisikan koneksi yang akan digunakan. Pada **Data Connection** pilih **DATA-WAREHOUSE.SIKMA\_DW**. Hal ini berarti koneksi tersebut akan menghubungkan dengan *database* **DATA-WAREHOUSE** pada *server* **SIKMA\_DW**.
3. Pada *dialog box* **Data Source Wizard**, pilih **Use a specific Windows user name and password**. Masukkan user name dan password.
4. Beri nama *data source* yang dibuat dengan **SIKMA DW**. Maka terbentuklah koneksi ke *database* **SIKMA DW**.

#### 5.2.2.2. Mendefinisikan *Data Source View* SIKMA\_BEASISWA

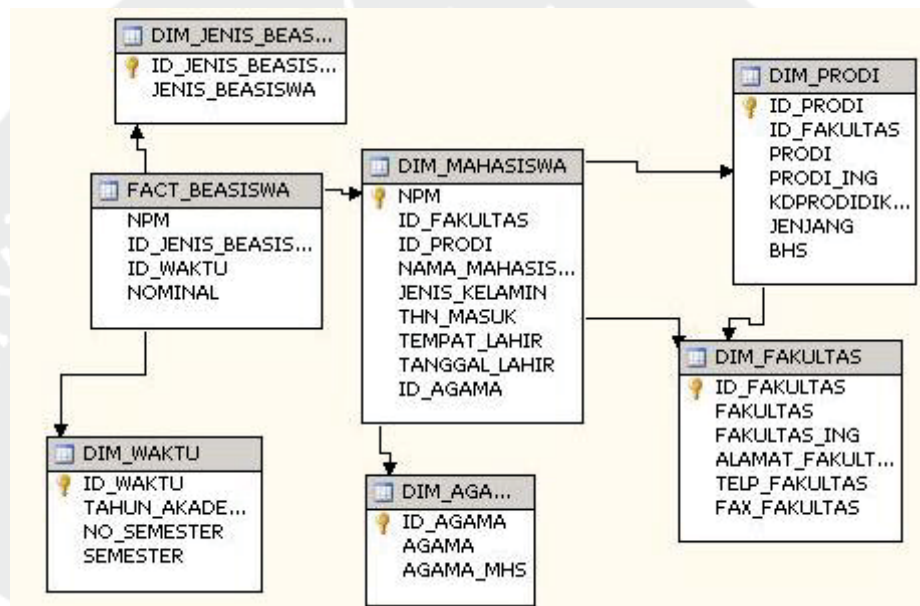
*Data Source View* digunakan untuk fitur-fitur tingkat tinggi seperti *caching metadata*, menambah *relationship*, menciptakan kalkulasi dan mengeset kunci logika (*logical key*). Langkah-langkah untuk menciptakan *Data Source View* adalah sebagai berikut:

1. Klik kanan **Data Source View** pada **Solution Explorer** dan pilih **New Data Source View**
2. Mendefinisikan *Data Source* yang akan digunakan. Pada **Relational Data Source** pilih **SIKMA DW** (*Data Source* yang sudah dibuat sebelumnya).
3. Tentukan objek dari *relational database* yang akan digunakan dalam *data source view*. Tabel yang tersedia adalah tabel yang terdapat dalam *data warehouse* (*database* **SIKMA\_DW**). Pilih tabel dimensi

**DIM\_MAHASISWA, DIM\_JENIS\_BEASISWA, DIM\_WAKTU, DIM\_AGAMA, DIM\_FAKULTAS, DIM\_KAB\_KODYA, DIM\_KAB\_SMA, DIM\_PRODI, dan DIM\_SMA** dan tabel fakta **FACT\_BEASISWA** karena seluruh tabel tersebut akan digunakan dalam analisis.

4. Beri nama *data source view* yang dibuat dengan **SIKMA DW**.

Setelah *Data Source View* terbentuk, dapat diamati keterkaitan antar tabel pada *Data Source Pane* (**SIKMA DW.dsv [Design]**).



**Gambar 5.196. Keterkaitan Antar Tabel Beasiswa**

### 5.2.2.3. Mendefinisikan *Cube* SIKMA\_BEASISWA

Langkah-langkah untuk menciptakan *cube* adalah sebagai berikut:

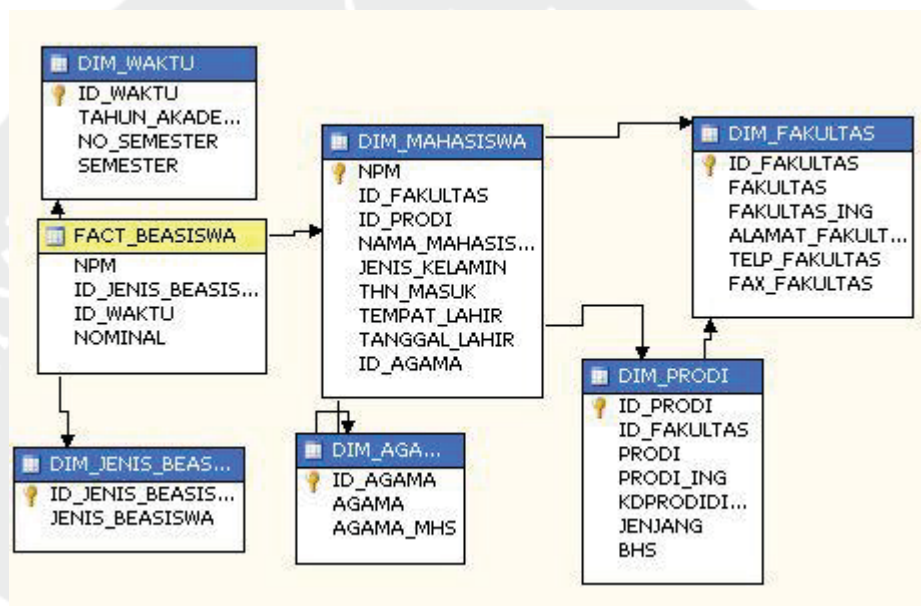
1. Klik kanan **Cubes** pada **Solution Explorer** dan pilih **New Cube**.
2. Memilih metode yang akan digunakan untuk membangun *cube*. Karena *cube* akan dibangun menggunakan *Data Source* yang sudah didefinisikan sebelumnya, Pilih **Use existing tables**.
3. Mendefinisikan *measure group* yang akan digunakan. Pilih **spama** (*Data Source View* yang sudah dibuat sebelumnya). Centang **FACT\_BEASISWA** sebagai *measure group* yang dipilih.
4. Menentukan *measure* yang digunakan. Centang **FACT BEASISWA**, **NOMINAL**, dan **FACT BEASISWA Count**.
5. Menentukan *dimension* yang digunakan. Centang **DIM\_WAKTU**, **DIM\_JENIS\_BEASISWA**, dan **DIM\_MAHASISWA**.



6. Beri nama *cube* yang dibuat dengan **SIKMA DW 1**.

Setelah *cube* selesai dibuat, akan muncul *cube designer* dengan nama sesuai dengan nama *cube* yaitu **SIKMA DW.Cube [Design]**. **Dimension** pada *Solution Explorer* secara otomatis terisi dengan dimensi-dimensi yang ada dalam *cube*.

SQL Server akan mengubah warna-warna tabel pada *Data Source view* menjadi biru untuk tabel dimensi dan kuning untuk tabel fakta.



**Gambar 5.197. Hasil Pembuatan *Cube* Beasiswa**

#### 5.2.2.4. Memodifikasi Dimensi, *Measure*, Atribut, dan Hirarki SIKMA\_BEASISWA

Tahap ini diperlukan untuk mengubah *cube* sesuai dengan kebutuhan analisis dan mengubah data agar lebih mudah digunakan dalam analisis. Modifikasi akan dilakukan pada setiap dimensi dan beberapa fakta yang ada pada *cube*.

##### 5.2.2.4.1. Modifikasi pada Dimensi DIM\_WAKTU

Modifikasi yang dilakukan pada **DIM\_WAKTU** adalah sebagai berikut:

1. Mengubah Hirarki

Hirarki **DIM\_WAKTU** dengan tingkatan hirarki sebagai berikut:

TAHUN AKADEMIK → SEMESTER



**Gambar 5.198. Hirarki pada Dimensi Waktu**

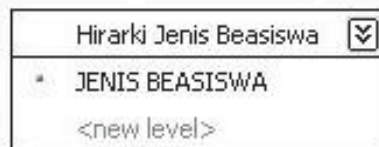
#### 5.2.2.4.2. Modifikasi pada Dimensi DIM\_JENIS\_BEASISWA

Modifikasi yang dilakukan pada **DIM\_JENIS\_BEASISWA** adalah sebagai berikut:

##### 1. Mengubah Hirarki

Hirarki **DIM\_JENIS\_BEASISWA** dengan tingkatan hirarki sebagai berikut:

JENIS BEASISWA



**Gambar 5.199. Hirarki pada Dimensi Jenis Beasiswa**

#### 5.2.2.4.3. Modifikasi pada Dimensi DIM\_MAHASISWA

Modifikasi yang dilakukan pada **DIM\_MAHASISWA** adalah sebagai berikut:

##### 1. Mengubah Hirarki

Hirarki **DIM\_MAHASISWA** dengan tingkatan hirarki sebagai berikut:

##### b. Hirarki Mahasiswa

NPM → NAMA MHS → THN MASUK → TMP LAHIR → TGL LAHIR  
→ NAMA PROPINSI → NAMA KAB KODYA

##### c. Hirarki Fakultas

FAKULTAS → PRODI

##### d. Hirarki Daerah SMA

NAMA PROPINSI SMA → NAMA KAB KODYA SMA



**Gambar 5.200. Hirarki pada Dimensi MAHASISWA**

#### 5.2.2.5. Mendefinisikan Kalkulasi pada *Cube*

Sesuai dengan kebutuhan analisis, maka perlu didefinisikan beberapa *calculated member* pada *cube*. *Calculated member* merupakan anggota *measure* yang didefinisikan berdasarkan pada kombinasi data *cube*, operator aritmatika, angka-angka, dan fungsi. Sebagai contoh dapat diciptakan sebuah *calculated member* untuk menghitung penjumlahan dua *measure* pada *cube*. Definisi pada *calculated member* disimpan pada *cube*, tetapi nilai-nilainya dihitung pada waktu *query*. Ada dua kalkulasi yang dibuat yaitu kalkulasi untuk menghitung rata-rata nominal beasiswa dan jumlah nominal mahasiswa. Berikut akan dijelaskan tahap-tahap pembuatan kalkulasi:

1. Akses *Cube Designer* SIKMA DW dan klik tab **Calculations**.
2. Pada *toolbar* di *calculation*, pilih **new Calculated Member** untuk membuat *calculated member* baru.
3. Akan muncul panel **Calculation Expression**, lakukan modifikasi berikut:

##### a. Rata-rata nominal beasiswa

Kalkulasi ini dibuat untuk menghitung rata-rata nominal beasiswa yang diperoleh mahasiswa setiap kali mahasiswa memperoleh beasiswa yaitu dengan membagi *measure* nominal beasiswa dengan *measure* jumlah baris beasiswa yang tidak sama dengan kosong. Spesifikasi pada *measure* ini:

|                           |                                                                                                       |
|---------------------------|-------------------------------------------------------------------------------------------------------|
| <b>Name</b>               | : [Rata2Nominal Beasiswa]                                                                             |
| <b>Parent Hierarchy</b>   | : Measure                                                                                             |
| <b>Expression</b>         | : IIF([Measures].[FACT BEASISWA Count]=0, null,[Measures].[NOMINAL]/[Measures].[FACT BEASISWA Count]) |
| <b>Format String</b>      | : "Rp #,##0.00"                                                                                       |
| <b>Visible</b>            | : True                                                                                                |
| <b>Associated Measure</b> | : FACT BEASISWA                                                                                       |

**Gambar 5.201. Calculated Member Rata-Rata Nominal Beasiswa**

#### b. Nominal beasiswa

Kalkulasi ini dibuat untuk menghitung nominal beasiswa yang diperoleh mahasiswa setiap kali mahasiswa memperoleh beasiswa yaitu dengan mengkali measure nominal beasiswa dengan 1. Spesifikasi pada measure ini:

**Name** : [Nominal(RP)]  
**Parent Hierarchy** : Measure  
**Expression** : IIF([Measures].[FACT BEASISWA Count]=0, null, [Measures].[NOMINAL]\*1)  
**Format String** : "Rp #,##0.00"  
**Visible** : True  
**Associated Measure** : FACT BEASISWA

**Gambar 5.202. Calculated Member Nominal Beasiswa**

### 5.2.3. SIKMA\_ALUMNI

Langkah-langkah dalam pembangunan *cube* SIKMA\_ALUMNI adalah mendefinisikan *data source*, mendefinisikan *data source view*, mendefinisikan

*cube*, menambah atau mengubah *measure*, atribut, dan dimensi, mendefinisikan kalkulasi pada *cube*, dan kemudian *men-deploy* proyek.

#### 5.2.3.1. Mendefinisikan *Data Source* SIKMA\_ALUMNI

*Data Source* merepresentasikan koneksi ke data yang akan digunakan dalam analisis. Langkah-langkah untuk menciptakan *Data Source* adalah sebagai berikut :

1. Klik kanan **Data Source** pada **Solution Explorer** dan pilih **New Data Source**.
2. Mendefinisikan koneksi yang akan digunakan. Pada **Data Connection** pilih **DATA-WAREHOUSE.SIKMA\_DW\_ALUMNI**. Hal ini berarti koneksi tersebut akan menghubungkan dengan *database* **DATA-WAREHOUSE** pada *server* **SIKMA\_DW\_ALUMNI**.
3. Pada *dialog box* **Data Source Wizard**, pilih **Use a specific Windows user name and password**. Masukkan user name dan password.
4. Beri nama *data source* yang dibuat dengan **alumni**. Maka terbentuklah koneksi ke *database* **alumni**.

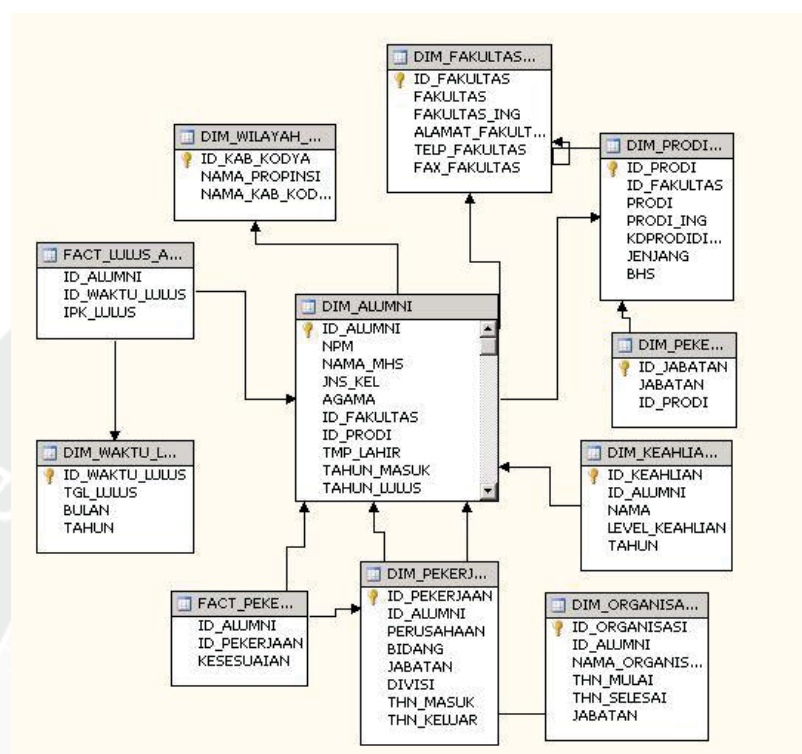
#### 5.2.3.2. Mendefinisikan *Data Source View* SIKMA\_ALUMNI

*Data Source View* digunakan untuk fitur-fitur tingkat tinggi seperti *caching metadata*, menambah *relationship*, menciptakan kalkulasi dan mengeset kunci logika (*logical key*). Langkah-langkah untuk menciptakan *Data Source View* adalah sebagai berikut:

1. Klik kanan **Data Source View** pada **Solution Explorer** dan pilih **New Data Source View**
2. Mendefinisikan *Data Source* yang akan digunakan. Pada **Relational Data Source** pilih **alumni** (*Data Source* yang sudah dibuat sebelumnya).
3. Tentukan objek dari *relational database* yang akan digunakan dalam *data source view*. Tabel yang tersedia adalah tabel yang terdapat dalam *data warehouse* (*database* **SIKMA\_DW\_ALUMNI**). Pilih semua tabel dimensi dan tabel fakta karena seluruh tabel tersebut akan digunakan dalam analisis.
4. Beri nama *data source view* yang dibuat dengan **alumni**.



Setelah *Data Source View* terbentuk, dapat diamati keterkaitan antar tabel pada *Data Source Pane* (**alumni.dsv [Design]**).



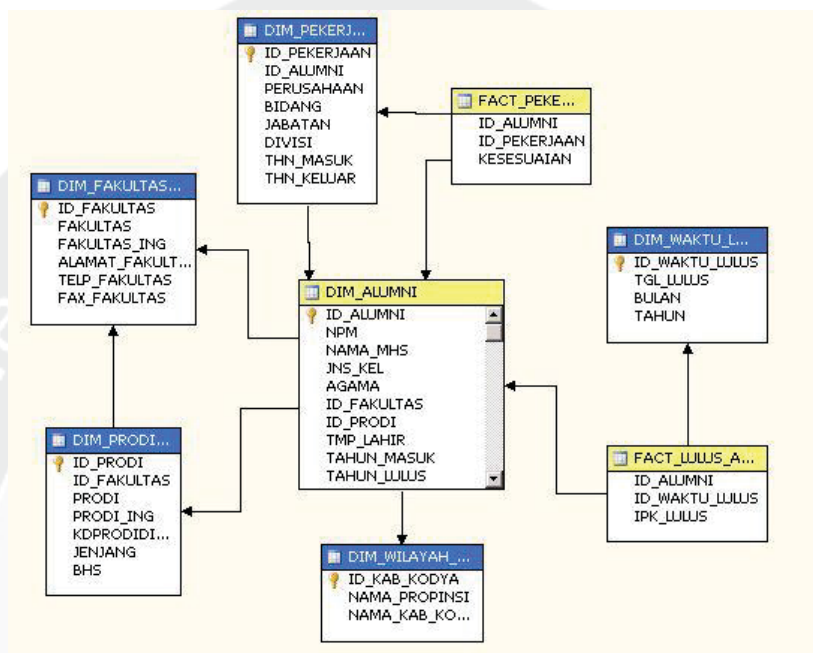
**Gambar 5.203. Keterkaitan Antar Tabel ALUMNI**

### 5.2.3.3. Mendefinisikan *Cube* SIKMA\_ALUMNI

Langkah-langkah untuk menciptakan *cube* adalah sebagai berikut:

1. Klik kanan **Cubes** pada **Solution Explorer** dan pilih **New Cube**.
2. Memilih metode yang akan digunakan untuk membangun *cube*. Karena *cube* akan dibangun menggunakan *Data Source* yang sudah didefinisikan sebelumnya, Pilih **Use existing tables**.
3. Mendefinisikan measure group yang akan digunakan. Pilih **alumni** (*Data Source View* yang sudah dibuat sebelumnya). Centang **FACT\_Pekerjaan** dan **FACT\_Lulus\_Alumni** sebagai measure group yang dipilih.
4. Menentukan measure yang digunakan. Centang **FACT Lulus Alumni**, **IPK Lulus**, **FACT Lulus Alumni Count**, **FACT Pekerjaan**, **KESESUAIAN**, dan **FACT Pekerjaan Count**.
5. Menentukan *dimension* yang digunakan. Centang **DIM\_ALUMNI**, **DIM\_WAKTU\_LULUS**, dan **DIM\_Pekerjaan\_Alumni**.
6. Beri nama *cube* yang dibuat dengan **alumni**.

Setelah *cube* selesai dibuat, akan muncul *cube designer* dengan nama sesuai dengan nama *cube* yaitu **alumni.Cube [Design]**. **Dimension** pada *Solution Explorer* secara otomatis terisi dengan dimensi-dimensi yang ada dalam *cube*. SQL Server akan mengubah warna-warna tabel pada *Data Source view* menjadi biru untuk tabel dimensi dan kuning untuk tabel fakta.



**Gambar 5.204. Hasil Pembuatan Cube Alumni**

#### 5.2.3.4. Memodifikasi Dimensi, Measure, Atribut, dan Hirarki SIKMA\_ALUMNI

Tahap ini diperlukan untuk mengubah *cube* sesuai dengan kebutuhan analisis dan mengubah data agar lebih mudah digunakan dalam analisis. Modifikasi akan dilakukan pada setiap dimensi dan beberapa fakta yang ada pada *cube*.

#### 5.2.3.5. Mendefinisikan Kalkulasi pada Cube

Sesuai dengan kebutuhan analisis, maka perlu didefinisikan beberapa *calculated member* pada *cube*. *Calculated member* merupakan anggota *measure* yang didefinisikan berdasarkan pada kombinasi data *cube*, operator aritmatika, angka-angka, dan fungsi. Sebagai contoh dapat diciptakan sebuah *calculated member* untuk menghitung penjumlahan dua *measure* pada *cube*. Didefinisi pada *calculated member* disimpan pada *cube*, tetapi nilai-nilainya dihitung pada waktu

*query*. Ada tujuh kalkulasi yang dibuat yaitu kalkulasi untuk menghitung rata-rata IPK alumni, minimal gaji alumni (Rp), maksimal gaji alumni (Rp), rata-rata gaji pertama alumni, rata-rata lama pembuatan tugas akhir, rata-rata lama masa studi alumni, dan persentase kesesuaian pekerjaan alumni dengan prodi alumni. Berikut akan dijelaskan tahap-tahap pembuatan kalkulasi:

1. Akses *Cube Designer* SIKMA DW ALUMNI dan klik tab **Calculations**.
2. Pada *toolbar* di *calculation*, pilih **new Calculated Member** untuk membuat *calculated member* baru.
3. Akan muncul panel **Calculation Expression**, lakukan modifikasi berikut:

#### a. Rata-rata IPK alumni

Kalkulasi ini dibuat untuk menghitung rata-rata IPK alumni yaitu dengan membagi *measure* IPK alumni dengan *measure* jumlah alumni.

Spesifikasi pada *measure* ini:

**Name** : [Rata2 IPK]  
**Parent Hierarchy** : Measure  
**Expression** : IIF([Measures].[JUMLAH ALUMNI]=0,null,[Measures].[IPK LULUS]/[Measures].[JUMLAH ALUMNI])  
**Format String** : "#,#.00"  
**Visible** : True  
**Associated Measure** : FACT LULUS ALUMNI

The screenshot shows a configuration window for a calculated member. The 'Name' field contains '[Rata2 IPK]'. Under 'Parent Properties', 'Parent hierarchy' is set to 'Measures'. The 'Expression' field contains the DAX formula: `IIF ([Measures].[JUMLAH ALUMNI]=0, null, [Measures].[IPK LULUS] / [Measures].[JUMLAH ALUMNI])`. Under 'Additional Properties', 'Format string' is set to '#,#.00', 'Visible' is set to 'True', and 'Associated measure group' is set to 'FACT LULUS ALUMNI'.

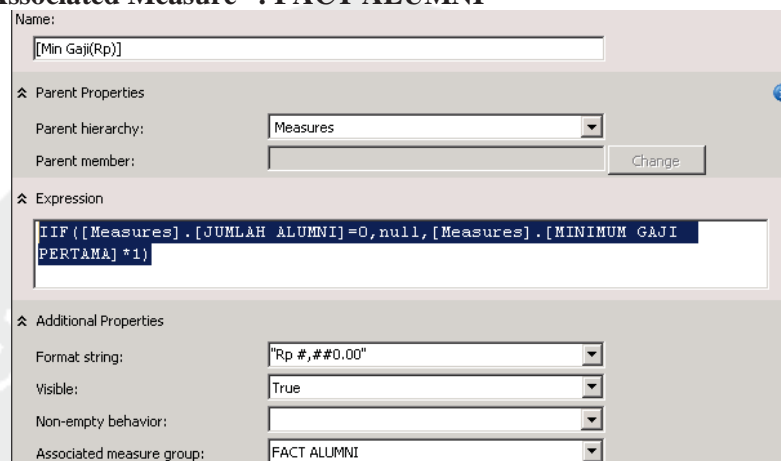
**Gambar 5.205. Calculated Member Rata-Rata IPK**

#### b. Minimal gaji alumni (Rp)

Kalkulasi ini dibuat untuk menghitung nominal minimal gaji alumni yaitu dengan mengkali *measure* minumum gaji pertama dengan 1.

Spesifikasi pada *measure* ini:

**Name** : [Min Gaji(Rp)]  
**Parent Hierarchy** : Measure  
**Expression** : IIF([Measures].[JUMLAH ALUMNI]=0,null,[Measures].[MINIMUM GAJI PERTAMA]\*1)  
**Format String** : "Rp #,##0.00"  
**Visible** : True  
**Associated Measure** : FACT ALUMNI



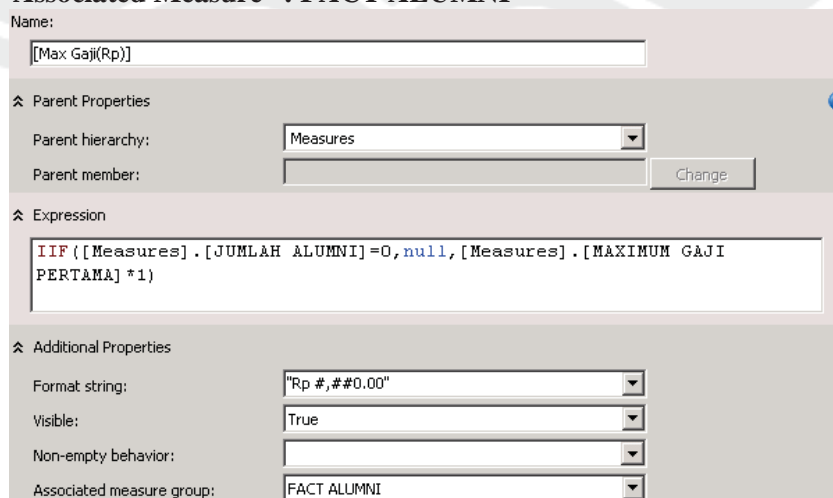
Name: [Min Gaji(Rp)]  
 Parent hierarchy: Measures  
 Parent member:  Change  
 Expression: IIF ([Measures] . [JUMLAH ALUMNI]=0, null, [Measures] . [MINIMUM GAJI PERTAMA] \*1)  
 Format string: "Rp #,##0.00"  
 Visible: True  
 Non-empty behavior:   
 Associated measure group: FACT ALUMNI

**Gambar 5.206. Calculated Member Minimal Gaji Alumni**

**c. Maximal gaji alumni (Rp)**

Kalkulasi ini dibuat untuk menghitung nominal maksimal gaji alumni yaitu dengan mengkali *measure* maksimal gaji pertama dengan 1. Spesifikasi pada measure ini:

**Name** : [Max Gaji(Rp)]  
**Parent Hierarchy** : Measure  
**Expression** : IIF([Measures].[JUMLAH ALUMNI]=0,null,[Measures].[MAXIMUM GAJI PERTAMA]\*1)  
**Format String** : "Rp #,##0.00"  
**Visible** : True  
**Associated Measure** : FACT ALUMNI



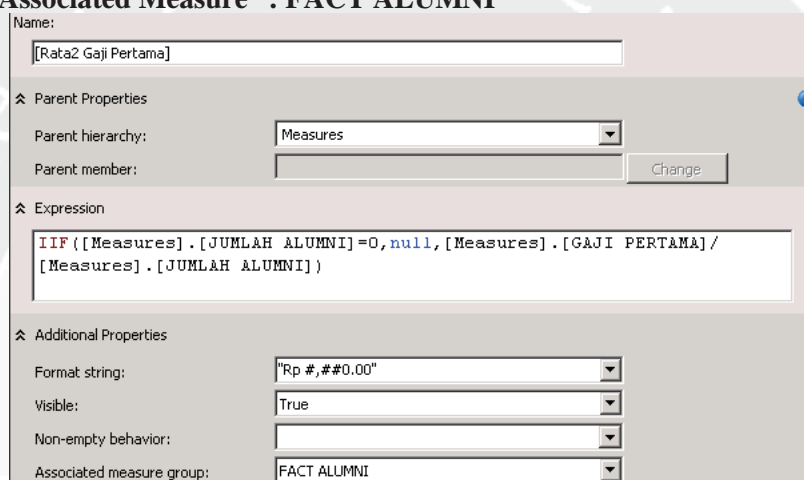
Name: [Max Gaji(Rp)]  
 Parent hierarchy: Measures  
 Parent member:  Change  
 Expression: IIF ([Measures] . [JUMLAH ALUMNI]=0, null, [Measures] . [MAXIMUM GAJI PERTAMA] \*1)  
 Format string: "Rp #,##0.00"  
 Visible: True  
 Non-empty behavior:   
 Associated measure group: FACT ALUMNI

**Gambar 5.207. Calculated Member Maximal Gaji Alumni**

#### d. Rata-rata gaji pertama alumni

Kalkulasi ini dibuat untuk menghitung rata-rata gaji pertama yang diperoleh alumni yaitu dengan membagi *measure* gaji pertama dengan jumlah alumni. Spesifikasi pada measure ini:

**Name** : [Rata2 Gaji Pertama]  
**Parent Hierarchy** : Measure  
**Expression** : `IIF([Measures].[JUMLAH ALUMNI]=0,null,[Measures].[GAJI PERTAMA]/[Measures].[JUMLAH ALUMNI])`  
**Format String** : "Rp #,##0.00"  
**Visible** : True  
**Associated Measure** : FACT ALUMNI



The screenshot shows the configuration for a calculated member in SQL Server Data Tools. The 'Name' field contains '[Rata2 Gaji Pertama]'. Under 'Parent Properties', the 'Parent hierarchy' is set to 'Measures' and 'Parent member' is empty. The 'Expression' field contains the DAX formula: `IIF([Measures].[JUMLAH ALUMNI]=0,null,[Measures].[GAJI PERTAMA]/[Measures].[JUMLAH ALUMNI])`. Under 'Additional Properties', the 'Format string' is set to '"Rp #,##0.00"', 'Visible' is set to 'True', 'Non-empty behavior' is empty, and 'Associated measure group' is set to 'FACT ALUMNI'.

**Gambar 5.208. Calculated Member Rata-Rata Gaji Pertama**

#### e. Rata-rata lama pembuatan tugas akhir

Kalkulasi ini dibuat untuk menghitung rata-rata lama pembuatan tugas akhir yaitu dengan membagi *measure* lama tugas akhir dengan jumlah alumni. Spesifikasi pada measure ini:

**Name** : [RATA2 LAMA TA]  
**Parent Hierarchy** : Measure  
**Expression** : `IIF([Measures].[JUMLAH ALUMNI]=0,null,[Measures].[LAMA TA]/[Measures].[JUMLAH ALUMNI])`  
**Format String** : "#,##.00"  
**Visible** : True  
**Associated Measure** : FACT ALUMNI



**Gambar 5.209. Calculated Member Rata-Rata Pembuatan Tugas Akhir**

**f. Rata-rata lama masa studi alumni**

Kalkulasi ini dibuat untuk menghitung rata-rata lama masa studi alumni yaitu dengan membagi *measure* lama studi dengan jumlah alumni.

Spesifikasi pada measure ini:

**Name** : [RATA2 LAMA STUDI]  
**Parent Hierarchy** : Measure  
**Expression** : IIF([Measures].[JUMLAH ALUMNI]=0,null,[Measures].[LAMA STUDI]/[Measures].[JUMLAH ALUMNI])  
**Format String** : "#,#.00"  
**Visible** : True  
**Associated Measure** : FACT ALUMNI

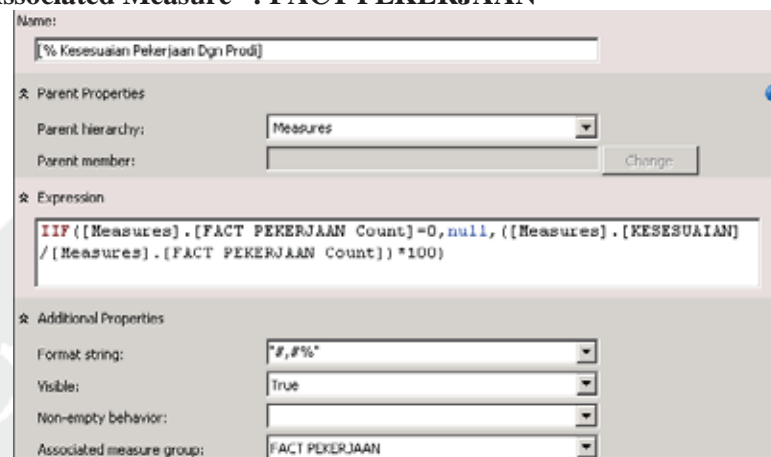
**Gambar 5.210. Calculated Member Rata-Rata Lama Masa Studi Alumni**

**g. Persentase kesesuaian pekerjaan alumni dengan prodi alumni**

Kalkulasi ini dibuat untuk menghitung persentase kesesuaian pekerjaan alumni dengan prodi alumni yaitu dengan membagi *measure* kesesuaian dengan fakta pekerjaan dan dikali 100. Spesifikasi pada measure ini:

**Name** : [% Kesesuaian Pekerjaan Dgn Prodi]  
**Parent Hierarchy** : Measure

**Expression** : **IIF**([Measures].[FACT PEKERJAAN Count]=0,  
**null**,([Measures].[KESESUAIAN]/[Measures].[FACT  
 PEKERJAAN Count])\*100)  
**Format String** : "#,##%"  
**Visible** : **True**  
**Associated Measure** : **FACT PEKERJAAN**



**Gambar 5.211. Calculated Member Persentase Kesesuaian Pekerjaan Alumni dengan Prodi**

### 5.3. Pembuatan Report

Setelah *cube* selesai dibuat, *cube* tersebut digunakan untuk membuat *report* yang akan digunakan untuk membantu dalam pengambilan keputusan strategis universitas. Report dibangun dengan menggunakan *tool* Report Portal yang terhubung dengan *cube* yang telah dibangun sebelumnya.

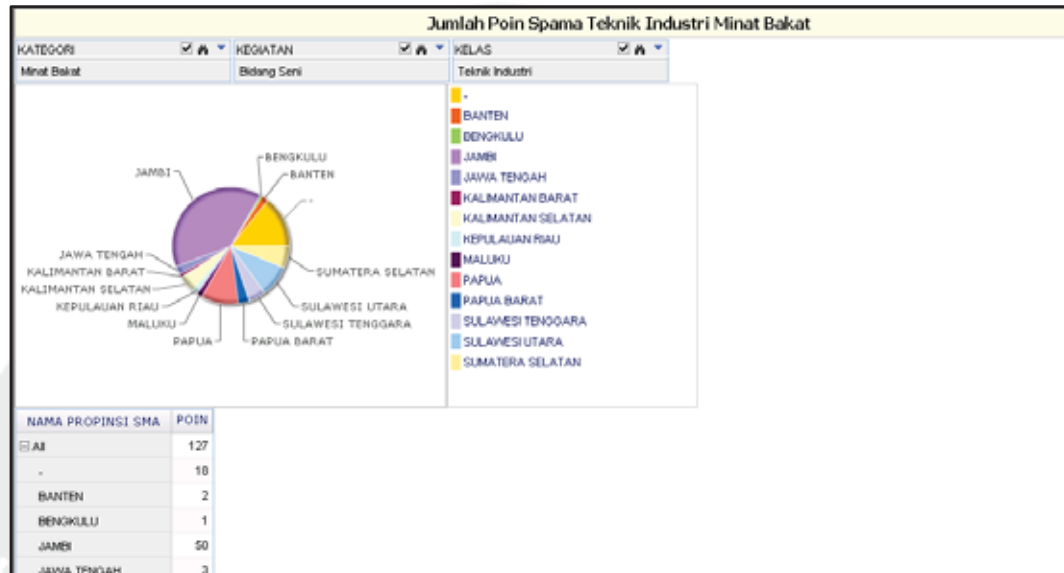
#### 5.3.1. Membuat OLAP Report

OLAP Report dapat dibuat dengan langkah-langkah sebagai berikut:

1. Mengakses Menu Design pada report portal dan memilih OLAP Report.
2. Memilih Connect to Cube untuk koneksi ke cube yang ada pada analysis server. Database pada Analysis server yang digunakan adalah spama (nama proyek SIKMA), beasiswa (nama proyek BEASISWA), dan alumni (nama proyek ALUMNI).
3. Tarik dan letakkan measure dan field yang disesuaikan dengan kebutuhan report serta menentukan field filter sesuai dengan kebutuhan.
4. OLAP Report dapat disajikan menggunakan grafik dengan mengklik menu grafik dan memilih jenis grafik yang diinginkan.

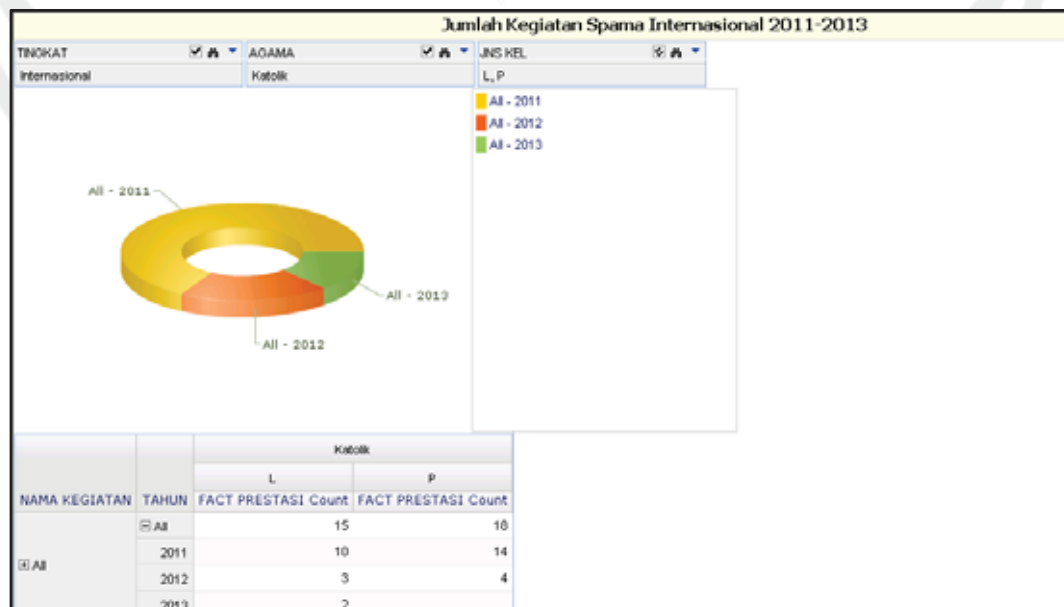
Berikut ini merupakan beberapa contoh report serta grafik berdasarkan analisis yang telah dilakukan:

1. Jumlah poin spama untuk kegiatan minat bakat dan kategori di bidang seni berdasarkan daerah asal sma pada prodi teknik industri



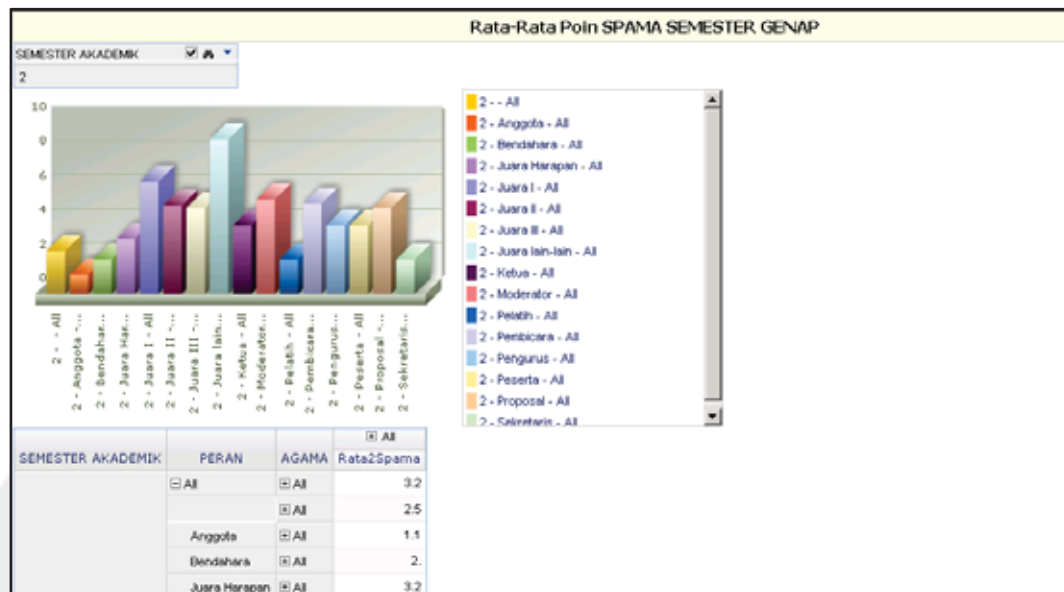
**Gambar 5.212. Report Jumlah Poin Spama**

2. Jumlah kegiatan spama antara mahasiswa laki-laki yang beragama katolik dan mahasiswa perempuan yang beragama katolik untuk kegiatan yang bertaraf internasional berdasarkan nama kegiatan dan tahun pelaksanaan



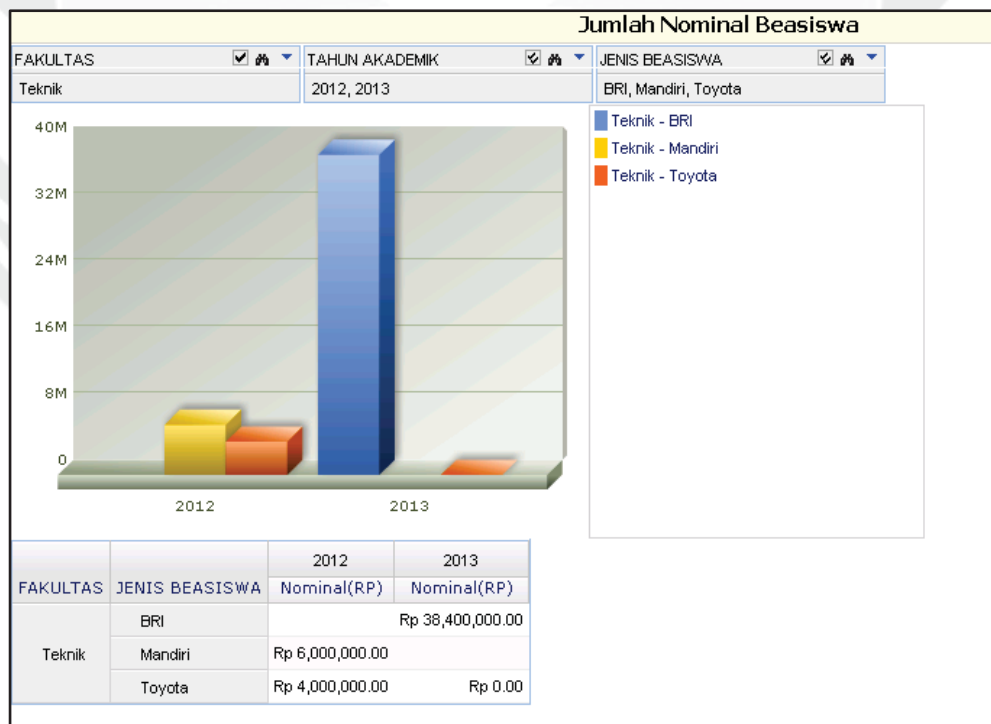
**Gambar 5.213. Report Jumlah Kegiatan Spama**

3. Rata-rata poin spama pada semester akademik genap berdasarkan peran kegiatan, agama mahasiswa, dan daerah asli mahasiswa



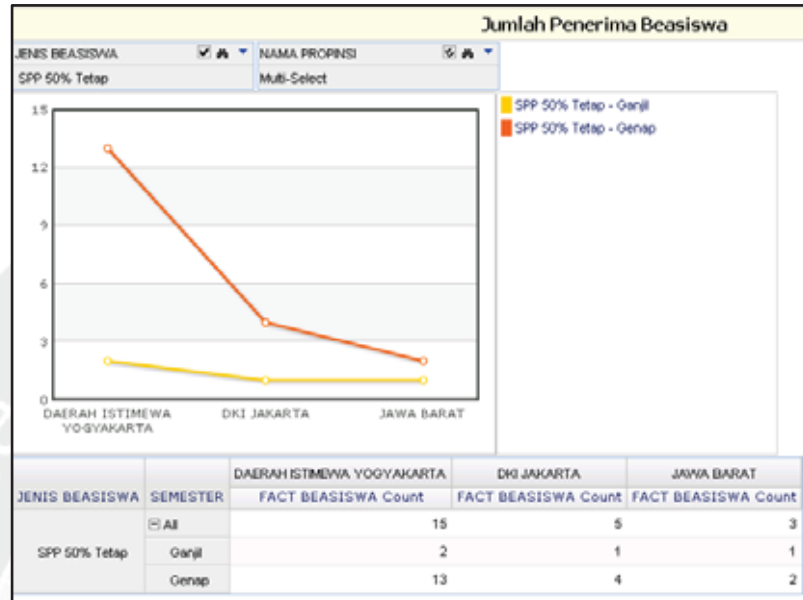
**Gambar 5.214. Report Rata-Rata Poin Spama**

4. Jumlah nominal beasiswa yang diterima mahasiswa fakultas teknik berdasarkan jenis beasiswa untuk tahun 2012 sampai tahun 2013



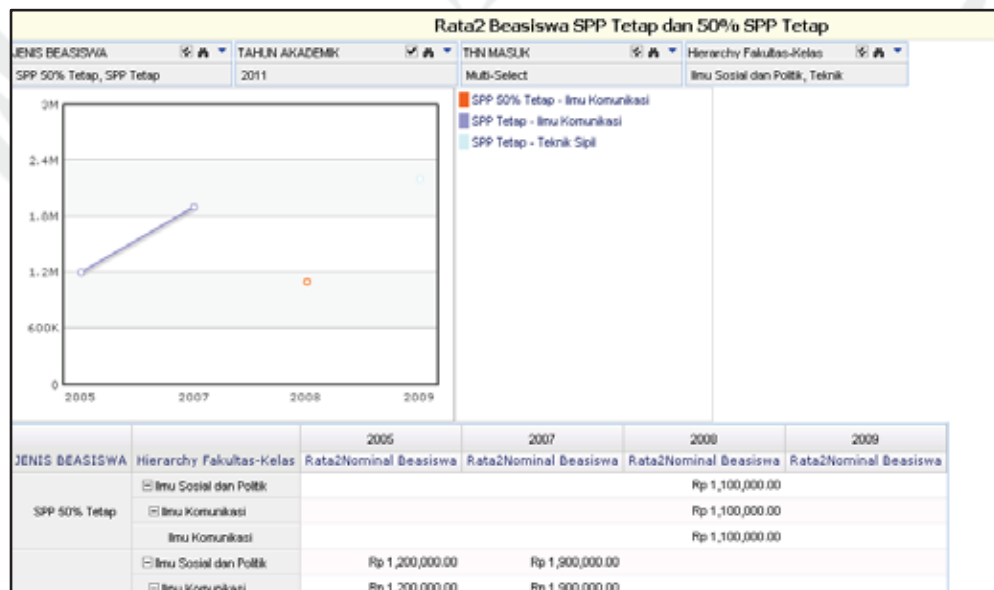
**Gambar 5.215. Report Jumlah Nominal Beasiswa**

5. Jumlah penerima beasiswa 50% spp tetap untuk semua prodi di setiap semester untuk mahasiswa yang berasal dari Bandung, Jakarta, dan Yogyakarta



**Gambar 5.216. Report Jumlah Penerima Beasiswa**

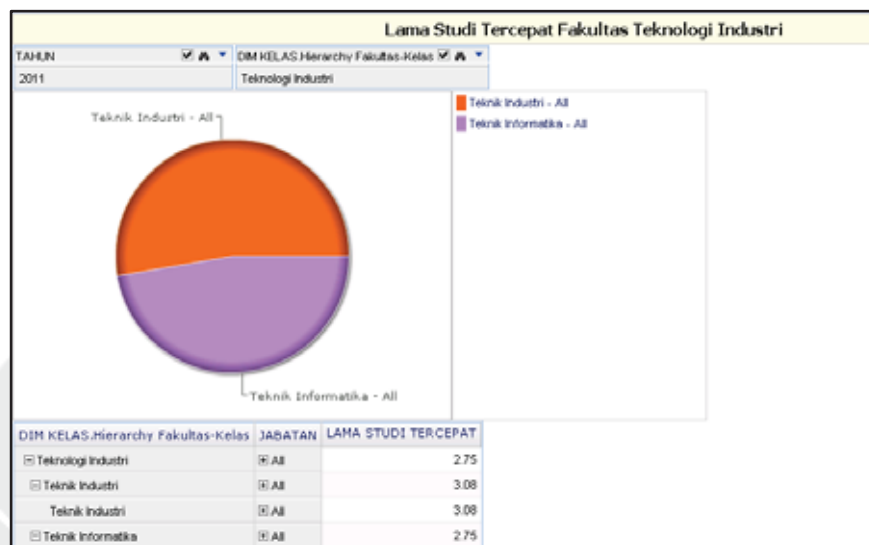
6. Rata-rata beasiswa tahun 2011 untuk beasiswa 50% spp tetap dan spp tetap untuk fakultas ilmu sosial dan politik dan teknik yang tahun masuk mahasiswanya adalah tahun 2005, 2007, dan 2008



**Gambar 5.217. Report Rata-Rata Beasiswa**

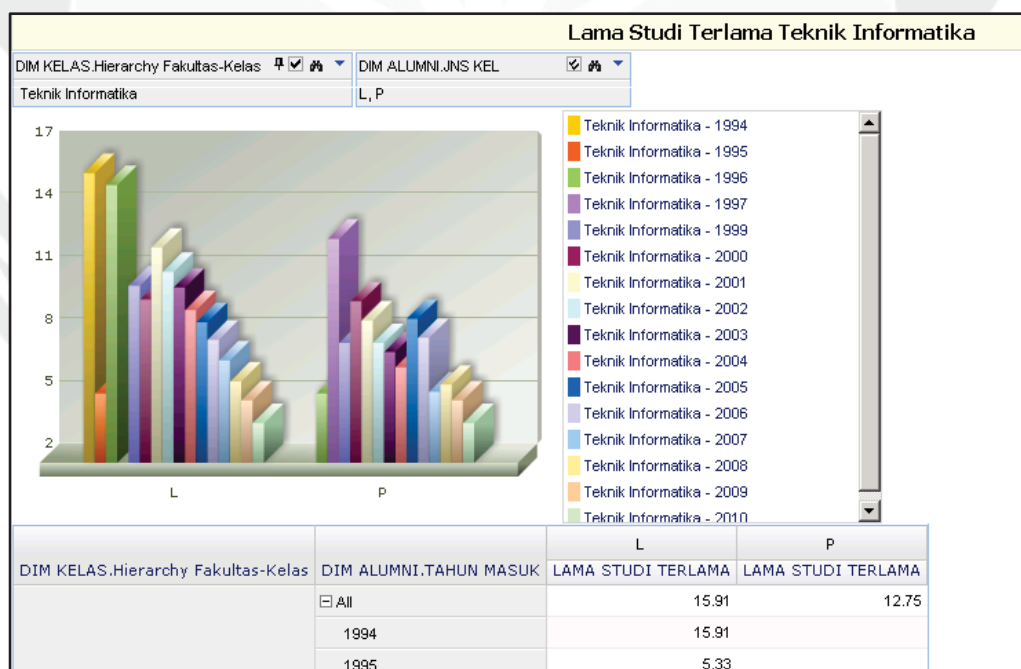


7. Lama studi tercepat pada fakultas teknologi industri pada tahun 2011 berdasarkan jabatan kerja alumni



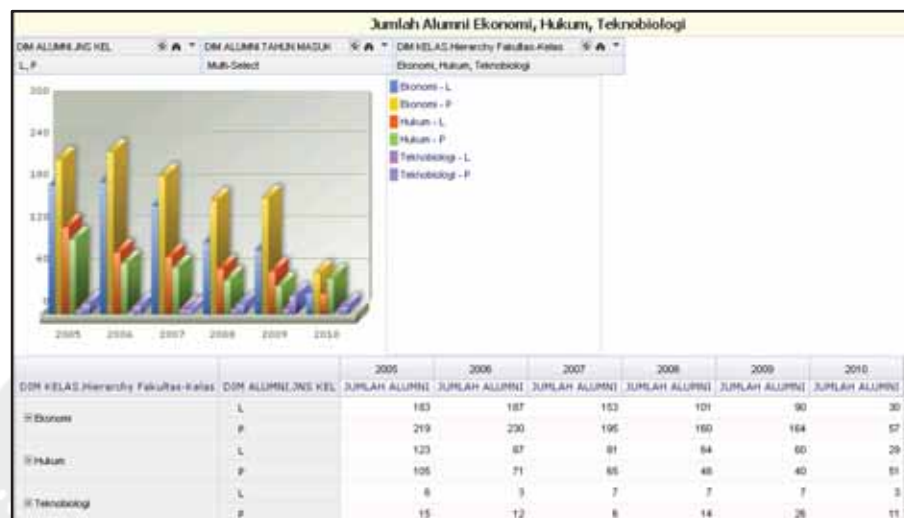
Gambar 5.218. Report Lama Studi Tercepat

8. Lama studi terlama pada prodi teknik informatika berdasarkan tahun masuk antara alumni laki-laki dan alumni perempuan



Gambar 5.219. Report Lama Studi Terlama

9. Jumlah alumni berdasarkan fakultas ekonomi, hukum, dan teknobiologi, serta jenis kelamin alumni, dan tahun masuk kuliah alumni pada tahun 2005-2014



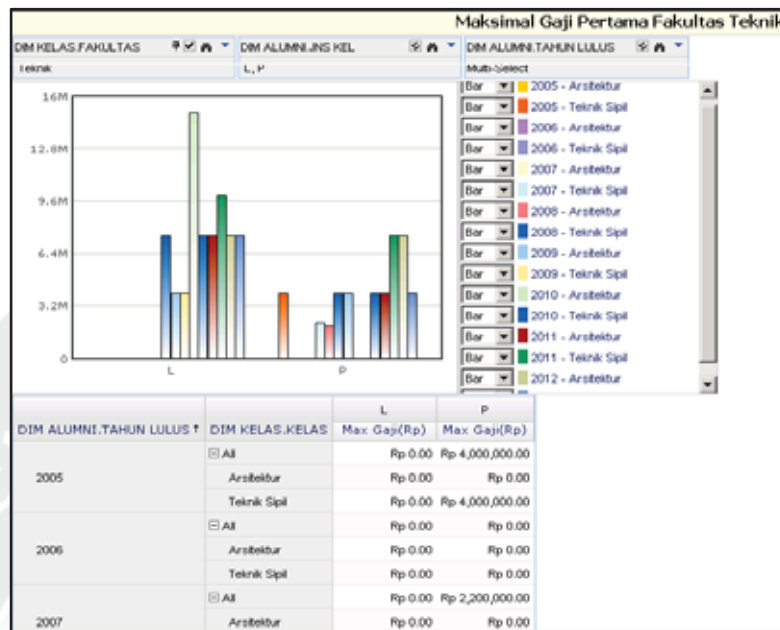
**Gambar 5.220. Report Jumlah Alumni**

10. Lama pembuatan tugas akhir yang terlama untuk semua fakultas yang tahun masuk alumni pada tahun 2007 berdasarkan jenis kelamin alumni



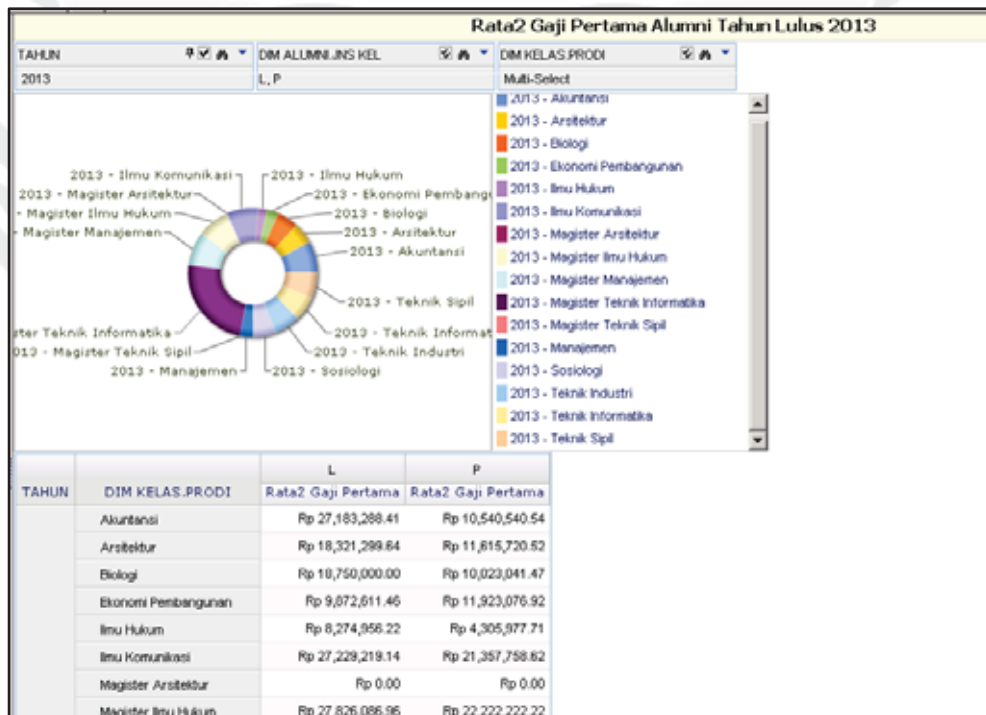
**Gambar 5.221. Report Lama Pembuatan Tugas Akhir**

11. Maksimal gaji pertama alumni pada fakultas teknik berdasarkan tahun lulus alumni, agama alumni, dan jenis kelamin alumni



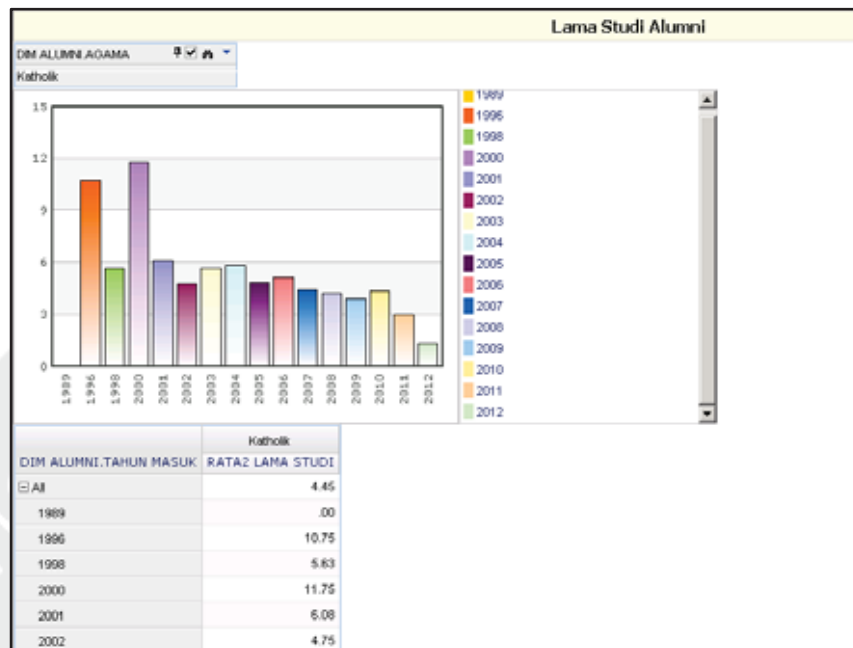
Gambar 5.222. Report Maksimal Gaji Pertama

12. Rata-rata gaji pertama alumni yang tahun lulusnya adalah tahun 2013 untuk semua prodi berdasarkan jenis kelamin



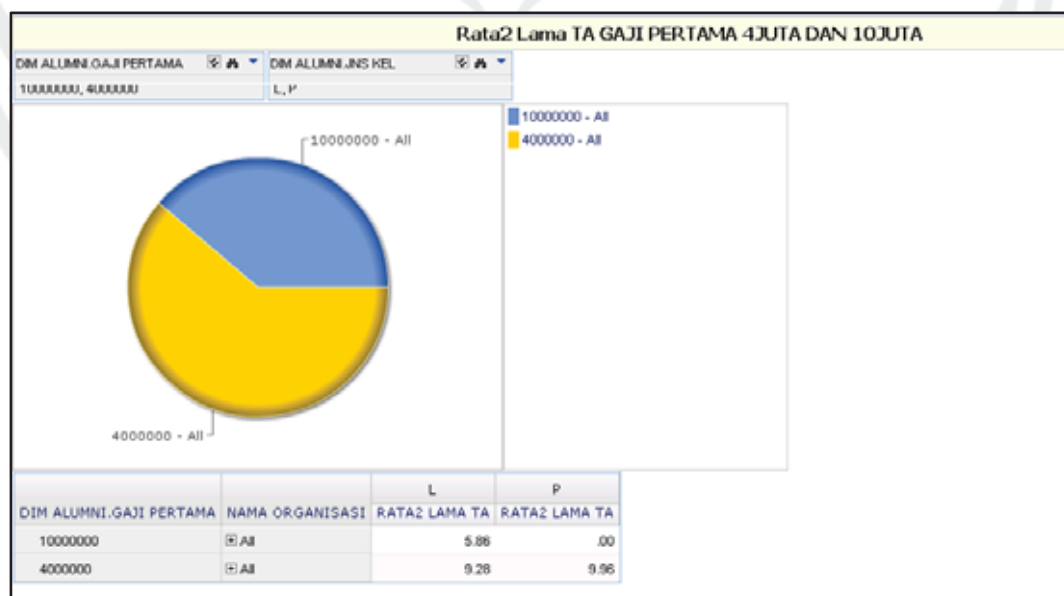
Gambar 5.223. Report Rata-Rata Gaji Pertama

13. Lama studi alumni berdasarkan tahun masuk alumni, agama, dan alumni yang daerah asalnya adalah Yogyakarta



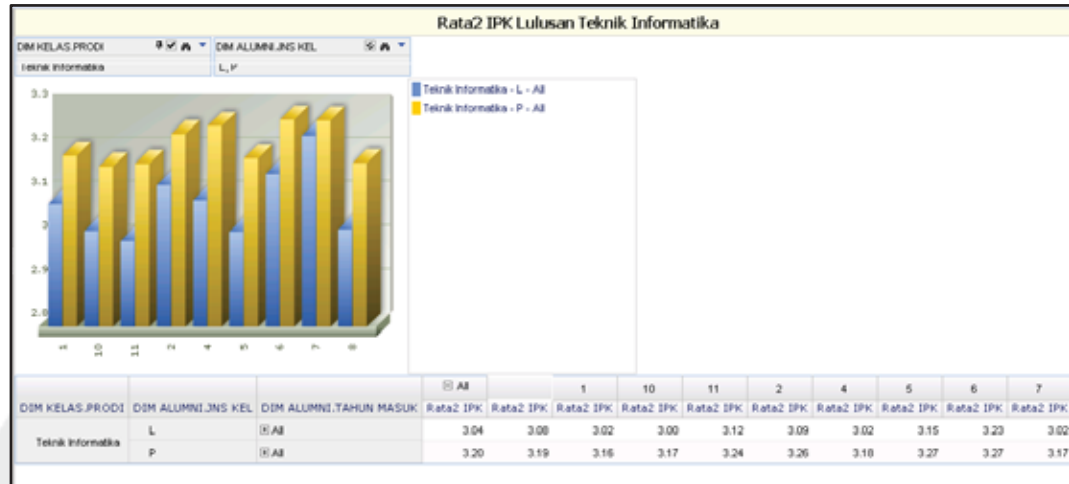
**Gambar 5.224. Report Lama Studi Alumni**

14. Rata-rata lama pembuatan tugas akhir alumni yang mempunyai gaji Rp 4.000.000,00 dan Rp 10.000.000,00 berdasarkan jenis kelamin alumni dan jabatan kerja alumni



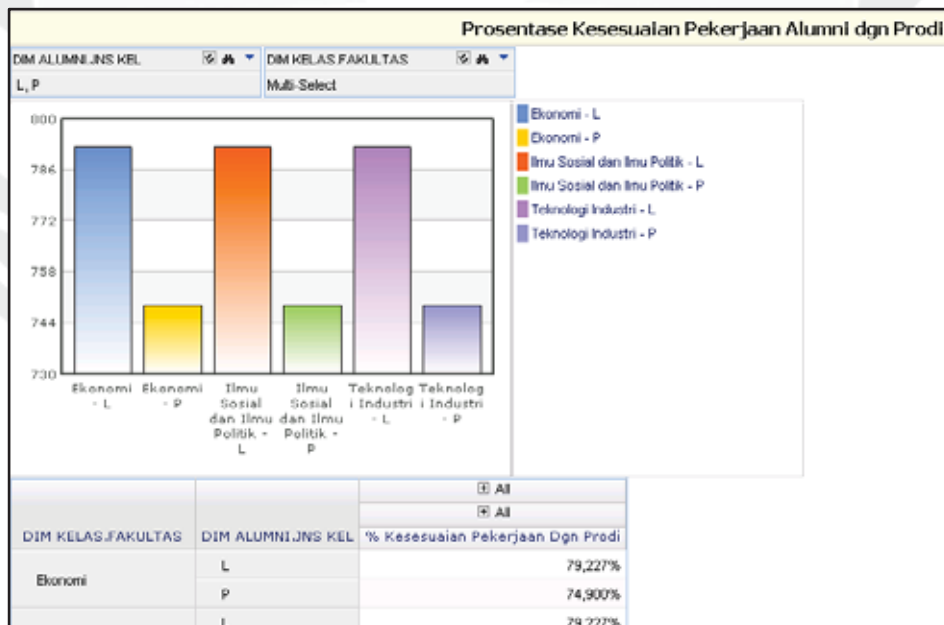
**Gambar 5.225. Report Rata-Rata Lama Pembuatan Tugas Akhir**

15. Rata-rata IPK lulusan fakultas teknologi industri yang mempunyai program studi teknik informatika berdasarkan jenis kelamin, bulan lulus, dan tahun masuk alumni



**Gambar 5.226. Report Rata-Rata IPK Lulusan**

16. Prosentase kesesuaian pekerjaan alumni dan program studi untuk fakultas ekonomi, ilmu sosial dan politik, dan teknologi industri berdasarkan jenis kelamin, asal provinsi alumni, dan asal kabupaten kodya alumni



**Gambar 5.227. Report Prosentase Kesesuaian Pekerjaan Alumni Dengan Prodi**



## 17. Penyebaran poin spama berdasarkan nama provinsi



**Gambar 5.228. Report Penyebaran Poin Spama**

### 5.3.2. Membuat KPI Report

Key Performance Indicator (KPI) merupakan alat yang dapat membantu perusahaan, organisasi, instansi, dan asosiasi dalam mengukur kemajuan dalam mencapai tujuan. KPI digunakan untuk mengukur tingkat keberhasilan kinerja perusahaan saat ini dan juga membantu organisasi mencapai tujuan organisasi dalam jangka panjang. KPI *report* dapat dibuat dengan mengklik menu **Design** pada report portal dan pilih **KPI Report**. Berikut ini adalah cara membuat KPI Report untuk jumlah beasiswa, rata-rata poin spama, dan rata-rata gaji pertama:

#### 1. KPI Report jumlah beasiswa mahasiswa fakultas teknik pada tahun 2013

|                                                |                                       |
|------------------------------------------------|---------------------------------------|
| KPI Name                                       | JUMLAHBEASISWA                        |
| Current Value Report                           | JUMLAHNOMINALBEASISWA - Row: 1, Color |
| Previous Value Report (optional)               | JUMLAHNOMINALBEASISWA - Row: 1, Color |
| Goal ( <input type="checkbox"/> Report Based ) | 50000000.00                           |
| Green Threshold (1%-140%) ?                    | 80 %                                  |
| Red Threshold (1%-140%) ?                      | 60 %                                  |
| Units (% , kilos , liters , pounds)            |                                       |
| Comments                                       |                                       |
| Icon Type                                      | Traffic Light                         |
| Refresh Rate                                   | No refreshing                         |

**Gambar 5.229. Membuat KPI Report Jumlah Beasiswa**

Nilai sekarang diambil dari Report Jumlah Nominal Beasiswa baris 1 kolom 1 yang merupakan jumlah nominal beasiswa mahasiswa fakultas teknik pada tahun 2013. Nilai sebelumnya diambil dari Report Jumlah Nominal Beasiswa baris 1 kolom 2 yang merupakan jumlah nominal beasiswa mahasiswa fakultas teknik pada tahun 2012. Red Threshold dibuat 60% dengan tujuan agar yang berada di area kuning adalah Rp 2.000.000,00 sampai Rp 3.000.000,00, sedangkan yang berada di area hijau adalah yang lebih dari *goal* (lebih dari Rp 3.000.000,00).



**Gambar 5.230. KPI Report Jumlah Beasiswa**

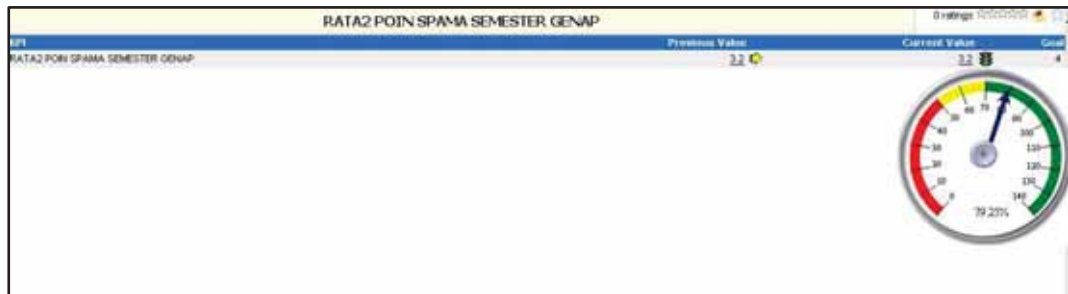
## 2. KPI Report rata-rata poin spama semester genap untuk semua fakultas

|                                                |                                     |   |
|------------------------------------------------|-------------------------------------|---|
| KPI Name                                       | RATA2 POIN SPAMA SEMESTER GENAP     |   |
| Current Value Report                           | Rata-Rata Poin SPAMA SEMESTER GENAP | + |
| Previous Value Report (optional)               | Rata-Rata Poin SPAMA SEMESTER GENAP | + |
| Goal ( <input type="checkbox"/> Report Based ) | 4.00                                |   |
| Green Threshold (1%-140%) ?                    | 70 %                                |   |
| Red Threshold (1%-140%) ?                      | 50 %                                |   |
| Units (% , kilos , liters , pounds)            |                                     |   |
| Comments                                       |                                     |   |
| Icon Type                                      | Traffic Light                       |   |
| Refresh Rate                                   | No refreshing                       |   |

**Gambar 5.231. Membuat KPI Report Rata-Rata Poin Spama**

Nilai sekarang diambil dari Report Rata-Rata Poin SPAMA baris 1 kolom 1 yang merupakan rata-rata poin spama semua mahasiswa pada semester genap tahun 2013. Nilai sebelumnya diambil dari Report Rata-Rata Poin SPAMA baris 1 kolom 2 yang merupakan rata-rata poin spama semua mahasiswa pada

semester genap tahun 2012. Red Threshold dibuat 50% dengan tujuan agar yang berada di area kuning adalah 2 sampai 2,8, sedangkan yang berada di area hijau adalah yang lebih dari *goal* (lebih dari 2,8).



Gambar 5.232. KPI Report Rata-Rata Poin Spama

### 3. Rata-Rata Gaji Pertama Semua Prodi Tahun 2012 dan 2013

|                                                |                                            |   |
|------------------------------------------------|--------------------------------------------|---|
| KPI Name                                       | RATA2 GAJI PERTAMA SEMUA PRODI 2012        |   |
| Current Value Report                           | Rata2 Gaji Pertama Semua Prodi 2012 2013 - |   |
| Previous Value Report (optional)               | Rata2 Gaji Pertama Semua Prodi 2012 2013 - |   |
| Goal ( <input type="checkbox"/> Report Based ) | 17500000.00                                |   |
| Green Threshold (1%-140%) ?                    | 70                                         | % |
| Red Threshold (1%-140%) ?                      | 60                                         | % |
| Units (% , kilos, liters, pounds)              |                                            |   |
| Comments                                       |                                            |   |
| Icon Type                                      | Traffic Light                              |   |
| Refresh Rate                                   | No refreshing                              |   |

Gambar 5.233. Membuat KPI Report Rata-Rata Gaji Pertama

Nilai sekarang diambil dari Report Rata-Rata Gaji Pertama baris 1 kolom 1,2 yang merupakan rata-rata gaji pertama semua mahasiswa tahun 2013. Nilai sebelumnya diambil dari Report Rata-Rata Gaji Pertama baris 1 kolom 3,4 yang merupakan rata-rata gaji pertama semua mahasiswa tahun 2012. Red Threshold dibuat 60% dengan tujuan agar yang berada di area kuning adalah Rp 10.500.000,00 sampai Rp 12.250.000,00 sedangkan yang berada di area hijau adalah yang lebih dari *goal* (lebih dari Rp 12.250.000,00).



**Gambar 5.234. KPI Report Rata-Rata Gaji Pertama**

#### **5.4. Hasil Pengujian**

Pada bagian ini akan ditampilkan hasil pengujian setiap fungsi dari system yang telah dikembangkan. Hasil pengujian dapat dilihat pada tabel 5.1.

Tabel 5.1. Tabel Hasil Pengujian

| Identifikasi | Deskripsi                                                        | Prosedur Pengujian                                                        | Masukan                                                                                   | Keluaran yang diharapkan                                   | Kriteria evaluasi hasil                                    | Hasil yang didapat                                         | Kesimpulan |
|--------------|------------------------------------------------------------------|---------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------|------------|
| P-01-01      | Pengujian terhadap antarmuka jumlah lulusan                      | Memilih jumlah lulusan sebagai data yang ditampilkan                      | -Klik jumlah lulusan dan pilih dimensi-dimensi yang akan ditampilkan                      | -Data jumlah lulusan akan ditampilkan                      | -Data jumlah lulusan akan ditampilkan                      | -Data jumlah lulusan akan ditampilkan                      | Handal     |
| P-01-02      | Pengujian terhadap antarmuka lama studi tercepat                 | Memilih lama studi tercepat sebagai data yang ditampilkan                 | -Klik lama studi tercepat dan pilih dimensi-dimensi yang akan ditampilkan                 | -Data lama studi tercepat akan ditampilkan                 | -Data lama studi tercepat akan ditampilkan                 | -Data lama studi tercepat akan ditampilkan                 | Handal     |
| P-01-03      | Pengujian terhadap antarmuka lama studi terlama                  | Memilih lama studi terlama sebagai data yang ditampilkan                  | -Klik lama studi terlama dan pilih dimensi-dimensi yang akan ditampilkan                  | -Data lama studi terlama akan ditampilkan                  | -Data lama studi terlama akan ditampilkan                  | -Data lama studi terlama akan ditampilkan                  | Handal     |
| P-01-04      | Pengujian terhadap antarmuka lama pembuatan tugas akhir tercepat | Memilih lama pembuatan tugas akhir tercepat sebagai data yang ditampilkan | -Klik lama pembuatan tugas akhir tercepat dan pilih dimensi-dimensi yang akan ditampilkan | -Data lama pembuatan tugas akhir tercepat akan ditampilkan | -Data lama pembuatan tugas akhir tercepat akan ditampilkan | -Data lama pembuatan tugas akhir tercepat akan ditampilkan | Handal     |

| <b>Identifik asi</b> | <b>Deskripsi</b>                                                | <b>Prosedur Pengujian</b>                                                | <b>Masukan</b>                                                                           | <b>Keluaran yang diharapkan</b>                           | <b>Kriteria evaluasi hasil</b>                            | <b>Hasil yang didapat</b>                                 | <b>Kesimpulan</b> |
|----------------------|-----------------------------------------------------------------|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|-------------------|
| P-01-05              | Pengujian terhadap antarmuka lama pembuatan tugas akhir terlama | Memilih lama pembuatan tugas akhir terlama sebagai data yang ditampilkan | -Klik lama pembuatan tugas akhir terlama dan pilih dimensi-dimensi yang akan ditampilkan | -Data lama pembuatan tugas akhir terlama akan ditampilkan | -Data lama pembuatan tugas akhir terlama akan ditampilkan | -Data lama pembuatan tugas akhir terlama akan ditampilkan | Handal            |
| P-01-06              | Pengujian terhadap antarmuka minimal IPK lulusan                | Memilih minimal IPK lulusan sebagai data yang ditampilkan                | -Klik minimal IPK lulusan dan pilih dimensi-dimensi yang akan ditampilkan                | -Data minimal IPK lulusan akan ditampilkan                | -Data minimal IPK lulusan akan ditampilkan                | -Data minimal IPK lulusan akan ditampilkan                | Handal            |
| P-01-07              | Pengujian terhadap antarmuka maksimal IPK lulusan               | Memilih maksimal IPK lulusan sebagai data yang ditampilkan               | -Klik maksimal IPK lulusan dan pilih dimensi-dimensi yang akan ditampilkan               | -Data maksimal IPK lulusan akan ditampilkan               | -Data maksimal IPK lulusan akan ditampilkan               | -Data maksimal IPK lulusan akan ditampilkan               | Handal            |
| P-01-08              | Pengujian terhadap antarmuka rata-rata IPK lulusan              | Memilih rata-rata IPK lulusan sebagai data yang ditampilkan              | -Klik rata-rata IPK lulusan dan pilih dimensi yang akan ditampilkan                      | -Data rata-rata IPK lulusan akan ditampilkan              | -Data rata-rata IPK lulusan akan ditampilkan              | -Data rata-rata IPK lulusan akan ditampilkan              | Handal            |



| <b>Identifik asi</b> | <b>Deskripsi</b>                                    | <b>Prosedur Pengujian</b>                                    | <b>Masukan</b>                                             | <b>Keluaran yang diharapkan</b>                            | <b>Kriteria evaluasi hasil</b>                             | <b>Hasil yang didapat</b>                                  | <b>Kesimpulan</b> |
|----------------------|-----------------------------------------------------|--------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------|-------------------|
| P-02-01              | Pengujian terhadap antarmuka maksimal gaji pertama  | Memilih maksimal gaji pertama sebagai data yang ditampilkan  | -Klik maksimal gaji pertama sebagai data yang ditampilkan  | -Data maksimal gaji pertama sebagai data yang ditampilkan  | -Data maksimal gaji pertama sebagai data yang ditampilkan  | -Data maksimal gaji pertama sebagai data yang ditampilkan  | Handal            |
| P-02-02              | Pengujian terhadap antarmuka minimal gaji pertama   | Memilih minimal gaji pertama sebagai data yang ditampilkan   | -Klik minimal gaji pertama sebagai data yang ditampilkan   | -Data minimal gaji pertama sebagai data yang ditampilkan   | -Data minimal gaji pertama sebagai data yang ditampilkan   | -Data minimal gaji pertama sebagai data yang ditampilkan   | Handal            |
| P-02-03              | Pengujian terhadap antarmuka rata-rata gaji pertama | Memilih rata-rata gaji pertama sebagai data yang ditampilkan | -Klik rata-rata gaji pertama sebagai data yang ditampilkan | -Data rata-rata gaji pertama sebagai data yang ditampilkan | -Data rata-rata gaji pertama sebagai data yang ditampilkan | -Data rata-rata gaji pertama sebagai data yang ditampilkan | Handal            |

| Identifikasi | Deskripsi                                                                      | Prosedur Pengujian                                                                      | Masukan                                                                               | Keluaran yang diharapkan                                                              | Kriteria evaluasi hasil                                                               | Hasil yang didapat                                                                    | Kesimpulan |
|--------------|--------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|------------|
| P-02-04      | Pengujian terhadap antarmuka prosentase kesesuaian pekerjaan dengan prodi      | Memilih prosentase kesesuaian pekerjaan dengan prodi sebagai data yang ditampilkan      | -Klik prosentase kesesuaian pekerjaan dengan prodi sebagai data yang ditampilkan      | -Data prosentase kesesuaian pekerjaan dengan prodi sebagai data yang ditampilkan      | -Data prosentase kesesuaian pekerjaan dengan prodi sebagai data yang ditampilkan      | -Data prosentase kesesuaian pekerjaan dengan prodi sebagai data yang ditampilkan      | Handal     |
| P-02-05      | Pengujian terhadap antarmuka waktu tunggu lulusan memperoleh pekerjaan pertama | Memilih waktu tunggu lulusan memperoleh pekerjaan pertama sebagai data yang ditampilkan | -Klik waktu tunggu lulusan memperoleh pekerjaan pertama sebagai data yang ditampilkan | -Data waktu tunggu lulusan memperoleh pekerjaan pertama sebagai data yang ditampilkan | -Data waktu tunggu lulusan memperoleh pekerjaan pertama sebagai data yang ditampilkan | -Data waktu tunggu lulusan memperoleh pekerjaan pertama sebagai data yang ditampilkan | Handal     |

| <b>Identifik asi</b> | <b>Deskripsi</b>                              | <b>Prosedur Pengujian</b>                                      | <b>Masukan</b>                                               | <b>Keluaran yang diharapkan</b>                              | <b>Kriteria evaluasi hasil</b>                               | <b>Hasil yang didapat</b>                                    | <b>Kesimpulan</b> |
|----------------------|-----------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|-------------------|
| P-03-01              | Pengujian terhadap jumlah nominal beasiswa    | Memilih jumlah nominal beasiswa sebagai data yang ditampilkan  | -Klik jumlah nominal beasiswa sebagai data yang ditampilkan  | -Data jumlah nominal beasiswa sebagai data yang ditampilkan  | -Data jumlah nominal beasiswa sebagai data yang ditampilkan  | -Data jumlah nominal beasiswa sebagai data yang ditampilkan  | Handal            |
| P-03-02              | Pengujian terhadap rata-rata nominal beasiswa | Memilih rata-rata nominal sebagai data yang ditampilkan        | -Klik rata-rata nominal sebagai data yang ditampilkan        | -Data rata-rata nominal sebagai data yang ditampilkan        | -Data rata-rata nominal sebagai data yang ditampilkan        | -Data rata-rata nominal sebagai data yang ditampilkan        | Handal            |
| P-03-03              | Pengujian terhadap jumlah penerima beasiswa   | Memilih jumlah penerima beasiswa sebagai data yang ditampilkan | -Klik jumlah penerima beasiswa sebagai data yang ditampilkan | -Data jumlah penerima beasiswa sebagai data yang ditampilkan | -Data jumlah penerima beasiswa sebagai data yang ditampilkan | -Data jumlah penerima beasiswa sebagai data yang ditampilkan | Handal            |

| <b>Identifik asi</b> | <b>Deskripsi</b>                                         | <b>Prosedur Pengujian</b>                                                   | <b>Masukan</b>                                                            | <b>Keluaran yang diharapkan</b>                                           | <b>Kriteria evaluasi hasil</b>                                            | <b>Hasil yang didapat</b>                                                 | <b>Kesimpulan</b> |
|----------------------|----------------------------------------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|-------------------|
| P-04-01              | Pengujian terhadap jumlah poin spama                     | Memilih jumlah poin spama sebagai data yang ditampilkan                     | -Klik jumlah poin spama sebagai data yang ditampilkan                     | -Data jumlah poin spama sebagai data yang ditampilkan                     | -Data jumlah poin spama sebagai data yang ditampilkan                     | -Data jumlah poin spama sebagai data yang ditampilkan                     | Handal            |
| P-04-02              | Pengujian terhadap rata-rata poin spama                  | Memilih rata-rata poin spama sebagai data yang ditampilkan                  | -Klik rata-rata poin spama sebagai data yang ditampilkan                  | -Data rata-rata poin spama sebagai data yang ditampilkan                  | -Data rata-rata poin spama sebagai data yang ditampilkan                  | -Data rata-rata poin spama sebagai data yang ditampilkan                  | Handal            |
| P-04-03              | Pengujian terhadap jumlah mahasiswa pengumpul poin spama | Memilih jumlah mahasiswa pengumpul poin spama sebagai data yang ditampilkan | -Klik jumlah mahasiswa pengumpul poin spama sebagai data yang ditampilkan | -Data jumlah mahasiswa pengumpul poin spama sebagai data yang ditampilkan | -Data jumlah mahasiswa pengumpul poin spama sebagai data yang ditampilkan | -Data jumlah mahasiswa pengumpul poin spama sebagai data yang ditampilkan | Handal            |

### 5.5. Pengujian Terhadap Pengguna

Pengujian terhadap system yang akan dikembangkan ini dilakukan terhadap 4 responden, yang terdiri dari 2 orang wakil dekan III, 1 orang kepala KKACM, dan 1 orang kepala KSI. Responden diminta untuk mencoba Sistem Kemanusiaan dan Alumni (SIKMA) ini, kemudian mengisi kuesioner secara langsung. Hasil pengujian dapat dilihat pada tabel 5.2

**Tabel 5.2 Hasil Uji Terhadap Pengguna**

| No | Pernyataan                                                                           | SS | S | N | TS | STS |
|----|--------------------------------------------------------------------------------------|----|---|---|----|-----|
| 1  | Tampilan sistem menarik (Learnability)                                               |    |   |   |    |     |
| 2  | Sistem mudah digunakan (Learnability)                                                |    |   |   |    |     |
| 3  | Sistem mengurangi waktu dalam mencari data yang diperlukan (Learnability)            |    |   |   |    |     |
| 4  | Sistem mudah dipelajari (Learnability)                                               |    |   |   |    |     |
| 5  | Sistem mempermudah proses mencari data spama, beasiswa, maupun alumni (Flexibility)  |    |   |   |    |     |
| 6  | Sistem menawarkan banyak cara dalam menampilkan data yang dicari (Flexibility)       |    |   |   |    |     |
| 7  | Sistem mempermudah proses pengambilan keputusan (Effectiveness)                      |    |   |   |    |     |
| 8  | Sistem menampilkan data yang diinginkan secara cepat (Effectiveness)                 |    |   |   |    |     |
| 9  | Sistem mempermudah memonitoring aktivitas spama, beasiswa, maupun alumni (Usability) |    |   |   |    |     |
| 10 | Sistem membantu dalam mendapatkan data spama, beasiswa, maupun alumni (Usability)    |    |   |   |    |     |

Keterangan:

SS = Sangat Setuju

S = Setuju

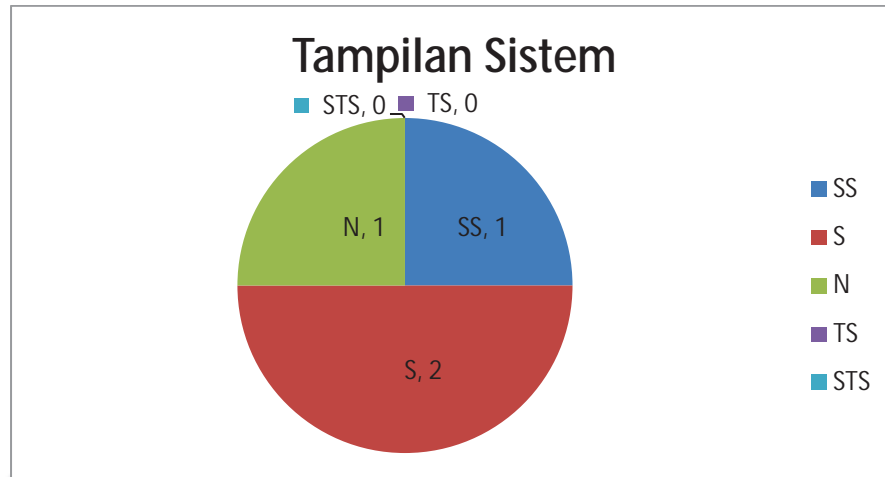
N = Netral

ST = Sangat Setuju

STS = Sangat Tidak Setuju

#### 5.5.1. Pengujian Tampilan Sistem

Pengujian ini dilakukan untuk membuktikan bahwa tampilan sistem ini menarik baik dari segi warna maupun segi desain halaman. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 1 responden sangat setuju, 2 responden setuju, 1 responden netral, 0 responden tidak setuju, dan 0 responden sangat tidak setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.235.

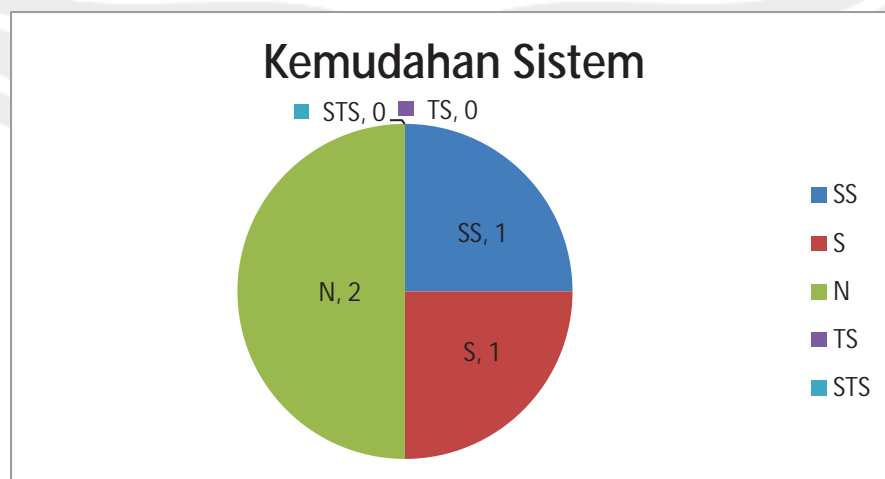


**Gambar 5.235. Pengujian Tampilan Sistem**

Dari gambar 5.235. Dapat disimpulkan bahwa tampilan sistem menarik sehingga pengguna tertarik untuk menggunakan sistem. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 25%, setuju 25%, netral 50%, tidak setuju 0%, dan sangat tidak setuju 0%.

#### 5.5.2. Pengujian Kemudahan Penggunaan Sistem

Pengujian ini dilakukan untuk membuktikan bahwa sistem mudah digunakan dari segi data yang akan ditampilkan. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 1 responden sangat setuju, 1 responden setuju, 2 responden netral, 0 responden tidak setuju, dan 0 responden sangat tidak setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.236.



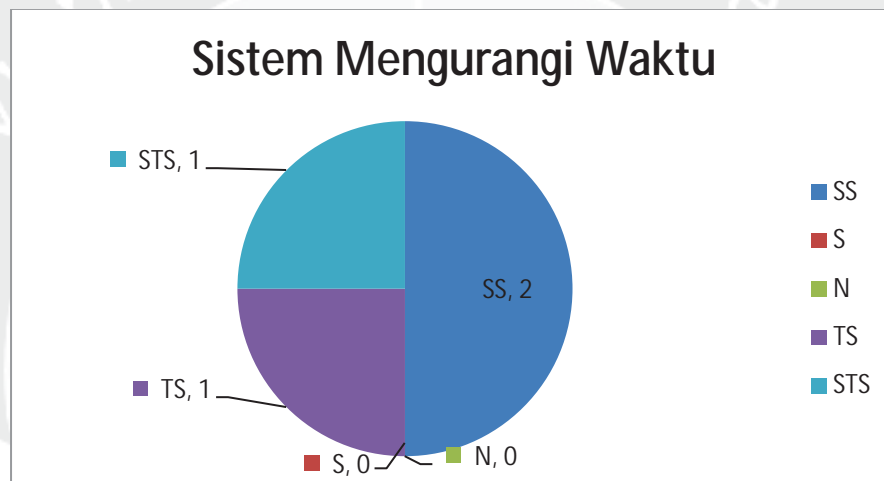
**Gambar 5.236. Pengujian Kemudahan Sistem**



Dari gambar 5.236. dapat disimpulkan bahwa sistem mudah digunakan. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 25%, setuju 25%, netral 50%, tidak setuju 0%, dan sangat tidak setuju 0%.

### 5.5.3. Pengujian Sistem Mengurangi Waktu Dalam Mencari Data

Pengujian ini dilakukan untuk membuktikan bahwa sistem dapat mengurangi waktu pengguna dalam mencari data. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 2 responden sangat setuju, 0 responden setuju, 0 responden netral, 1 responden tidak setuju, dan 1 responden sangat tidak setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.237.

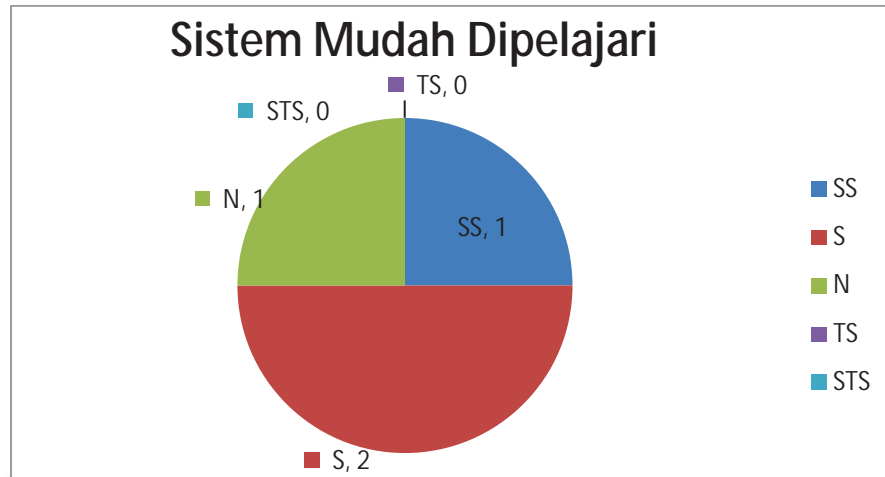


**Gambar 5.237. Pengujian Sistem Mengurangi Waktu**

Dari gambar 5.237. dapat disimpulkan bahwa sistem mengurangi waktu dalam mencari data. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 50%, setuju 0%, netral 0%, tidak setuju 25%, dan sangat tidak setuju 25%.

### 5.5.4. Pengujian Sistem Mudah Dipelajari

Pengujian ini dilakukan untuk membuktikan bahwa sistem ini mudah untuk dipelajari karena sistem ini sederhana. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 1 responden sangat setuju, 2 responden setuju, 1 responden netral, 0 responden tidak setuju, dan 0 responden sangat tidak setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.238.

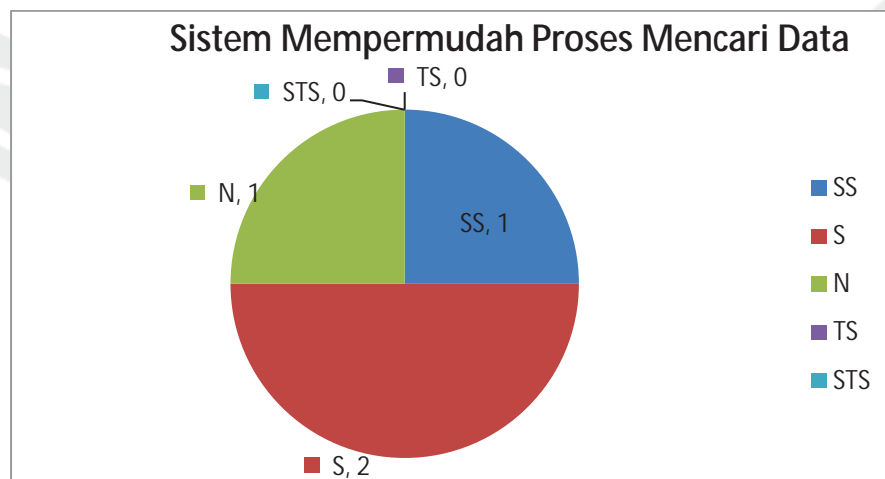


**Gambar 5.238. Pengujian Sistem Mudah Dipelajari**

Dari gambar 5.238. dapat disimpulkan bahwa sistem mudah untuk dipelajari. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 25%, setuju 50%, netral 25%, tidak setuju 0%, dan sangat tidak setuju 0%.

#### 5.5.5. Pengujian Sistem Mempermudah Proses Mencari Data

Pengujian ini dilakukan untuk membuktikan bahwa sistem ini mempermudah proses mencari data. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 1 responden sangat setuju, 2 responden setuju, 1 responden netral, 0 responden tidak setuju, dan 0 responden sangat tidak setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.239.

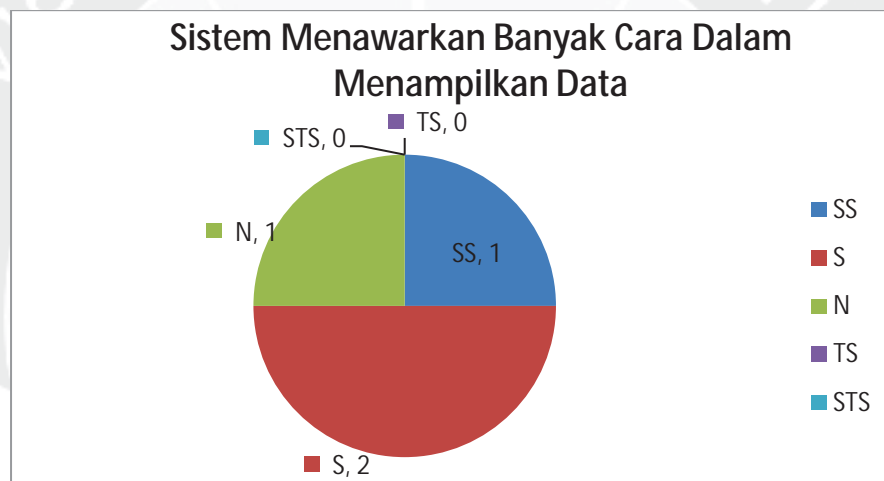


**Gambar 5.239. Pengujian Sistem Mempermudah Proses Mencari Data**

Dari gambar 5.239. dapat disimpulkan bahwa sistem mempermudah proses mencari data. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 25%, setuju 50%, netral 25%, tidak setuju 0%, dan sangat tidak setuju 0%.

#### 5.5.6. Pengujian Sistem Menawarkan Banyak Cara Dalam Menampilkan Data

Pengujian ini dilakukan untuk membuktikan bahwa sistem ini menawarkan banyak cara dalam menampilkan data. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 1 responden sangat setuju, 2 responden setuju, 1 responden netral, 0 responden tidak setuju, dan 0 responden sangat tidak setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.240.



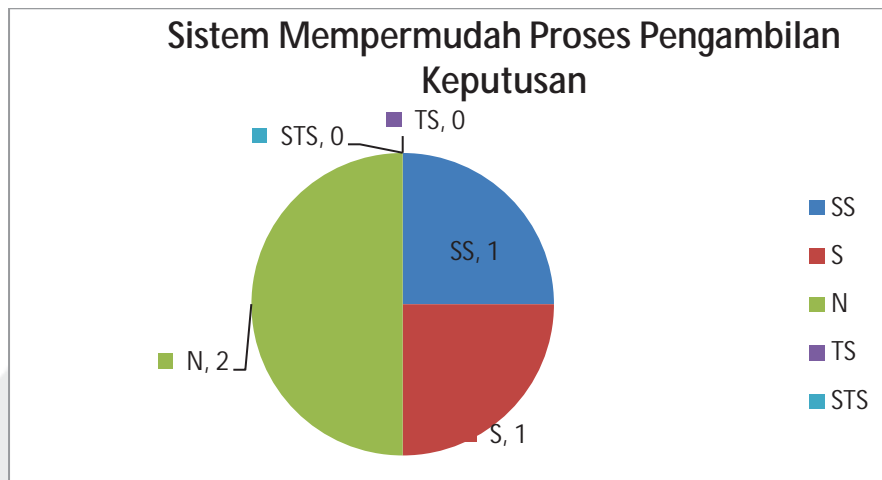
**Gambar 5.240. Pengujian Sistem Menawarkan Banyak Cara Dalam Menampilkan Data**

Dari gambar 5.240. dapat disimpulkan bahwa sistem menawarkan banyak cara dalam menampilkan data. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 25%, setuju 50%, netral 25%, tidak setuju 0%, dan sangat tidak setuju 0%.

#### 5.5.7. Pengujian Sistem Mempermudah Proses Pengambilan Keputusan

Pengujian ini dilakukan untuk membuktikan bahwa sistem ini mempermudah proses pengambilan keputusan. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 1 responden sangat setuju, 1 responden setuju, 2 responden netral, 0 responden tidak setuju, dan 0 responden sangat tidak

setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.241.

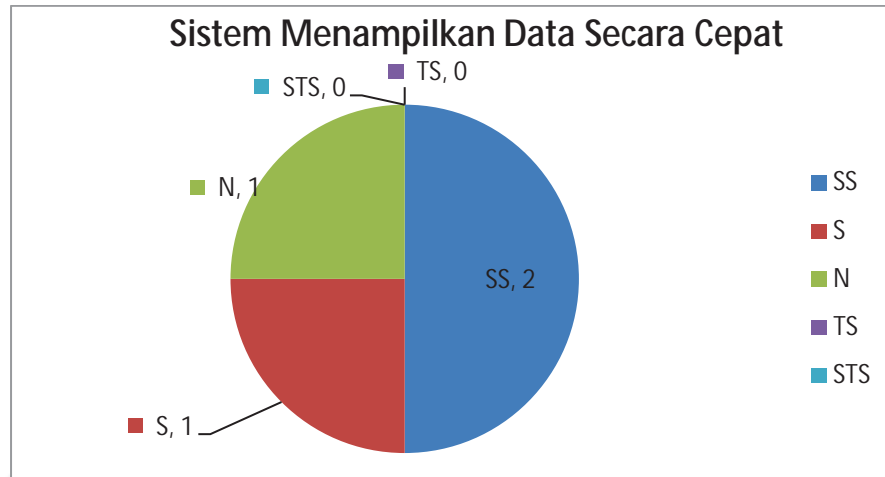


**Gambar 5.241. Pengujian Sistem Mempermudah Proses Pengambilan Keputusan**

Dari gambar 5.241. dapat disimpulkan bahwa sistem mempermudah pengumpulan keputusan. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 25%, setuju 25%, netral 50%, tidak setuju 0%, dan sangat tidak setuju 0%.

#### **5.5.8. Pengujian Sistem Menampilkan Data Secara Cepat**

Pengujian ini dilakukan untuk membuktikan bahwa sistem ini dapat menampilkan data yang diminta secara cepat. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 2 responden sangat setuju, 1 responden setuju, 1 responden netral, 0 responden tidak setuju, dan 0 responden sangat tidak setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.242.

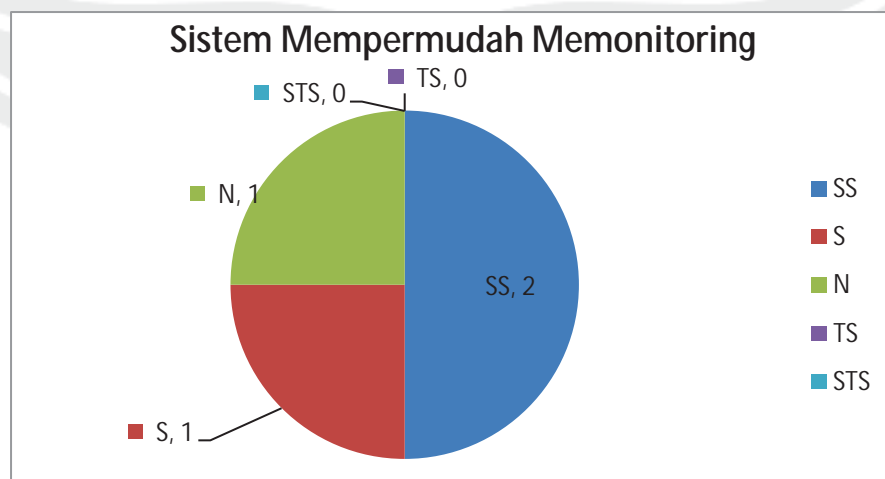


**Gambar 5.242. Pengujian Sistem Menampilkan Data Secara Cepat**

Dari gambar 5.242. dapat disimpulkan bahwa sistem dapat menampilkan data secara cepat. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 50%, setuju 25%, netral 25%, tidak setuju 0%, dan sangat tidak setuju 0%.

#### 5.5.9. Pengujian Sistem Mempermudah Memonitoring Aktivitas Kemahasiswaan dan Alumni

Pengujian ini dilakukan untuk membuktikan bahwa sistem mempermudah memonitoring aktivitas kemahasiswaan dan alumni. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 2 responden sangat setuju, 1 responden setuju, 1 responden netral, 0 responden tidak setuju, dan 0 responden sangat tidak setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.243.

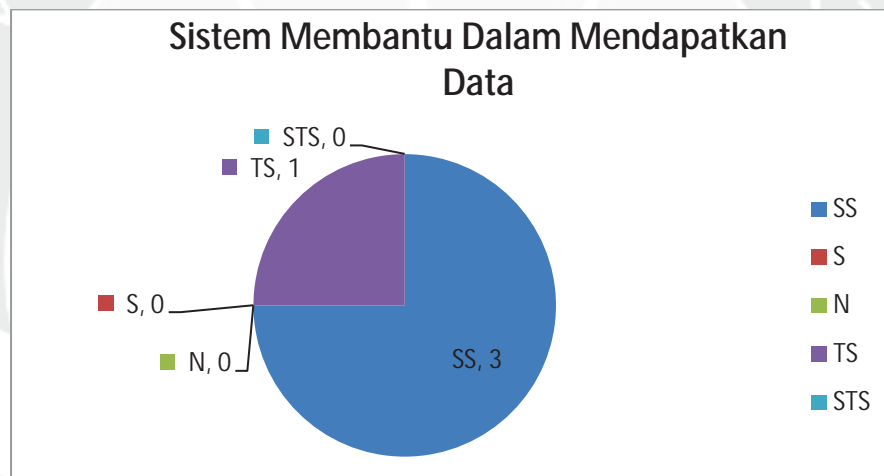


**Gambar 5.243. Pengujian Sistem Mempermudah Memonitoring Aktivitas Kemahasiswaan dan Alumni**

Dari gambar 5.243. dapat disimpulkan bahwa sistem dapat mempermudah memonitoring aktivitas kemahasiswaan dan alumni. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 50%, setuju 25%, netral 25%, tidak setuju 0%, dan sangat tidak setuju 0%.

#### 5.5.10. Pengujian Sistem Membantu Dalam Mendapatkan Data Kemahasiswaan dan Alumni

Pengujian ini dilakukan untuk membuktikan bahwa sistem membantu dalam mendapatkan data kemahasiswaan dan alumni. Dari kuesioner yang terkumpul, rekapitulasi data yang diperoleh adalah 3 responden sangat setuju, 0 responden setuju, 0 responden netral, 1 responden tidak setuju, dan 0 responden sangat tidak setuju. Berikut merupakan grafik pie dari hasil rekapitulasi data untuk pengujian tampilan sistem pada gambar 5.244.



**Gambar 5.244. Pengujian Sistem Membantu Dalam Mendapatkan Data Kemahasiswaan dan Alumni**

Dari gambar 5.244. dapat disimpulkan bahwa sistem dapat membantu dalam mendapatkan data kemahasiswaan dan alumni. Hal ini dibuktikan dengan prosentase jawaban sangat setuju 75%, setuju 0%, netral 0%, tidak setuju 25%, dan sangat tidak setuju 0%.