

# LECTURE NOTES

## 7023T Advanced Database System

### Session 09

## Designing and Developing ETL System

## LEARNING OUTCOMES

- Peserta diharapkan mampu memahami konsep dasar proses ETL (*extract – transform – load*).
- Peserta diharapkan dapat menjelaskan persyaratan dan kondisi (*requirements* dan *constraint*) apa saja yang perlu dipenuhi oleh ETL *tools*.
- Peserta diharapkan mampu mengidentifikasi kemampuan dari ETL *tools* yang saat ini beredar.
- Peserta diharapkan mampu memahami proses perancangan ETL.

### OUTLINE MATERI (Sub-Topic):

1. Pendahuluan
2. STM and Staging Table
3. Pengujian
4. Penanganan Kesalahan

## Pendahuluan

Kimball mengusulkan 10 langkah untuk membangun sebuah sistem ETL, baik yang dikerjakan secara manual dengan *script* maupun menggunakan *ETL tools*. Gambar 1 memperlihatkan diagram yang menjelaskan ke-10 langkah tersebut dan kaitannya dengan 34 subsistem dari ETL. Kondisi yang harus dipenuhi sebelum melakukan proses ETL diantaranya ketersediaan desain logis, rencana *high-level architecture*, dan STM (*source to target mapping*) untuk semua data. Desain fisik dan implementasi juga harus dipersiapkan sebelumnya. Hal lain yang penting adalah melakukan data *profiling* sebelum memulai pengembangan ETL.

ETL PROCESS STEP	ETL SUBSYSTEM			
	EXTRACTING DATA	CLEANING AND CONFORMING	DELIVERING FOR PRESENTATION	MANAGING THE ETL ENVIRONMENT
<b>Plan</b>				
Create a high level, one-page schematic of the source-to-target flow.	1			
Test, choose, and implement an ETL tool (Chapter 5).				
Develop default strategies for dimension management, error handling, and other processes.	3	4, 5, 6	10	
Drill down by target table, graphically sketching any complex data restructuring or transformations, and develop preliminary job sequencing.		4, 5, 6	11	22
<b>Develop One-Time Historic Load Process</b>				
Build and test the historic dimension table loads.	3	4, 7, 8	9, 10, 11, 12, 15	
Build and test the historic fact table loads, including surrogate key lookup and substitution.	3	4, 5, 8	13, 14	
<b>Develop Incremental Load Process</b>				
Build and test the dimension table incremental load processes.	2, 3	4, 7, 8	9, 10, 11, 12, 15, 16, 17	
Build and test the fact table incremental load processes	2, 3	4, 5, 8	13, 14, 16, 18	
Build and test aggregate table loads and/or OLAP processing.			19, 20	
Design, build, and test the ETL system automation.		6	17, 18, 21	22, 23, 24, 30

**Gambar 1.** 10 langkah ETL menurut Kimball dan kaitannya dengan 34 subsistem dari ETL

**Sources**

- Customer Master (RDBMS)
  - Slowly changing on demographics and account status
  - 25M customers – 10k new or changed customers/day
- Geography Master (RDBMS)
  - 15,000 geogs
- DMR system (COBOL flat file, 2000 fields, one row per customer)
  - "Unbucketize" from 13 months in one row
  - Process 750k customers/day
  - Missed meter reads, estimated bills, restatement of bills
- Meters (MS Access)
  - Old (pre-1972) meter types are not in the Meters Group's system
  - There are 73 known meter types
- Period (Spreadsheet)
  - How/by whom maintained??

**Targets**

- Customer
  - check RI
- Geography
  - check RI
  - Labels need cosmetic work!
- Electricity Usage
- Electric Meter
  - check RI
- Meter read date
  - Usage Month

## STM and Staging Tables

Sistem ETL memanfaatkan tabel *staging* untuk menyimpan data yang berubah sejak proses pemuatan data terakhir. Pada saat data tersebut akan dimuat ke sistem DW, proses transformasi terhadap data pada tabel *staging* pada umumnya dilakukan secara inkremental. Tabel *staging* disarankan diletakkan pada database atau *schema* terpisah, hal ini dimaksudkan agar dapat dipindahkan dengan mudah ke server lain untuk mendukung beban proses ETL yang tinggi. Peletakan tabel *staging* pada database terpisah juga menjaga kerampingan data dan memudahkan proses transformasi secara inkremental. Tabel 1 berikut mengilustrasikan STM dari sumber data ke tabel *staging* dan dari tabel *staging* ke tabel dimensi DIM\_SALES.

**Tabel 1a.** Ilustrasi STM dari sumber data ke tabel *staging*

No	Source system	Source table	Source attribute	Transformation logic	Target table	Target attribute
1	System1	TableA	Manager_name	N/A	STG_TableA	Manager_name

**Tabel 1b.** Ilustrasi STM dari sumber data ke tabel *staging*

No	Source system	Source table	Source attribute	Transformation logic	Target table	Target attribute
1	Staging	STG_TableA	Manager_name	Parse name using a comma separator	DIM_SALES	MGR_FIRST_NAME

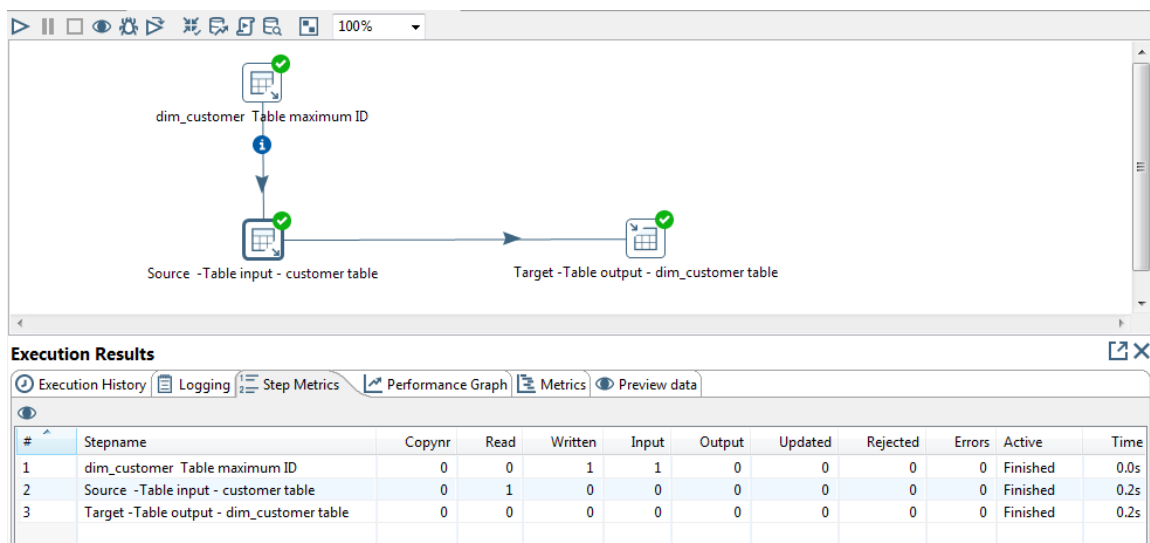
Berdasarkan informasi pada tabel 1a, proses pemuatan dari sumber data ke tabel *staging* STG\_TableA dapat dilakukan secara langsung karena tidak ada proses transformasi yang dilakukan (N/A). Sedangkan proses pemuatan data dari tabel *staging* STG\_TableA ke tabel dimensi DIM\_SALES membutuhkan beberapa proses transformasi seperti pemisahan kata berdasarkan karakter koma, penggabungan teks, penghapusan duplikasi, normalisasi string, pencocokkan, maupun *formatting*.

## Pengujian

Terdapat beberapa pendekatan dalam melakukan pengujian untuk melakukan validasi pergerakan data pada setiap *layer* sistem DW/BI. Saat ini ETL *tools* sudah dilengkapi dengan catatan berupa log yang memberikan informasi secara detail mengenai keseluruhan proses ETL. Catatan tersebut dapat digunakan untuk keperluan pengujian, sebagai contoh dengan melakukan perbandingan jumlah baris data yang berhasil diproses pada sebuah *layer* sumber dengan *layer* tujuan. Pendekatan lainnya, misalnya pengujian pada saat pemuatan data dilakukan perbandingan antara informasi penjualan pada masing-masing outlet pada *layer*

sumber dengan total penjualan untuk semua outlet pada *layer* target. Salah satu praktik yang dianjurkan adalah mempersiapkan strategi pengujian dengan cara membuat sekumpulan *script* untuk pengujian yang dapat digunakan untuk menguji proses pemuatan data dan menggunakannya pada pengujian pemuatan data tabel lainnya dengan hanya melakukan perubahan yang tidak signifikan. Pendekatan ini akan meningkatkan efisiensi dan konsistensi dari proses pengujian.

Gambar 3 memperlihatkan contoh log dari proses ETL pada perangkat lunak ETL dari Pentaho yang disebut Pentaho Data Integration tool. Proses ETL pada gambar tersebut dilakukan dalam tiga langkah, yaitu membaca dua tabel input: maximum\_ID dan customer dan memuat hasilnya ke tabel dimensi dim\_customer. Jumlah baris data yang diproses (*copy*, *read*, *write*, *update*, *reject*) pada setiap langkah disajikan pada log, termasuk jika ada kesalahan. Pada log juga disajikan waktu yang dibutuhkan untuk melakukan masing-masing langkah tersebut.



The screenshot shows the Pentaho Data Integration tool interface. At the top, a workflow diagram illustrates the ETL process: a source table 'dim\_customer Table maximum ID' feeds into a 'Source -Table input - customer table', which then feeds into a 'Target -Table output - dim\_customer table'. Below the diagram, the 'Execution Results' tab is active, displaying a table with the following data:

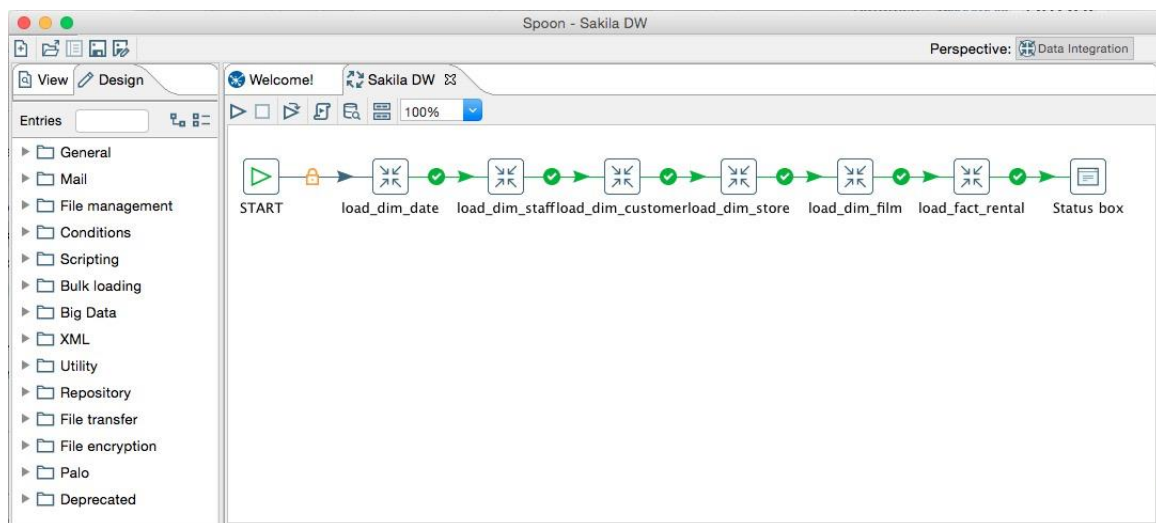
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time
1	dim_customer Table maximum ID	0	0	1	1	0	0	0	0	Finished	0.0s
2	Source -Table input - customer table	0	1	0	0	0	0	0	0	Finished	0.2s
3	Target -Table output - dim_customer table	0	0	0	0	0	0	0	0	Finished	0.2s

**Gambar 3.** Contoh log dari proses ETL pada Pentaho Data Integration tools

## Penanganan Kesalahan

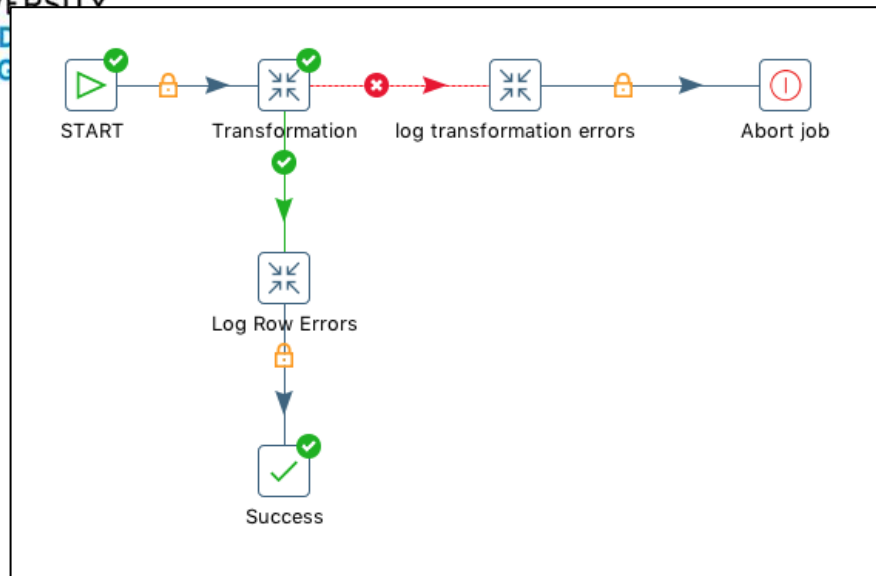
Setelah semua proses ETL untuk memuat semua tabel selesai dirancang, semua proses tersebut perlu diorganisasikan dalam suatu *workflow* atau urutan tertentu untuk melaksanakan proses pemuatan tabel-tabel dimensi dan tabel fakta sesuai dengan urutan tertentu. Gambar 4 memperlihatkan sebuah *workflow* ETL pada Pentaho Data Integration tools, pada *workflow* tersebut dapat dilihat bahwa pemuatan tabel-tabel dimensi (dim\_date,

dim\_staff, dim\_customer, dim\_store, dan dim\_film) dilakukan sebelum pemuatan tabel fakta (fact\_rental). ETL tools dilengkapi dengan kemampuan untuk merancang dan mengimplementasikan kondisi bersyarat saat melakukan pemuatan data dari berbagai sumber. Pada saat terjadi kesalahan, proses ETL dapat dikonfigurasi untuk membatalkan proses pemuatan data, melanjutkan eksekusi, atau melewati proses terhadap baris data yang menyebabkan kesalahan dan melanjutkan eksekusi ke baris data berikutnya. Gambar 5 memperlihatkan sebuah contoh *workflow* ETL dengan penanganan kesalahan. Pada gambar tersebut apabila terjadi kesalahan pada proses transformasi maka langkah yang akan dilakukan adalah membuat catatan log dari kesalahan tersebut kemudian membatalkan proses. Sedangkan jika proses transformasi berhasil dilakukan, langkah berikutnya adalah membuat catatan log kegagalan proses pada baris tertentu. ETL tools juga dilengkapi dengan kemampuan untuk menentukan nilai ambang (*threshold*) untuk meneruskan eksekusi hingga ditemukan jumlah kesalahan yang lebih besar daripada ambang yang sudah ditentukan sebelumnya.



**Gambar 4.** Contoh *workflow* ETL pada Pentaho Data Integration tools





**Gambar 5.** Contoh *workflow* ETL dengan penanganan kesalahan

Beberapa proses pemuatan data dapat diulangi dari posisi dimana mulai terjadi kesalahan, sementara untuk proses lainnya harus diulangi mulai dari awal. Sebagai contoh, diasumsikan data penjualan dari setiap outlet harus dimuat setiap hari pada malam hari agar pengguna dari kalangan bisnis dapat melakukan analisis penjualan setiap hari. Misalkan proses pemuatan data untuk tabel-tabel dimensi dan tabel fakta membutuhkan waktu sekitar 6 jam. Apabila terjadi kesalahan pada sebuah baris data yang diakibatkan oleh kesalahan format tanggal, maka proses pemuatan data untuk baris data tersebut dapat dilewatkan dan eksekusi dilanjutkan ke baris data berikutnya. Catatan kesalahan pada log dapat diinvestigasi pada kesempatan lain. Setelah hasil investigasi menemukan sumber kesalahannya, maka proses pemuatan khusus untuk data tersebut dapat diulangi lagi. Semua skenario terjadinya kesalahan dan penanganan yang diperlukan perlu dianalisis dengan baik, termasuk diantaranya menemukan dimana saja proses pemuatan data mungkin mengalami kesalahan dan menentukan prosedur penanganan kesalahan yang perlu dilakukan untuk mengatasi setiap kesalahan yang dapat diidentifikasi.



## SIMPULAN

- Sebelum melakukan proses ETL, tim ETL perlu mendokumentasikan rencana kerja dalam bentuk *high-level plan* agar proses ETL dapat berjalan sesuai dengan rencana
- Tabel *staging* berperan dalam memproses pemuatan data yang telah berubah sejak proses pemuatan terakhir dilakukan.
- Proses pengujian terhadap skenario ETL penting untuk dilakukan untuk mensimulasikan kondisi yang akan dihadapi saat proses ETL sesungguhnya dieksekusi.
- Skenario kesalahan dan penanganannya perlu dipersiapkan sebelum melakukan proses ETL, sehingga setiap kejadian kesalahan dapat diantisipasi sebelumnya.

## DAFTAR PUSTAKA

1. Kimball, R. (2008). *The Data Warehouse Lifecycle Toolkit*. John Wiley & Sons.
2. Kimball, R., & Ross, M. (2011). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons.
3. Inmon, W. H. (2005). *Building the Data Warehouse*. John Wiley & Sons.