

**MODUL PRAKTIKUM
KECERDASAN BISNIS**



**LABORATORIUM SISTEM INFORMASI DAN MULTIMEDIA
DIPLOMA KOMPUTER DAN SISTEM INFORMASI
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2014**

DAFTAR ISI

BAB I KONSEP DASAR KECERDASAN BISNIS

- 1.1. Tujuan Pembelajaran
- 1.2. Dasar Teori
 - 1.2.1. Latar Belakang
 - 1.2.2. Kecerdasan Bisnis
 - 1.2.3. *Data Warehouse*
- 1.3. Skenario
- 1.4. Latihan

BAB II ANALISA DAN PERANCANGAN OLAP

- 2.1. Tujuan Pembelajaran
- 2.2. Dasar Teori
- 2.3. Skenario
- 2.4. Latihan

BAB III PROSES EXTRACT TRANSFORM LOAD

- 3.1. Tujuan Pembelajaran
- 3.2. Dasar Teori
 - 3.2.1. *Extraction Transformation Loading (ETL)*
 - 3.2.2. *Pentaho Data Integration*
- 3.3. Skenario
 - 3.3.1. Pengaturan Koneksi Basis Data MySQL Menggunakan *Native JDBC*
 - 3.3.2. Pengaturan Koneksi Basis Data MySQL menggunakan *JNDI*
 - 3.3.3. Proses ETL untuk Dimensi
 - 3.3.4. Proses ETL untuk Fakta
- 3.4. Latihan

BAB IV SCHEMA WORKBENCH

- 4.1. Tujuan Pembelajaran

4.2. Dasar Teori

4.3. Skenario

 4.3.1. Koneksi Basis Data

 4.3.2. Schema

 4.3.3. Dimensi

 4.3.4. Cube

 4.3.5. Measure

4.4. Latihan

BAB V PENTaho BI SERVER

5.1. Tujuan Pembelajaran

5.2. Dasar Teori

5.3. Skenario

 5.3.1. Pengaturan Environment Variable

 5.3.2. Administrator Console dan User Console

 5.3.3. Publish Schema

 5.3.4. Analisis

5.4. Latihan

BAB VI CTOOLS DAN SAIKU

6.1. Tujuan Pembelajaran

6.2. Dasar Teori

 6.2.1. Community Data Browser

 6.2.2. Saiku untuk Pentaho

 6.2.3. Community Data Access (CDA)

 6.2.4. Community Dashboard Framework

 6.2.5. Community Dashboard Editor

 6.2.6. Community Charting Components (CCC)

6.3. Skenario

 6.3.1. Kegiatan 1: Instalasi CTools

 6.3.2. Kegiatan 2: Instalasi Saiku

6.3.3. Kegiatan 3: Analisis Menggunakan CDB

6.4. Latihan

BAB VII COMMUNITY DASHBOARD EDITOR

7.1. Tujuan Pembelajaran

7.2. Dasar Teori

7.3. Skenario

7.3.1. Header

7.3.2. Parameter

7.3.3. Objects

7.4. Latihan

BAB I

KONSEP DASAR KECERDASAN BISNIS

1.1. Tujuan Pembelajaran

- Mahasiswa dapat memahami dasar teori *Business Intelligence, Data Warehouse, OLAP*, dan *cube*.
- Mahasiswa dapat membuat pemodelan skema multidimensi dengan metode *star schema, snowflake*, dan *galaxy schema*.

1.2. Dasar Teori

1.2.1. Latar Belakang

Business Intelligence merupakan hasil evolusi dari *Executive Information System* (EIS) dan *Decision Support System* (DSS). EIS dan DSS memiliki tujuan untuk membantu dalam pengambilan keputusan. Implementasi *Business Intelligence* mampu meningkatkan nilai data dan informasi dalam organisasi.

Business Intelligence dalam prakteknya pada organisasi mampu memberikan nilai tambah untuk organisasi, diantaranya seperti contoh berikut :

- Peningkatan Laba

Business Intelligence pada sistem *Customer Relationship Manager* (CRM) mampu memberikan informasi dan pengetahuan tentang informasi pelanggan yang memberikan keuntungan maupun pelanggan yang tidak, sehingga organisasi dapat berfokus pada pelanggan yang memberikan keuntungan.

- Pengurangan Biaya

Business Intelligence pada sistem *Sales and Marketing* mampu memberikan informasi untuk mengevaluasi biaya yang dikeluarkan untuk tujuan pemasaran. Organisasi dapat mengurangi biaya pada media pemasaran yang kurang memberikan keuntungan dan menambah biaya investasi pada media pemasaran yang memberikan hasil maksimal.

1.2.2. Kecerdasan Bisnis

Kecerdasan bisnis merupakan metodologi, proses, arsitektur, dan teknologi yang memungkinkan transformasi data mentah menjadi informasi yang lebih bermakna dan bermanfaat. Kecerdasan bisnis memungkinkan *end-user* untuk membuat keputusan bisnisnya berdasarkan data *real-time*. Komponen utama dari *Business Intelligence* adalah :

- *Online Analytical Processing (OLAP)*
- *Data Warehouse*

Business Intelligence mampu membantu organisasi dalam mengambil keputusan dengan cara :

- Mengubah Data Menjadi Informasi
- Mengubah Informasi Menjadi Pengetahuan
- Mengubah Pengetahuan Menjadi Perencanaan

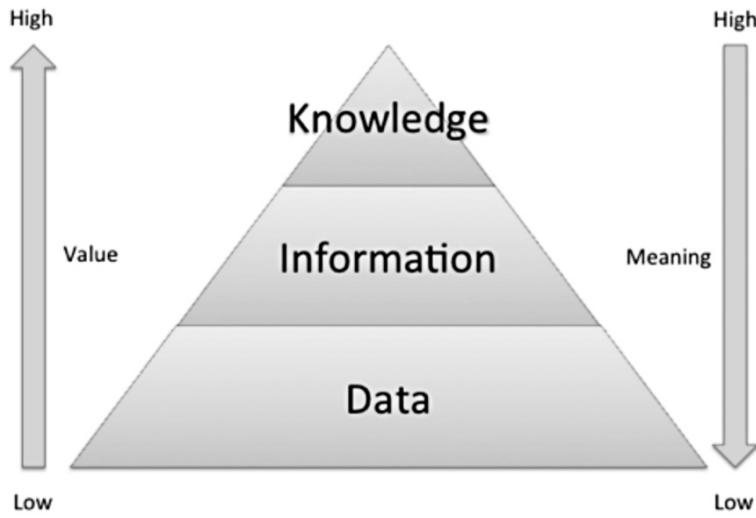


Figure 1. Data, Information, and Knowledge Hierarchy (Chaffey and Wood 2005)

Beberapa produk *Business Intelligence* yang dikembangkan oleh vendor software antara lain :

- Pentaho : <http://www.pentaho.com/>
- Oracle BI : <http://www.oracle.com/us/solutions/business-analytics/business-intelligence/overview/index.html>
- IBM Cognos : <http://www-01.ibm.com/software/analytics/cognos/>
- SiSense : <http://www.sisense.com/>
- SAP : <http://www.sap.com/pc/analytics/business-intelligence.html>
- IDashboards : <http://www.idashboards.com/>
- Jaspersoft : <https://www.jaspersoft.com/>
- Birst : <http://www.birst.com/>

1.2.3. Data Warehouse

1.2.3.1. Definisi

Data Warehouse adalah kumpulan data yang dihasilkan untuk mendukung pengambilan keputusan. *Data Warehouse* juga merupakan gudang data untuk masa kini dan masa lampau yang sangat bermanfaat bagi

kalangan management sebuah organisasi. (Turban).

1.2.3.2. Karakteristik

Data warehouse adalah koleksi data yang bersifat ***subject oriented, integrated, non-volatile*** dan ***time variant*** yang mendukung pengambilan keputusan. (Bill Inmon dan Hackathorn, 1994).

- *Subject Oriented*

Data Warehouse diorganisasikan berdasarkan subjek yang besar yang ada dalam organisasi (seperti data pelanggan, produk dan penjualan) bukan berdasarkan aplikasi yang digunakan. Hal ini menggambarkan bahwa menyimpan data *decision-support* lebih tepat daripada data *application-oriented*.

- *Non volatile*

Data Warehouse tidak diperbarui secara *realtime* melainkan secara periodik dari sistem operasional. Data baru selalu ditambahkan sebagai data yang berbeda bukan mengganti data yang sudah ada. Database selalu menyimpan data baru ini dan secara bertahap mengintegrasikan dengan data yang sebelumnya.

- *Time variant*

Data Warehouse hanya akurat dan valid pada beberapa waktu tertentu atau selama beberapa interval waktu. Data – data tersebut merupakan serangkain daripada *snapshot*.

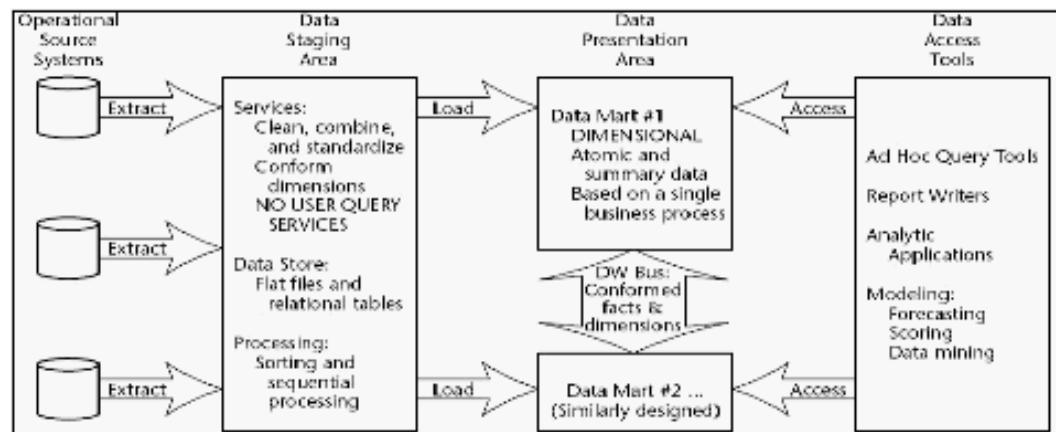
- *Integrated*

Terintegrasi karena menggabungkan beberapa data yang

berasal dari sumber dari aplikasi berbeda. Sering kali data-data tersebut tidak konsisten karena perbedaan format data. Sehingga integrasi data tersebut harus konsisten untuk menampilkan data yang seragam.

1.2.3.3. Arsitektur

Arsitektur *Data Warehouse* secara garis besar dibagi menjadi empat komponen utama yaitu operational source system, data staging area, data presentation area dan data access tools (Kimball & Ross, 2002).



(Kimball & Ross, 2002)

Operational Source System (OSS) merupakan sistem yang menyimpan semua data transaksi dari suatu aplikasi bisnis. Sehingga dibutuhkan kemampuan proses yang bagus dan ketersedian sistem yang handal. Data-data transaksi tersebut diolah (extract) kedalam data staging.

Data Staging Area (DSA) di dalamnya terdapat proses yang sering dikenal sebagai extract-transformation-load (ETL). Selain proses tersebut

diperlukan juga media penyimpanan sementara ketika data tersebut diolah dan sebelum diload ke dalam komponen selanjutnya.

ETL (*Extract Transformation Load*) merupakan proses pemindahan data operasional ke data warehouse. Terdapat tiga proses pada ETL ini, yaitu :

- *Extraction Data* ditemukan dan dipindahkan dari sistem operasional ke dalam data warehouse.
- *Transformation* Suatu program khusus yang dapat membersihkan data operasional agar sesuai dengan definisi awal aturan transformasi untuk data warehouse.
- *Loading* Suatu program yang seringkali digunakan untuk transformasi memindahkan data ke dalam tabel *data warehouse*.

Data Presentation Area merupakan tempat di mana data telah di aggregasi melalui proses ETL dan sudah siap untuk di query dalam proses analisa. Model data yang digunakan berbeda dengan model data pada OSS, di mana model yang digunakan adalah dimensional model bukan third-normal-form (3NF). Dalam dimensional model terdapat 2 tipe tabel yaitu fact dan dimension yang digabungkan dalam suatu data mart berdasarkan proses bisnis yang ada.

1.2.4. *Online Analytical Processing (OLAP)*

OLAP merupakan teknologi atau alat bantu untuk menganalisis data pada *Business Intelligence*. Sebuah *cube* dibangun untuk menyediakan multidimensi dan multilevel *view* yang memungkinkan untuk melakukan analisis pada data dengan beberapa perspektif dan beberapa rincian. Di dalam OLAP terdapat beberapa operasi – operasi dalam proses analisa query yaitu :

- *Rollup* , melakukan konsolidasi data dengan cara meningkatkan level aggregasi pada multidimensi data.
- *Drill-down* adalah kebalikan dari rollup dengan menurunkan level aggregasi ke level yang memberikan informasi lebih detil.
- *Slice dan dice* untuk melihat data dari titik pandangan yang berbeda

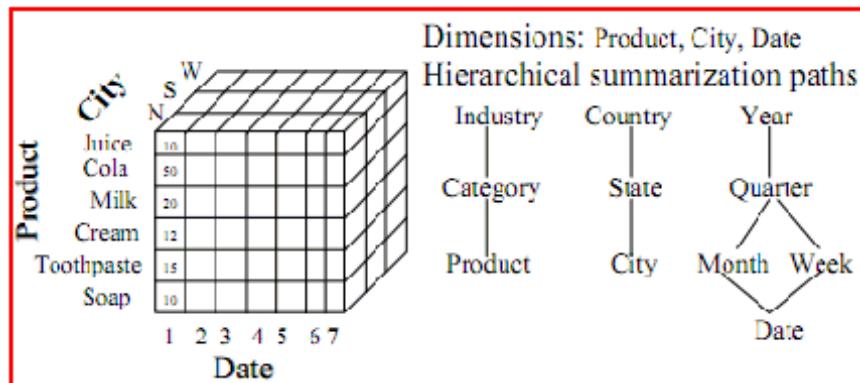
Perbedaan OLAP dengan *Online Transaction Processing* (OLTP) :

	OLTP	OLAP (DATA WAREHOUSE)
Sumber Data	Data operasional dari aplikasi	Gabungan dari beberapa data, dapat berasal dari kumpulan database OLTP
Fungsi	Membantu Operasional (aktivitas organisasi)	Membantu Pengambilan Keputusan (analisis)
Pengguna	Pegawai Administrasi	Manajerial
Bentuk Database	Berdasarkan aplikasi yang ada dalam organisasi (seperti data sistem CRM, data sistem akutansi)	Berdasarkan subjek besar yang ada dalam organisasi (seperti data pelanggan, produk dan penjualan) dan waktu
Data	Data Operasional subjek tertentu	Data Multidimensi dari beberapa subjek, Berupa Riwayat, Terintegrasi
Desain Database	Bentuk Normal (<i>Normalized</i>) dengan banyak tabel	Bentuk Tidak Normal (<i>De-normalized</i>) dengan sedikit tabel
Sifat Data	Data dinamis terus berubah sesuai operasional	Data statis, terus ditambahkan setiap interval

		waktu
Rentang Waktu	Menyimpan data saat ini	Menyimpan data riwayat

1.2.4.1. Model Multidimensi

Data Warehouse dibuat berdasarkan model data multidimensi yang diimplementasikan berupa bentuk *cube* yang disebut OLAP cube yang memiliki operasi-operasi standard OLAP.



OLAP Cube (Chen et all)

1.2.4.2. Pemodelan Data Warehouse

Pemodelan *Data Warehouse* menggunakan konsep pemodelan multidimensi yaitu :

- Terdiri dari tabel fakta dan beberapa tabel dimensi.
- Setiap tabel dimensi mempunyai primary key yang berelasi tepat satu komponen dari *composite key* di tabel fakta.
- Memiliki tabel dimensi waktu.
- Dapat dilakukan pilihan mengganti *natural keys* dengan *surrogate keys* (kunci pengganti) untuk mengantisipasi perubahan foreign key.

1.2.4.3. Tabel Fakta

Tabel fakta adalah tabel utama dalam konsep pemodelan multidimensi dimana terdapat hasil perhitungan bisnis (*measures*). Tabel fakta memiliki :

- Dua atau lebih *foreign key* ke tabel dimensi.
- *Composite key* atau *Concatenated key* yang merupakan gabungan dari *foreign key* ke tabel dimensi.
- *Data Grain* yang merupakan tingkat kekhususan untuk pengukuran atau kalkulasi, misalnya jumlah produksi, biaya produksi.

TABEL_FAKTA_PENJUALAN	
FK	<u>WAKTU_ID</u>
FK	<u>PRODUK_ID</u>
FK	<u>WILAYAH_ID</u>
	JUMLAH_PENJUALAN
	JUMLAH_RUPIAH

Contoh tabel fakta penjualan memiliki 3 dimensi
yaitu dimensi waktu , produk, dan wilayah

1.2.4.4. Tabel Dimensi

Tabel Dimensi merupakan tabel yang berisi penjelasan deskriptif dari sebuah bisnis (Kimbal, 2002). Dalam pemodelan multidimensi tabel dimensi memiliki banyak kolom atribut. Tabel dimensi memiliki :

- *Primary key* untuk mengidentifikasi setiap baris dimensi secara unik.
- Atribut berupa teks untuk mewakili deskripsi dan teks dari bisnis.
- Tabel dimensi terkadang dapat memiliki beberapa hirarki sehingga dapat dilakukan *drilling down* data.

- Tabel dimensi dapat memiliki kemampuan untuk mengambil agregasi data yang tinggi ke rendah.

TABEL_DIMENSI_PRODUK	
PK	<u>PRODUK_ID</u>
	PRODUK_NAMA
	PRODUK_DESKRIPSI
	PRODUK_BERAT

Contoh tabel dimensi produk

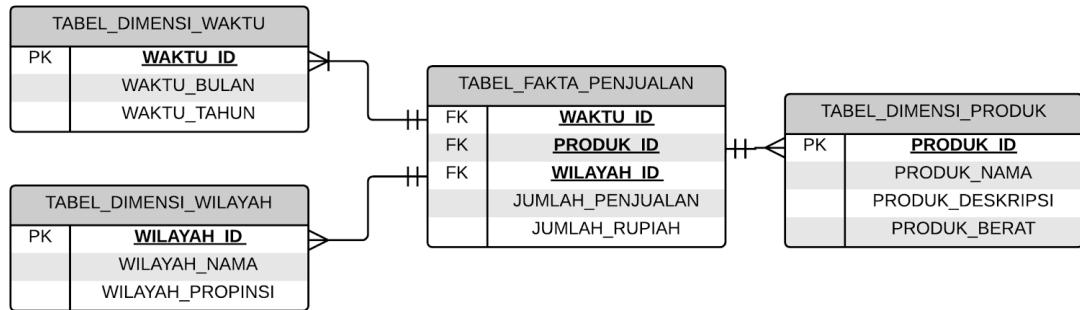
TABEL_DIMENSI_WILAYAH	
PK	<u>WILAYAH_ID</u>
	WILAYAH_NAMA
	WILAYAH_PROPINSI

Contoh tabel dimensi wilayah

TABEL_DIMENSI_WAKTU	
PK	<u>WAKTU_ID</u>
	WAKTU_BULAN
	WAKTU_TAHUN

Contoh tabel dimensi waktu

Tabel fakta dan dimensi di atas direlasikan sesuai dengan *foreign key* yang telah dibuat menjadi seperti berikut :



Contoh perspektif multidimensional *cube* dengan N dimensi berdasarkan tabel fakta dan dimensi di atas :

- 0 dimensi : Total Penjualan sampai sekarang yang terdiri dari tabel fakta penjualan.

Total	XXX
--------------	------------

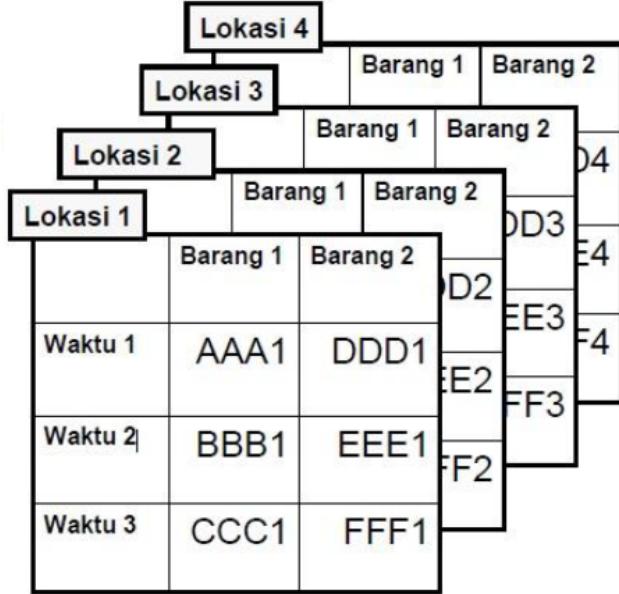
- 1 dimensi : Total penjualan untuk waktu tertentu

Waktu 1	AAA
Waktu 2	BBB
Waktu 3	CCC

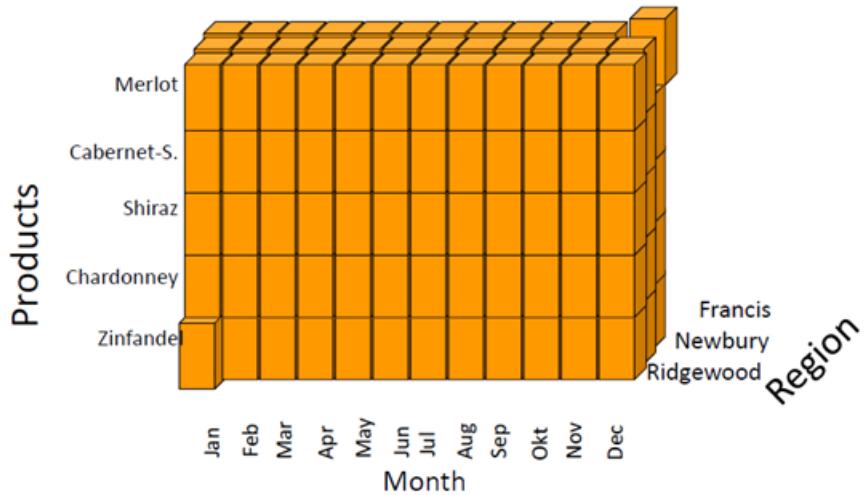
- 2 dimensi : Total penjualan untuk waktu tertentu dan barang tertentu

	Barang 1	Barang 2
Waktu 1	AAA	DDD
Waktu 2	BBB	EEE
Waktu 3	CCC	FFF

- 3 Dimensi : Total penjualan pada waktu tertentu untuk barang pada wilayah tertentu.



Sehingga jika digambarkan dalam representasi *cube* menjadi seperti berikut :

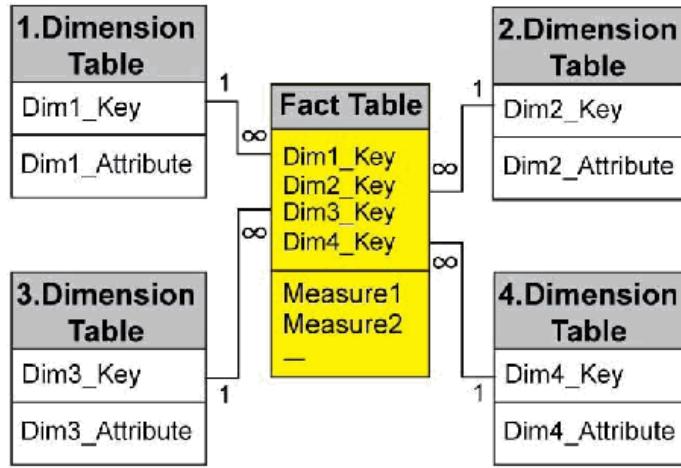


1.2.5. Skema Model Multidimensi

1.2.5.1. *Star Schema*

Star Schema adalah struktur logika yang memiliki tabel fakta yang berisi data fakta dan dikelilingi oleh tabel-tabel dimensi yang berisi data referensi

(bisa dalam bentuk denormalisasi).



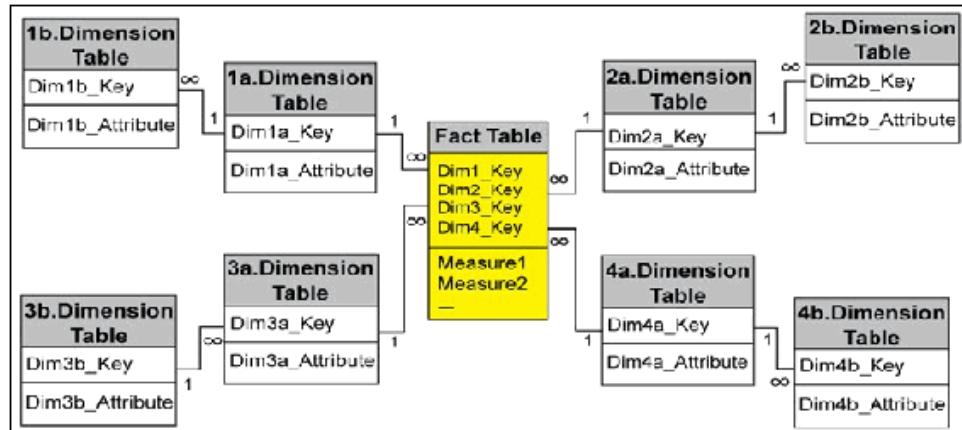
Contoh Skema Bintang

(www.juergen-konicek.de/Pictures/DWHSchemas.gif)

1.2.5.2. *Snowflake Schema*

Snowflake merupakan variasi lain dari *star schema* dimana tabel dimensi dari *star schema* dinormalisasi. Prinsip dasar dari skema ini tidak jauh berbeda dari *star schema*. Dalam menormalisasi tabel dimensi, ada beberapa pilihan yang dapat diperhatikan, antara lain :

- Secara parsial, lakukan normalisasi hanya beberapa tabel dimensi saja, dan sisakan yang lain tetap utuh.
- Secara lengkap atau parsial, lakukan normalisasi hanya pada beberapa tabel dimensi, dan tinggalkan yang tersisa dengan utuh.
- Secara parsial, lakukan normalisasi pada setiap tabel dimensi.
- Secara lengkap, lakukan normalisasi pada setiap tabel dimensi.

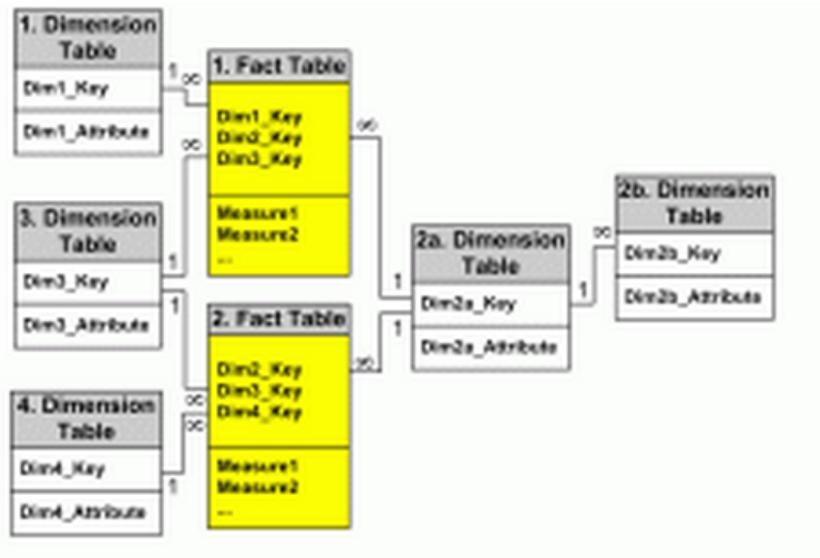


Contoh Skema Snowflake

(www.juergen-konicek.de/Pictures/DWHSchemas.gif)

1.2.5.3. *Fact Constellation* atau *Galaxy Schema*

Galaxy Schema terdiri dari kumpulan tabel fakta yang direlasikan secara tidak langsung oleh tabel-tabel dimensi yang sama.



Contoh *galaxy schema*

(www.juergen-konicek.de/Pictures/DWHSchemas.gif)

1.3. Skenario

Linus Wisata merupakan perusahaan yang bergerak di bidang jasa *traveling*. Linus Wisata menyediakan aplikasi *e-commerce* untuk menjual tiket kereta api secara online dan membuka gerai ruko sebagai agen penjualan. Saat ini sudah ada perusahaan lain yang memiliki model bisnis serupa. Persaingan harga dengan agen lain menjadi nilai kompetitif yang ditawarkan oleh perusahaan.

Untuk meningkatkan laba perusahaan, departemen pemasaran membuat program promosi berupa potongan harga untuk pembelian tiket kereta api dan tiket pesawat dengan destinasi tertentu. Namun perusahaan memiliki permasalahan karena tidak memiliki cukup informasi mengenai data penjualan pelanggan berdasarkan destinasi tujuan, waktu, dan kelas (eksekutif, bisnis, dan ekonomi). Strategi pemasaran akan memberikan promosi berupa potongan harga pada tiket dengan destinasi tujuan dengan penjualan tiket terbanyak.

Sebagai konsultan IT Anda diminta membuat rancangan skema multidimensi menggunakan model *star schema* yang akan digunakan untuk membuat aplikasi Dashboard. Buatlah rancangan skema berupa :

- Tabel fakta
- Tabel-tabel dimensi yang diperlukan
- Relasi antara tabel fakta dengan tabel dimensi
- Daftar tampilan perspektif *cube* penjualan tiket kereta dengan N dimensi berikut:
 - 0 Dimensi : Jumlah Penjualan
 - 1 Dimensi : Jumlah Penjualan dalam waktu tertentu
 - 2 Dimensi : Jumlah Penjualan dalam waktu dan destinasi tertentu
 - 3 Dimensi : Jumlah Penjualan dalam waktu, destinasi, dan kelas tertentu
- Buatlah struktur database tersebut di Database MySQL beserta relasinya.

1.4. Latihan

Buatlah rancangan skema *snowflake* untuk studi kasus di poin 1.3 dengan menambahkan normalisasi pada tabel dimensi Kelas (eksekutif, bisnis, ekonomi) dengan menambahkan satu tabel yaitu kereta api (misal : Malioboro Ekspres, Argo Parahyangan, Progo).

BAB II

ANALISA DAN PERANCANGAN OLAP

2.1. Tujuan Pembelajaran

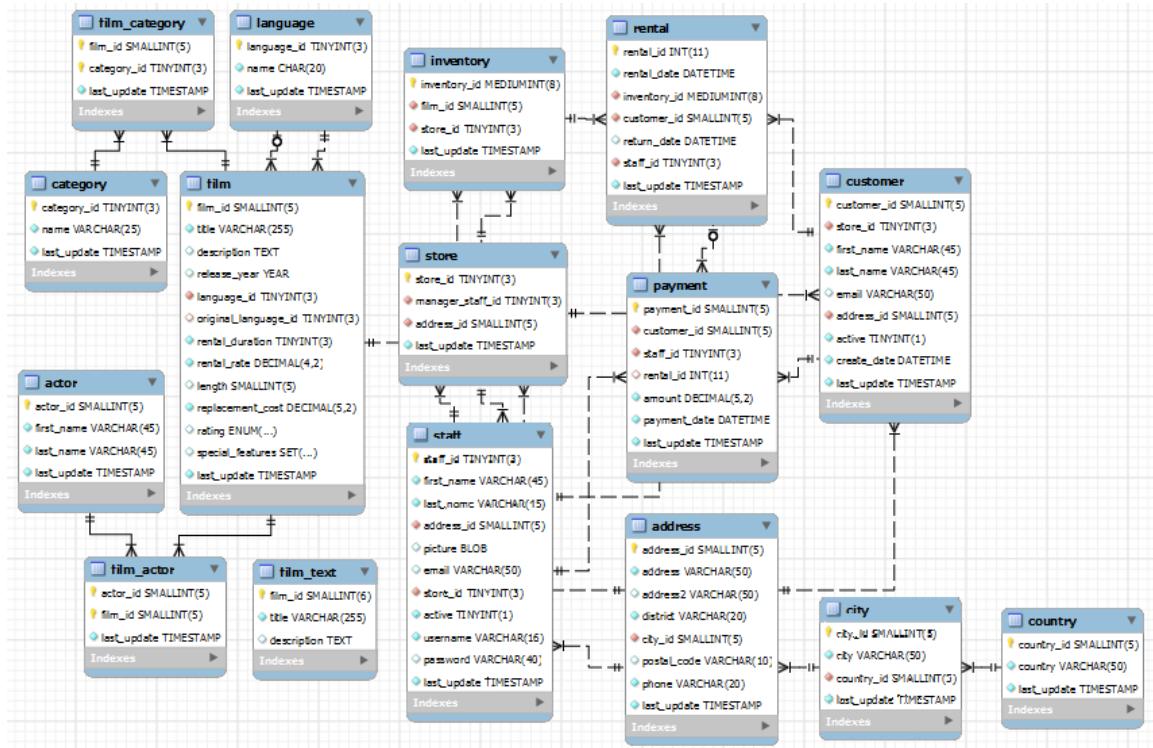
- Mahasiswa dapat menganalisa dan merancang tabel multidimensi.
- Mahasiswa dapat membedakan rancangan yang menggunakan *natural key* dan *surrogate key*

2.2. Dasar Teori

Model multidimensi merupakan suatu cara untuk merancang *data warehouse* (untuk penjelasan mendetail, dapat merujuk pada BAB I mengenai *Data Warehouse*).

2.3. Skenario

Terdapat sebuah basis data *sampel* bernama Sakila sebagai berikut:

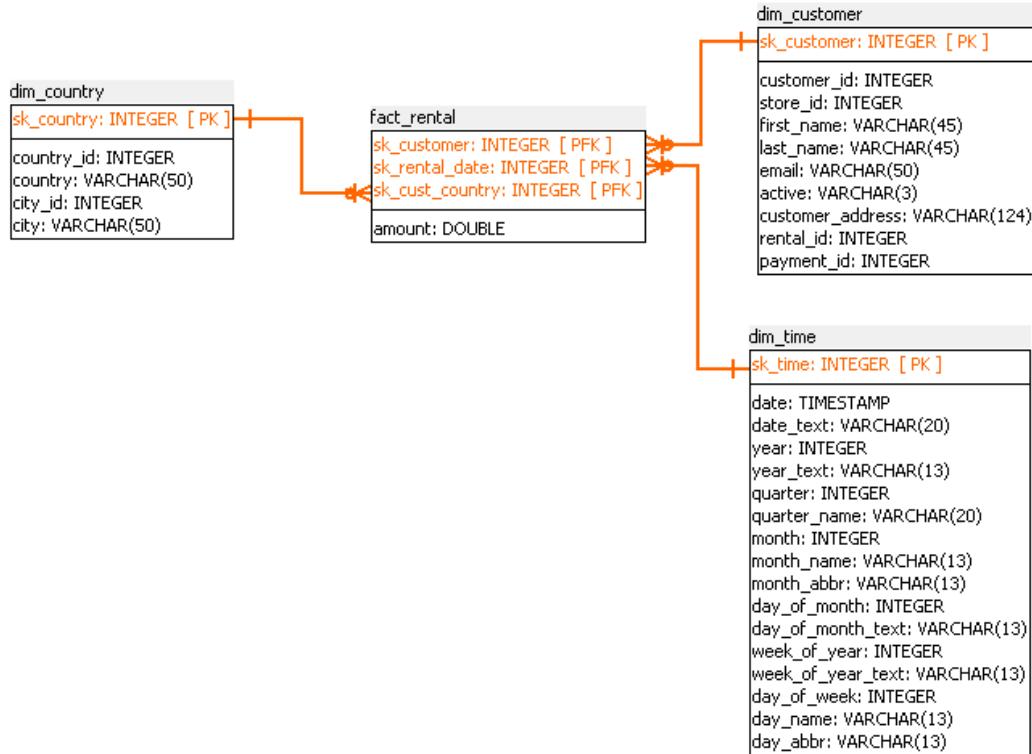


Akan dibuat rancangan model multidimensi untuk basis data Sakila. Dimulai dari tabel dimensi. Misal akan dibuat fakta sederhana mengenai *rental* yang berisi data mengenai siapa yang meminjam, dimana tempat tinggalnya dan kapan peminjamannya. *Measure* dari rental ini adalah jumlah uang yang harus dibayarkan peminjam dan berapa jumlah peminjamnya. Jumlah peminjam merupakan *derived measure* atau *measure* turunan, sehingga tidak perlu dimasukkan ke dalam tabel fakta (akan dibahas lebih lanjut pada BAB IV mengenai Mondrian Schema). Dimensi-dimensi yang akan dibuat antara lain adalah dimensi yang berkaitan dengan lokasi, waktu dan peminjam.

Star Schema

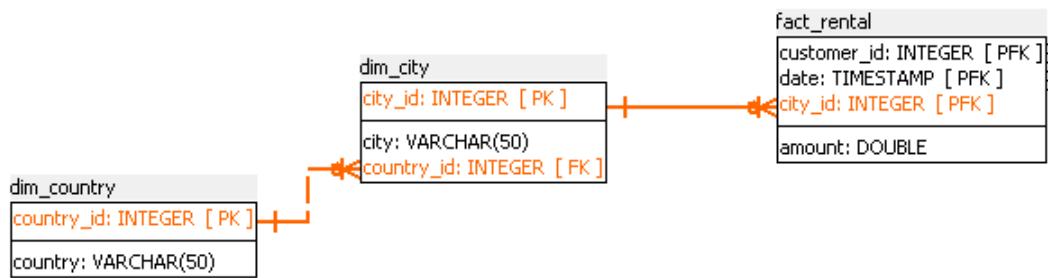
Rancangan *star schema* merupakan hasil denormalisasi dari hirarki yang ada. Dalam hirarki yang menggambarkan lokasi contohnya, dimisalkan akan diambil *city* dan *country*, dimana level tertinggi hirarki adalah *country* dan level dibawahnya adalah *city*.

Pada rancangan *star schema* dibawah ini digunakan *surrogate key* sebagai kunci uniknya. Salah satu penggunaan *surrogate key* adalah untuk mengantisipasi perubahan *foreign key* pada basis data asli.



Snowflake Schema

Rancangan *snowflake schema* merupakan perancangan multidimensi dimana tabel-tabel dimensinya dalam bentuk normal. Jika pada star schema *country* dan *city* berada pada satu dimensi yang sama, namun pada perancangannya menggunakan *snowflake*, *country* dan *city* akan ditempatkan pada tabel yang berbeda.



2.4. Latihan

Carilah basis data transaksional, kemudian buat analisa model multidimensional untuk basis data tersebut.

BAB III

PROSES EXTRACT TRANSFORM LOAD

3.1. Tujuan Pembelajaran

- Mahasiswa mampu membuat OLAP melalui proses transformasi menggunakan *Pentaho Data Integration* dengan basis data MySQL.

3.2. Dasar Teori

3.3.4. *Extraction Transformation Loading (ETL)*

ETL merupakan proses didalam basis data dan digunakan dalam pembentukan *data warehouse*. Proses ekstraksi data (*extraction*) merupakan suatu proses dimana data-data yang diperlukan diekstrak dari berbagai macam sumber, baik sumber yang homogen maupun heterogen. Proses transformasi (*transformation*) adalah proses dimana data-data yang akan diambil disesuaikan format ataupun strukturnya agar sesuai dengan analisis yang ingin dicapai. *Load* adalah proses penyimpanan data beserta format dan strukturnya ke basis data baru yang menjadi targetnya.

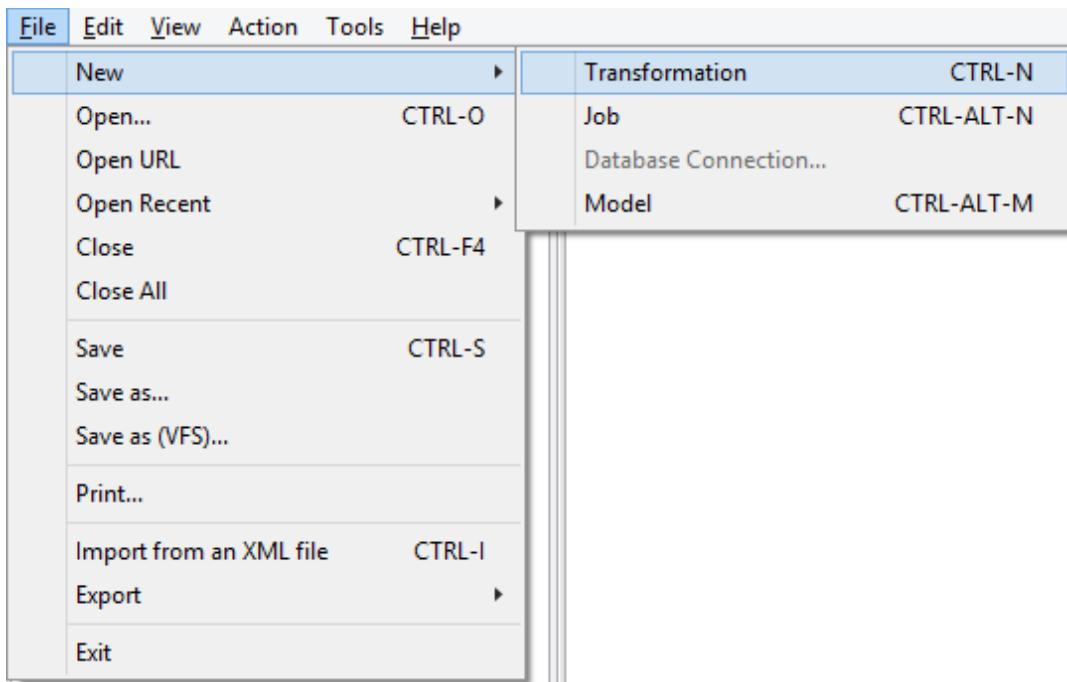
3.3.5. *Pentaho Data Integration*

Pentaho Data Integration (PDI), juga dikenal sebagai Kettle, merupakan salah satu perangkat yang dapat digunakan untuk melakukan proses *extraction*, *transformation*, dan *loading* (*ETL*). (<http://community.pentaho.com/>)

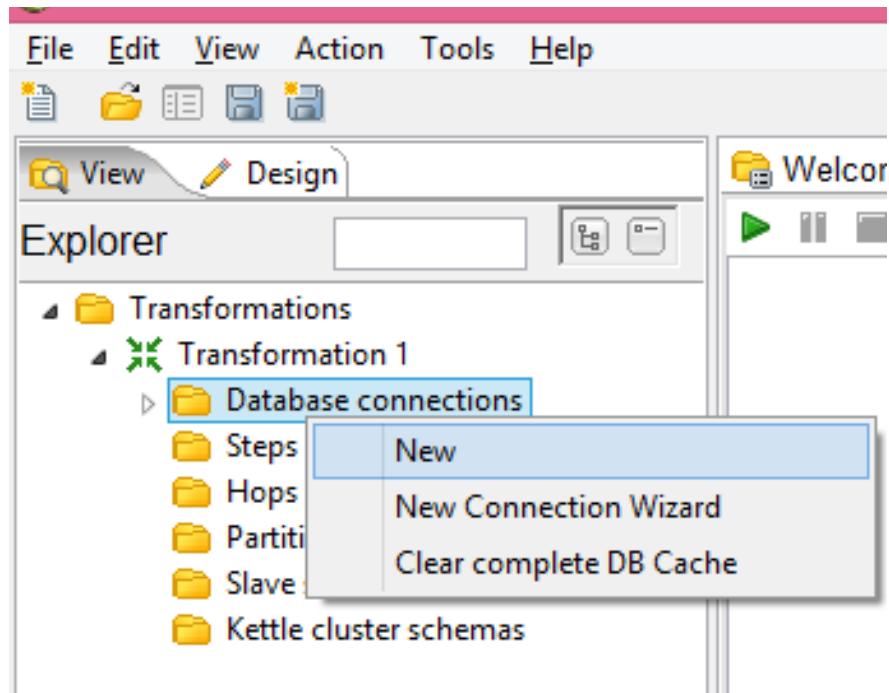
3.3. Skenario

Basis data yang digunakan dalam praktek kali ini adalah MySQL. Sebelum menjalankan PDI, pastikan *database connector library* untuk MySQL sudah tersedia dalam path {PDI path}\Pentaho Data integration\data-integration\libext\JDBC. Setelah *connector* tersedia, aktifkan PDI dengan melakukan *double click* pada file {PDI path}\Pentaho Data integration\data-integration\Spoon.bat.

Buatlah transformasi baru dengan cara sebagai berikut:



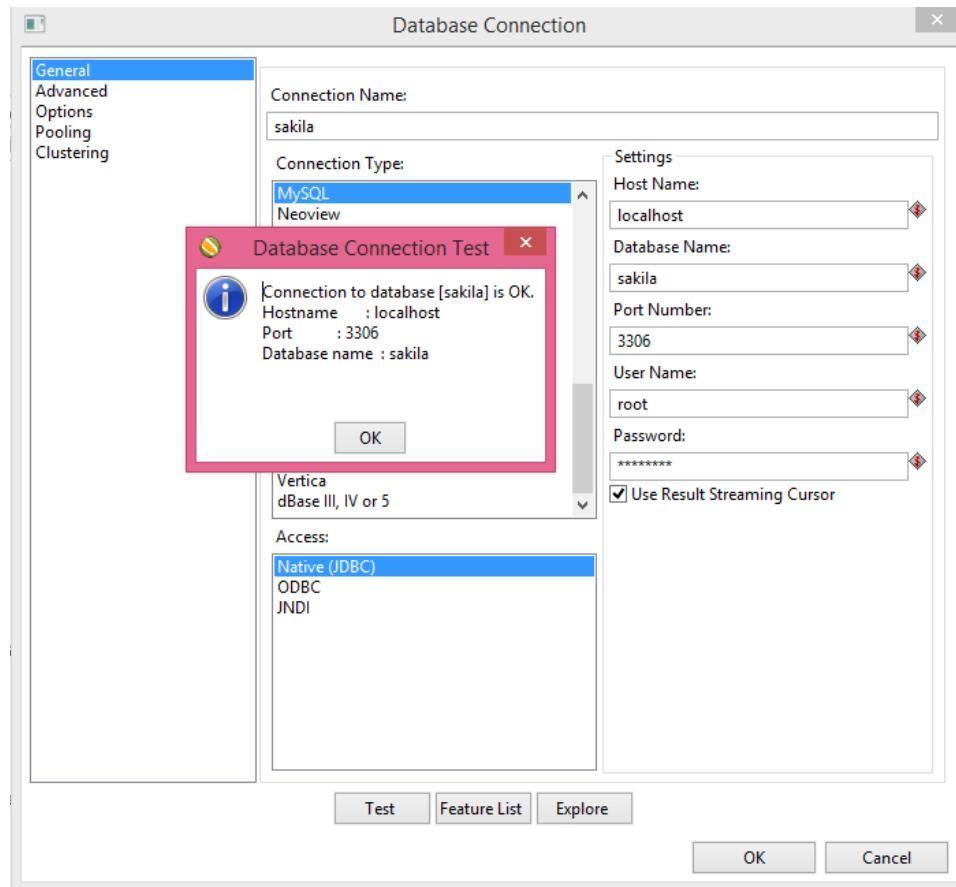
Setelah transformasi baru dibuat, selanjutnya diperlukan pengaturan koneksi ke dalam basis data. Koneksi basis data dapat dibuat dengan cara sebagai berikut:



3.3.1. Pengaturan Koneksi Basis Data MySQL Menggunakan *Native JDBC*

Pengaturan koneksi dengan menggunakan *Native JDBC* dilakukan dengan cara menyimpan pengaturannya didalam file transformasi yang sedang dibuat. Pengaturan ini merupakan *bad practice* karena jika terjadi perubahan pada salah satu koneksinya, maka file-file transformasi yang menggunakan pengaturan ini harus dibuka kembali menggunakan Spoon dan diubah satu per satu.

Ilustrasi pengaturan koneksi dengan menggunakan *Native JDBC* dapat dilihat pada gambar berikut:



3.3.2. Pengaturan Koneksi Basis Data MySQL menggunakan JNDI

Pengaturan koneksi basis data dengan menggunakan *JNDI* merupakan pengaturan *best practice*. Hal ini dikarenakan jika terjadi perubahan pengaturan, selama basis data yang digunakan masih sama dengan sebelumnya, maka tidak perlu dilakukan perubahan apapun pada file-file transformasi yang menggunakannya.

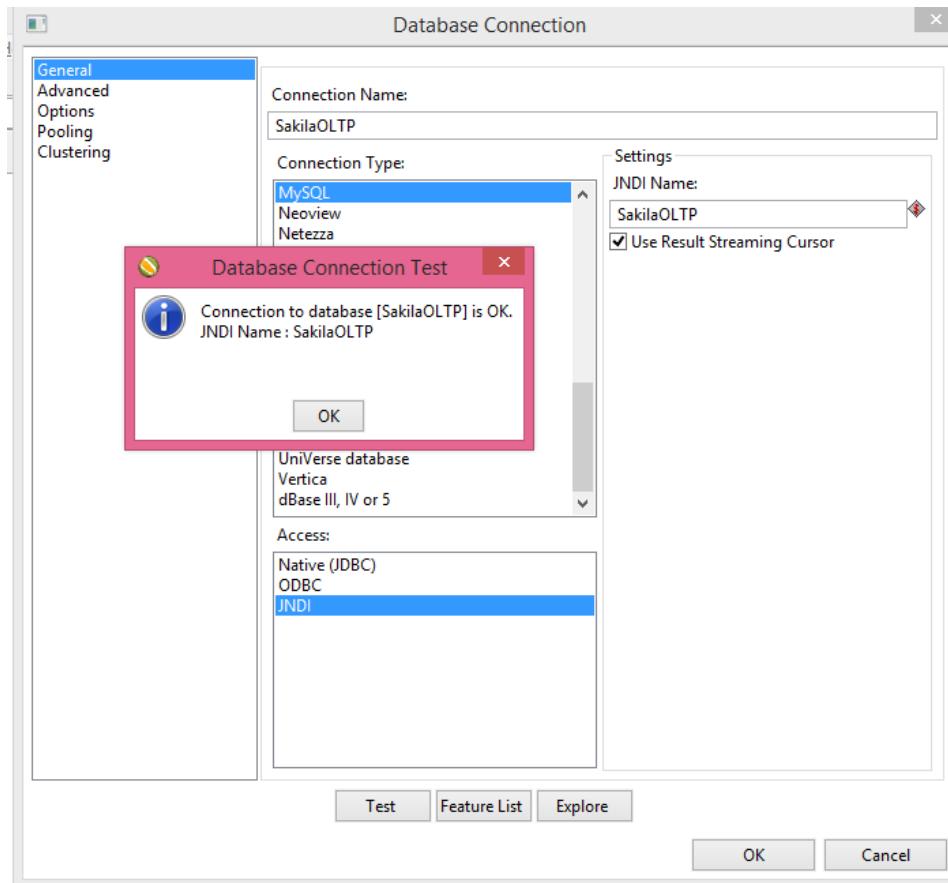
Untuk membuat koneksi dengan menggunakan *JNDI*, terlebih dahulu harus dibuat pengaturan pada file `{PDI path}\Pentaho\Data integration\data-integration\simple-jndi\jdbc.properties`. Di bagian paling bawah file, untuk membuat koneksi baru, tambahkan pengaturan dengan *template* sebagai berikut:

```
# Sakila OLTP
SakilaOLTP/type=javax.sql.DataSource
SakilaOLTP/driver=com.mysql.jdbc.Driver
SakilaOLTP/url=jdbc:mysql://localhost:3306/sakila
SakilaOLTP/user=root
SakilaOLTP/password=password
```

Baris dengan tanda **#** didepannya merupakan baris komentar. **SakilaOLTP** merupakan nama koneksi JNDI yang akan digunakan. Nama koneksi tidak terikat dengan aturan apapun namun sebaiknya disesuaikan dengan koneksi yang ingin dibangun. Pengaturan yang perlu diperhatikan pada file `jdbc.properties` dalam membuat koneksi JNDI adalah sebagai berikut:

- **type** diisikan dengan jenis koneksi yang akan digunakan.
- **driver** berisi driver basis data yang akan digunakan untuk membangun koneksi.
- **url** yang diisikan merupakan lokasi dimana basis data yang akan diakses berada.
- **user** berisi user basis data yang akan diakses.
- **password** berisi password dari user yang dituliskan di baris sebelumnya.

Setelah file `jdbc.properties` selesai dikonfigurasi, berikutnya dilakukan pula pengaturan pada PDI sebagai berikut:

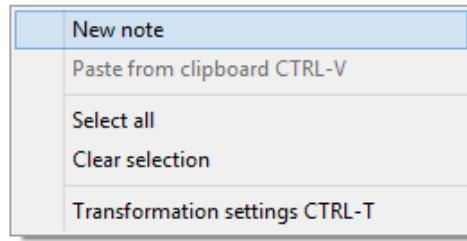


Perlu diperhatikan bahwa didalam kolom JNDI Name harus diisikan dengan nama JNDI yang telah diketikkan di dalam file `jdbc.properties`.

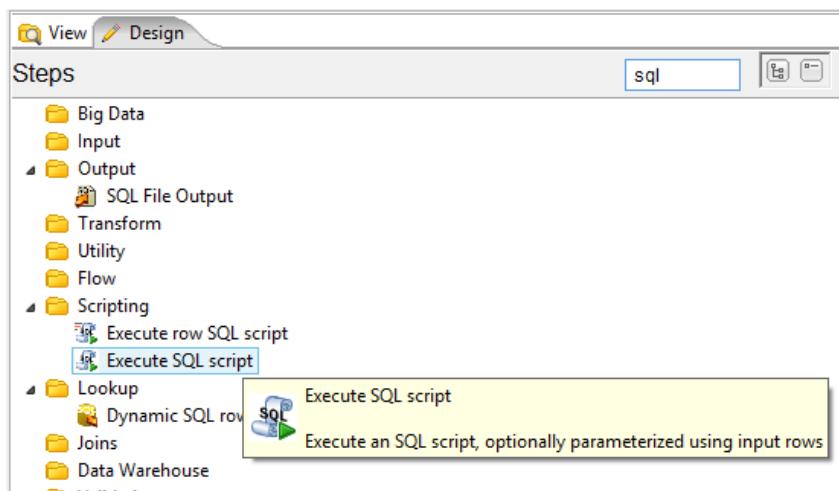
3.3.3. Proses ETL untuk Dimensi

Akan dibuat file transformasi untuk dimensi *country* sesuai dengan rancangan *star schema* yang ada di BAB II. Berikut adalah langkah-langkah untuk membuat transformasi dimensi *country*:

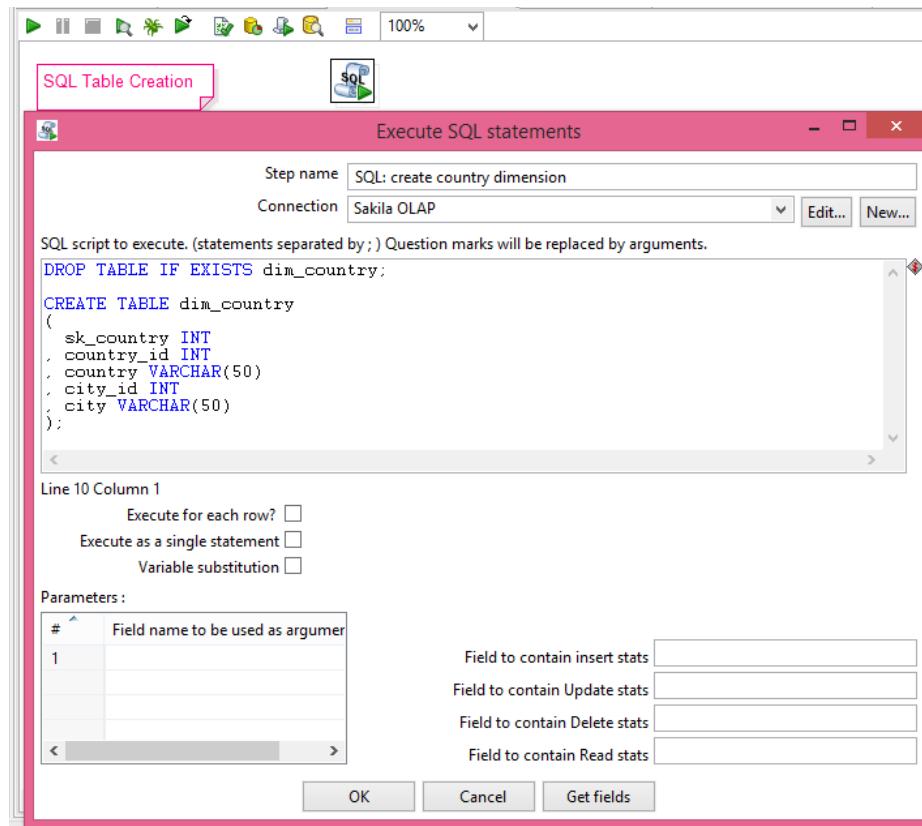
- Buat *note* dengan cara klik kanan pada tab transformasi yang baru dibuat. Note dapat juga digunakan sebagai komentar. Misal note diberi tulisan “SQL Table Creation”.



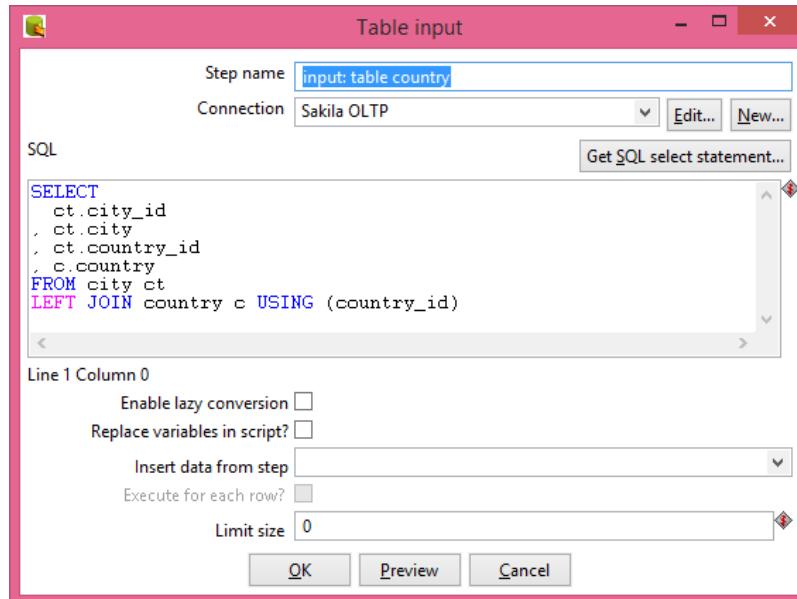
- Kemudian gunakan step (langkah) **Execute SQL Script** untuk dapat menjalankan baris perintah sql. Memilih step dapat dilakukan dengan cara mengetikkan bagian dari nama step pada kolom search seperti dibawah ini, kemudian drag and drop step yang ingin digunakan.



- Klik dua kali pada step yang sudah dipilih, lakukan pengaturan dan isikan tabel dimensi yang ingin dibuat.



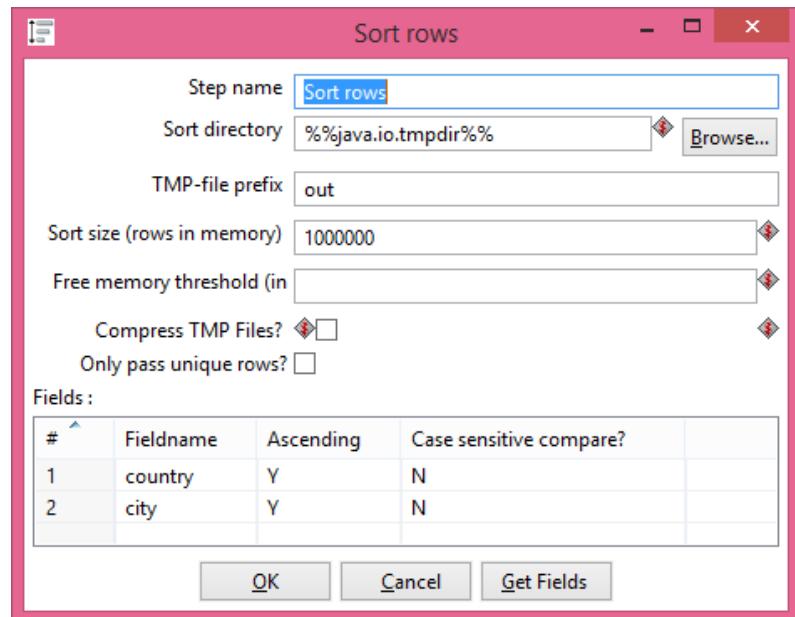
- Lakukan pencarian terhadap step Table Input, Sort Rows, Add Sequence, Select Values, Blocking Step, Generate Rows, dan Table Output.
- Pengaturan pada Table Input adalah sebagai berikut:



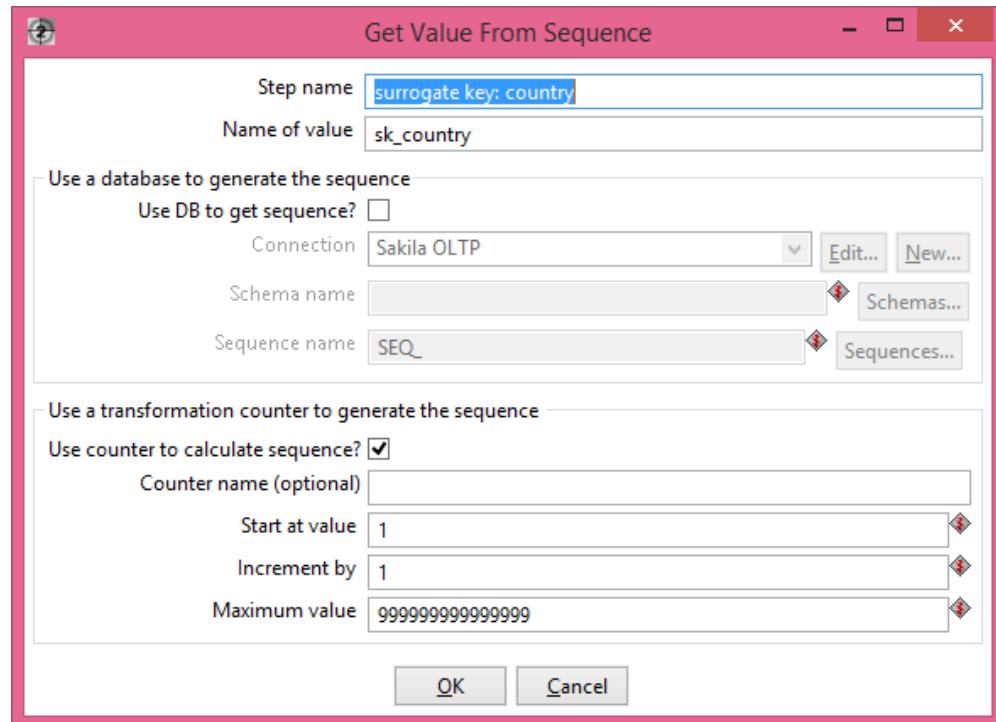
Step Table Input digunakan untuk mengekstraksi data dari basis data asal (OLTP) Pada kolom Connection, isikan koneksi ke basis data asal.

Untuk menggabungkan dua tabel, selain menggunakan query join, dapat juga digunakan Stream Lookup. Jika menggunakan Stream Lookup, perlu adanya dua step Table Input yang digunakan. Satu untuk query ke tabel city dan satu lagi untuk query ke tabel country.

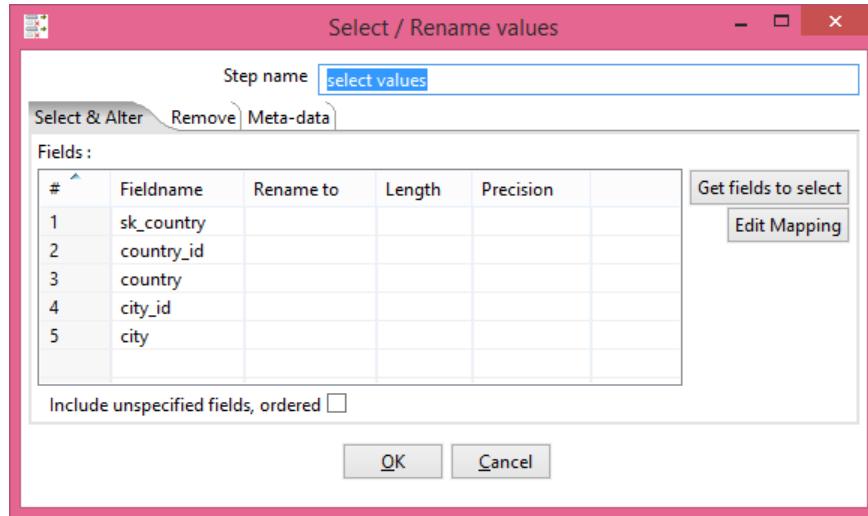
- Step Sort Rows digunakan untuk mengurutkan data yang telah kita ambil. Pada contoh dibawah ini, data akan diurutkan secara Ascending berdasarkan country terlebih dahulu, kemudian diurutkan secara Ascending berdasarkan city.



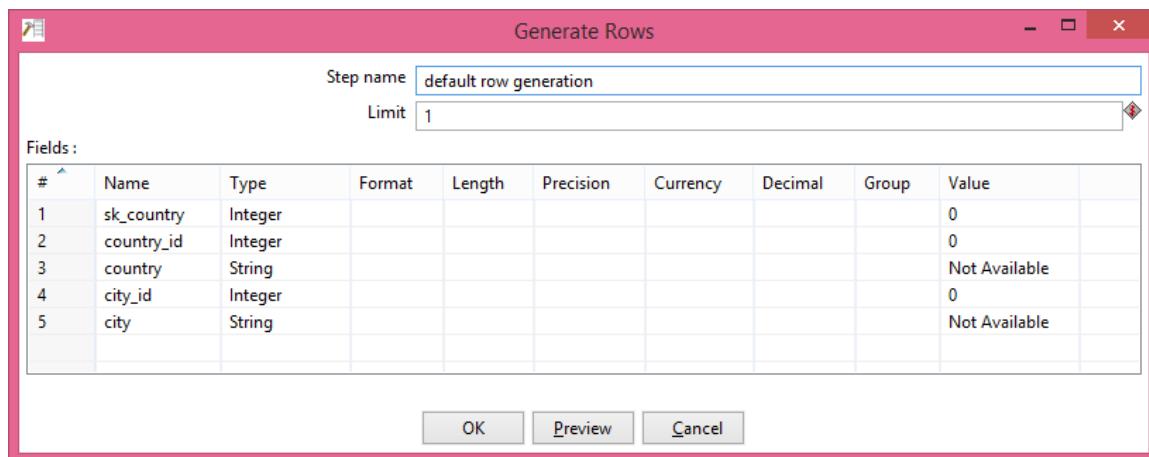
- Karena pada contoh ini digunakan Surrogate Key, maka diperlukan step Add Sequence untuk men-generate Surrogate Key.



- Select Values digunakan untuk menyaring data-data yang diperlukan untuk step selanjutnya. Select Values juga dapat digunakan untuk mengubah metadata.

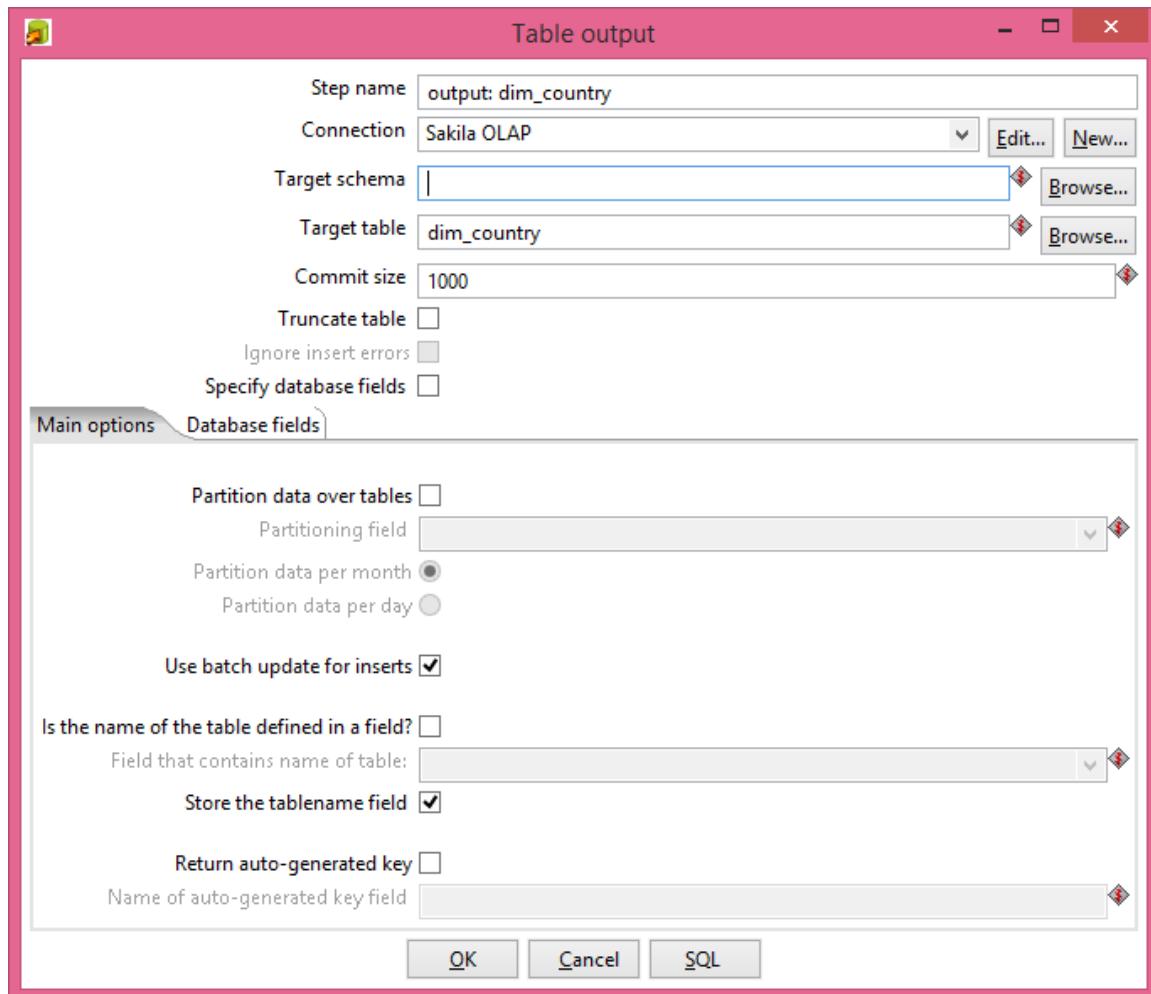


- Generate Rows digunakan untuk men-generate satu baris data yang berisi nilai default. Nilai default ini digunakan apabila terdapat data yang tidak ditemui referensinya didalam tabel dimensi.

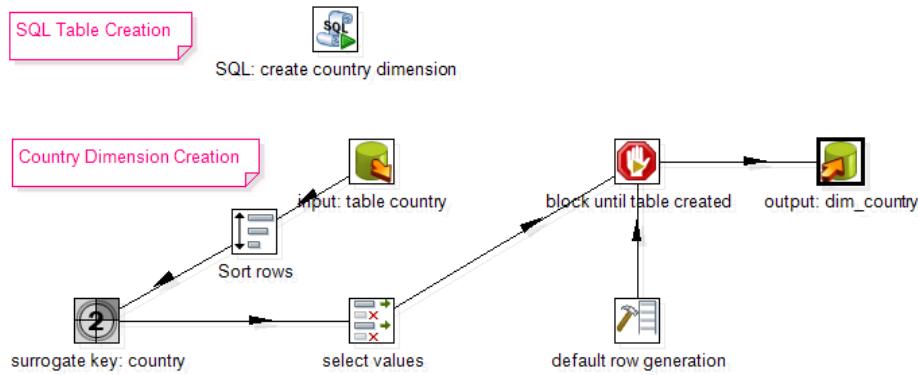


- Blocking Step digunakan untuk memastikan pada saat data siap dimasukkan ke dalam OLAP, tabel yang dituju telah berhasil dibuat. Oleh karena itu, step setelahnya tidak dapat dijalankan sebelum step query untuk membuat tabel dimensi country selesai dijalankan.

- Table Output digunakan untuk proses memasukkan data ke dalam basis data tujuan (OLAP).



Hasil akhir dari pembuatan proses transformasi dimensi adalah sebagai berikut:

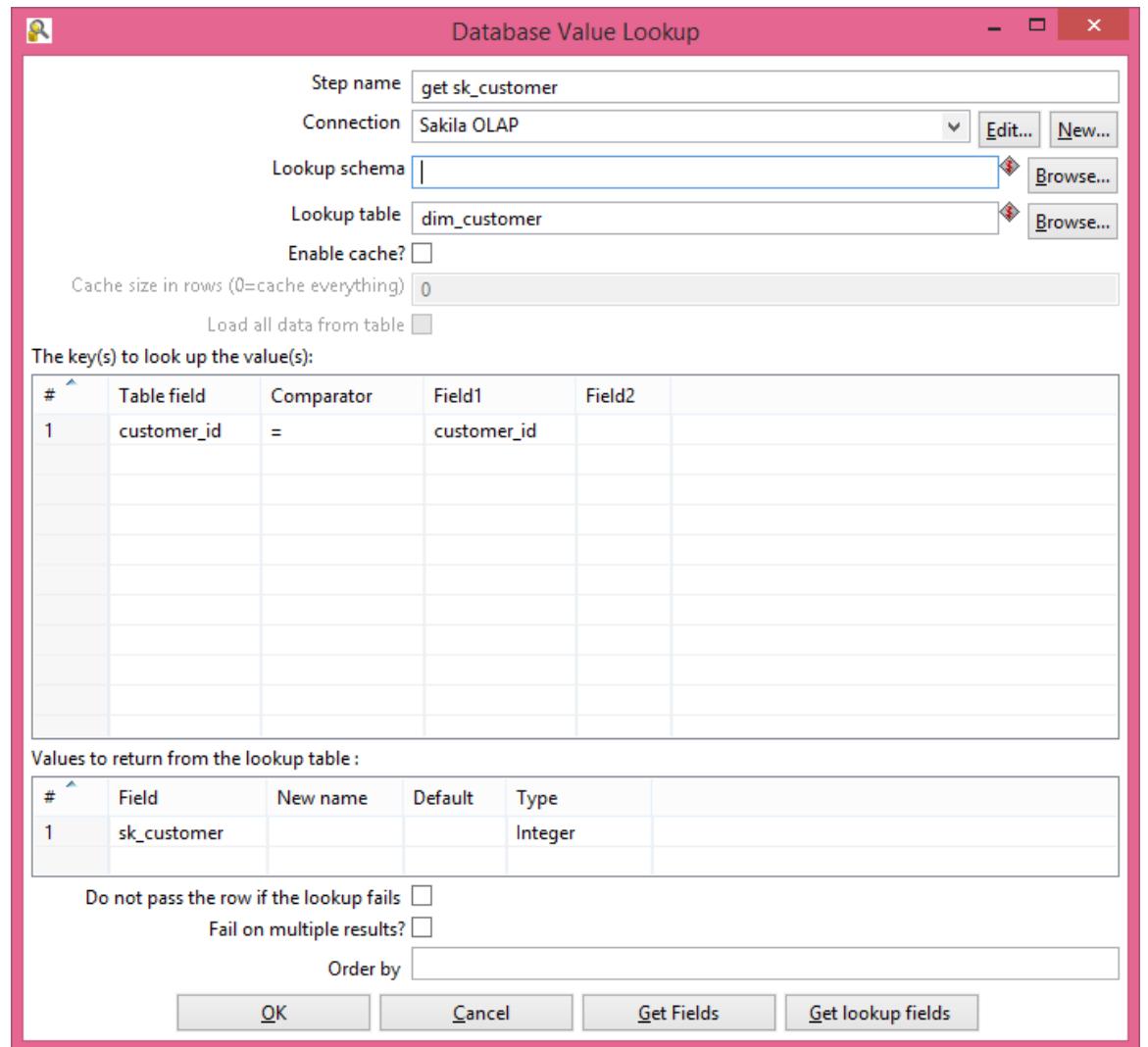


3.3.4. Proses ETL untuk Fakta

Untuk membuat tabel fakta, diperlukan perbandingan antara data yang diperlukan untuk tabel fakta dengan yang ada pada tabel dimensi untuk diambil key-nya. Key inilah yang kemudian digunakan untuk membangun tabel fakta. Selain key-key dari setiap dimensi yang diperlukan, terdapat juga measure sebagai pelengkap susunan tabel fakta.

Transformasi ini memerlukan step Table Input yang digunakan untuk mengambil seluruh data yang diperlukan oleh tabel fakta.

Tambahan yang perlu diperhatikan dalam membuat tabel fakta adalah step Database Lookup. Database lookup digunakan untuk mengambil key yang diperlukan dari dimensi yang sesuai. Sebagai contoh, berikut ini adalah database lookup untuk mendapatkan sk_customer.



Perlu diperhatikan bahwa nama step yang digunakan haruslah sesuai dengan fungsinya. Nama step perlu diubah terutama jika terdapat step yang sama pada satu transformasi, hal ini dikarenakan adanya batasan bahwa nama-nama step yang terdapat dalam satu transformasi haruslah unik.

Didalam step terdapat alur proses pengiriman data yang disimbolkan dengan panah. Data berasal dari suatu step ke step berikutnya yang ditunjuk oleh anak panah. Untuk membuat panah, dapat digunakan beberapa cara. Dua diantaranya adalah dengan cara: (1) menyeleksi kedua step yang inginkan dihubungkan, kemudian klik kanan pada salah satu step dan pilih New hop; (2) dengan

menggunakan mouse, klik step asal menggunakan scroll bar dan lepaskan pada step tujuan.

3.4. Latihan

Dengan menggunakan desain model dimensional pada latihan di BAB II, buatlah file-file transformasi untuk melakukan transformasi tabel dimensi dan tabel fakta.

BAB IV

SCHEMA WORKBENCH

4.5. Tujuan Pembelajaran

- Mahasiswa dapat membuat skema dari sebuah model multidimensional menggunakan Schema Workbench.

4.6. Dasar Teori

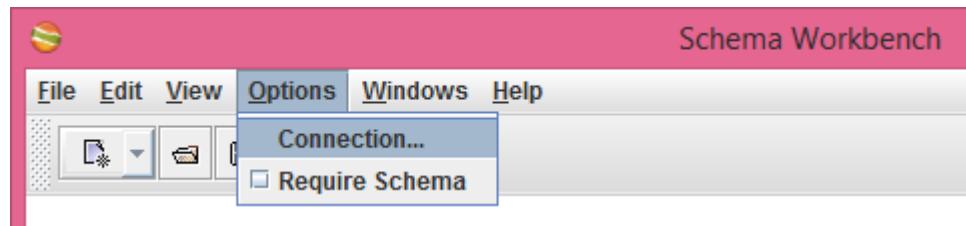
Schema Workbench merupakan perangkat desain yang digunakan untuk membuat dan mengetes skema OLAP cube. Data dapat dibentuk secara multidimensional sesuai dengan rancangan yang telah dibuat. Schema Workbench memungkinkan pengguna untuk membuat dimensi dan measure yang diperlukan untuk membangun cube yang merupakan representasi dari fakta.
[\(http://community.pentaho.com/\)](http://community.pentaho.com/)

4.7. Skenario

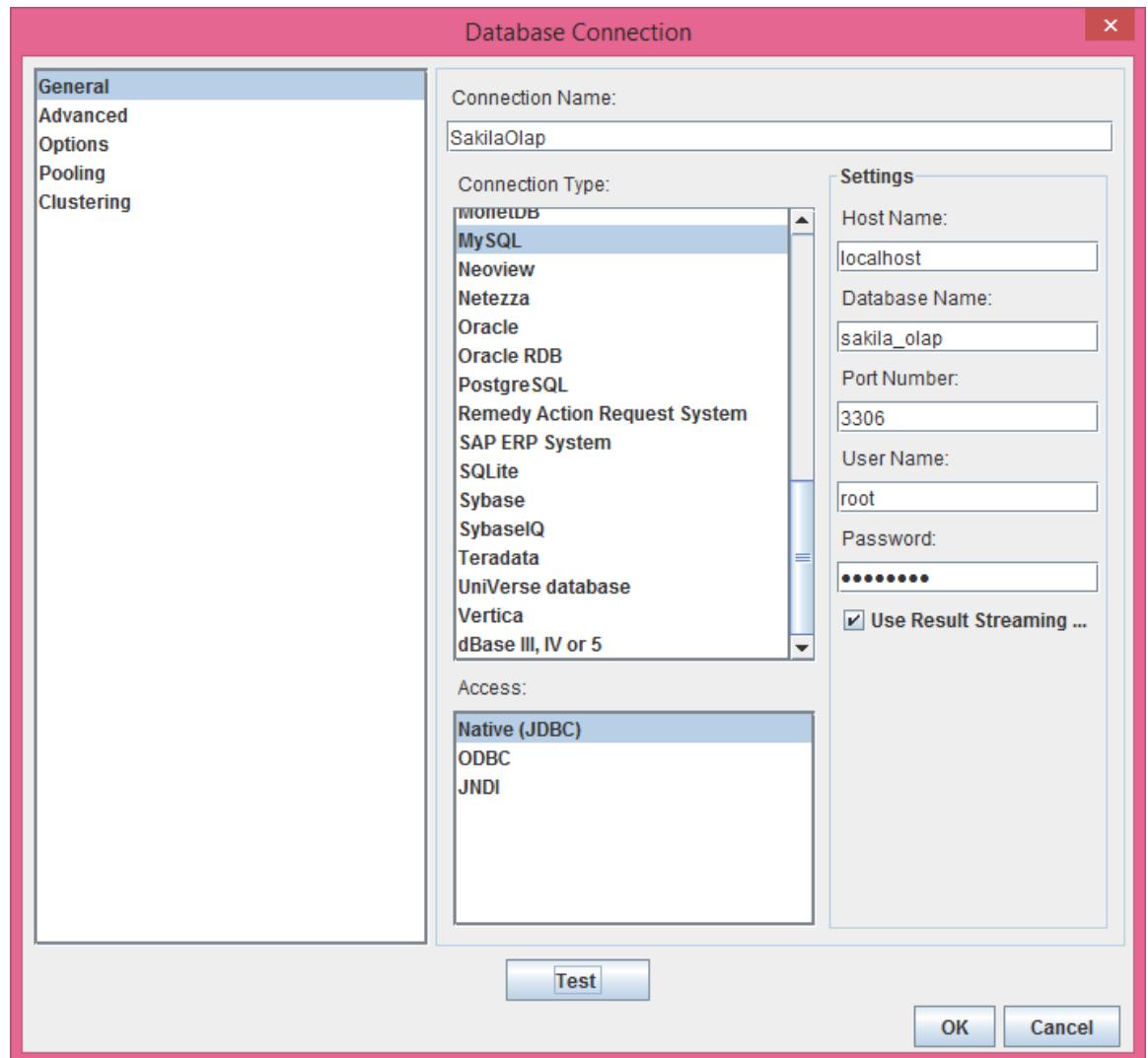
4.3.1. Koneksi basis data

Sebelum mengaktifkan Schema Workbench, pastikan driver koneksi basis data yang ingin digunakan sudah tersedia didalam folder {Schema Workbench path}\Schema Workbench\schema-workbench\drivers. Kemudian aktifkan Schema Workbench dengan double click pada file workbench.bat yang ada dalam folder {Schema Workbench path}\Schema Workbench\schema-workbench.

Untuk menghubungkan Schema Workbench dengan basis data, klik pada menu Options > Connections seperti gambar dibawah ini:



Kemudian buat koneksi dengan Native JNDI seperti dibawah ini:



4.3.2. Schema

Pengaturan yang perlu diperhatikan saat membuat skema baru adalah

nama skemanya.

Schema	
Attribute	
name	Sakila OLAP
description	
measuresCaption	
defaultRole	

4.3.3. Dimensi

Schema Workbench memungkinkan pembuatan dimensi beserta hierarki yang dimilikinya. Contoh dimensi yang berhierarki adalah dimensi country dimana level teratasnya adalah country dengan level berikutnya sekaligus level terakhirnya adalah city. Hal ini menunjukkan bahwa di dalam sebuah negara terdapat lebih dari satu kota.

Berikut ini adalah langkah-langkah dalam membuat sebuah dimensi dengan hierarki:

- Tambahkan satu dimensi untuk membuat dimensi country, beri nama country untuk dimensinya. Dimensi country ini merupakan standard dimension.

Attribute	
name	Country
description	
foreignKey	
type	StandardDimension
usagePrefix	
caption	
visible	<input checked="" type="checkbox"/>

- Pilih hierarki, kemudian klik kanan dan pilih menu Add Table. Pilih nama tabel yang akan digunakan.

Attribute	
schema	
name	dim_country
alias	

- Klik pada hirarki, masukkan kolom primary key dengan key dimensi yang digunakan.

Hierarchy for 'Country' Dimension	
Attribute	Value
name	Country
description	
hasAll	<input checked="" type="checkbox"/>
allMemberName	
allMemberCaption	
allLevelName	
defaultMember	
memberReaderClass	
primaryKeyTable	
primaryKey	sk_country
caption	
visible	<input checked="" type="checkbox"/>

- Klik kanan pada hirarki dan masukkan level baru. Level pertama untuk country dengan pengaturan sebagai berikut:

Level for 'Country' Hierarchy	
Attribute	Value
name	Country
description	
table	
column	country
nameColumn	
parentColumn	
nullParentValue	
ordinalColumn	country_id
type	String
internalType	String
uniqueMembers	<input type="checkbox"/>
levelType	Regular
hideMemberIf	Never
approxRowCount	
caption	
captionColumn	
formatter	
visible	<input checked="" type="checkbox"/>

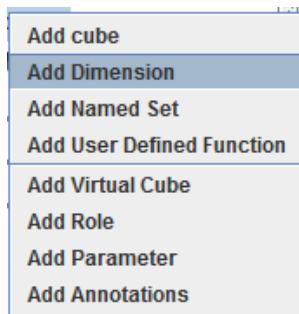
- Tambahkan level city. Pengaturan pada level city sama seperti pada level country dengan beberapa penyesuaian.

Ada dua jenis dimensi yang dapat dibuat, yakni Shared Dimension dan Local Dimension.

4.3.2.1. Shared Dimension

Shared Dimension merupakan dimensi yang dapat

digunakan oleh banyak cube (fakta). Penggunaan Shared Dimension memungkinkan model galaxy schema. Untuk membuat Shared Dimension, klik kanan pada Schema dan pilih opsi Add Dimension.



4.3.2.2. Local Dimension

Local Dimension merupakan dimensi yang hanya dimiliki oleh cube yang menaunginya dan hanya dapat digunakan pada cube tersebut. Untuk dapat menggunakannya, klik kanan pada cube dan pilih Add Dimension.

4.3.4. Cube

Cube merupakan representasi dari fakta. Untuk membuat cuba, klik kanan pada schema lalu klik Add Cube. Langkah-langkah pembuatan cuba adalah sebagai berikut:

- Beri nama cube sesuai dengan fakta yang ingin dibuat.
- Tambahkan tabel fakta.
- Tambahkan dimension jika menggunakan local dimension.
Tambahkan dimension usage jika menggunakan shared dimension.
Pengaturan untuk dimension usage adalah sebagai berikut:

Dimension Usage for 'Fact Rental' Cube	
Attribute	Value
name	Customer Country
foreignKey	sk_cust_country
source	Country
level	
usagePrefix	Customer Country
caption	
visible	<input checked="" type="checkbox"/>

- Tambahkan measure.

4.3.5. Measure

Untuk menambah measure, cukup klik kanan pada cube dan Add Measure. Pengaturan untuk measure adalah sebagai berikut:

Measure for 'Fact Rental' Cube	
Attribute	Value
name	Total Charge
description	
aggregator	sum
column	amount
formatString	#,##0.00
datatype	Numeric
formatter	
caption	
visible	<input checked="" type="checkbox"/>

Schema Workbench memungkinkan pembuatan derived measure atau measure turunan. Contohnya measure untuk menghitung jumlah rental, dapat dihitung dengan melakukan count pada setiap barisnya. Pengaturan untuk measure ini adalah sebagai berikut:

Measure for 'Fact Rental' Cube	
Attribute	Value
name	Total Rental
description	
aggregator	count
column	sk_customer
formatString	
datatype	Integer
formatter	
caption	
visible	<input checked="" type="checkbox"/>

4.8. Latihan

Dari tabel-tabel yang dibuat berdasarkan model multidimensional pada latihan sebelumnya, buatlah skema dengan menggunakan Schema Workbench.

BAB V

PENTAHO BI SERVER

5.1. Tujuan Pembelajaran

- Mahasiswa dapat menggunakan Administrator Console dan User Console
- Mahasiswa dapat melakukan drill up dan drill down data model multidimensional yang telah dibuat menggunakan User Console

5.2. Dasar Teori

Administrator Console merupakan antarmuka yang digunakan oleh administrator untuk melakukan berbagai macam pengaturan terkait dengan penggunaan User Console.

User Console merupakan antarmuka yang memungkinkan pengguna biasa untuk dapat mengakses OLAP.

5.3. Skenario

5.3.1. Pengaturan Environment Variable

Pastikan Path pada Environment Variable komputer telah menyediakan akses ke dalam bin JDK. Kemudian, pastikan juga terdapat variable PENTAHO_JAVA_HOME yang berisi path menuju JDK.

5.3.2. Administrator Console dan User Console

Sebelum menjalankan start-pac.bat, pastikan driver koneksi basis data telah tersedia dalam folder {Pentaho BI Server path}\Pentaho BI

Server\administration-console\jdbc, kemudian aktifkan Administrator Console.

Sebelum menjalankan start-pentaho.bat, pastikan driver koneksi basis data telah tersedia dalam folder F:\Master Portable\Pentaho BI Server\biserver-ce\tomcat\lib, kemudian aktifkan User Console.

Buka Administrator Console menggunakan browser dengan cara mengetikkan localhost/8099. Pergi ke menu Administration > Database Connection untuk membuat koneksi basis data. Buat koneksi baru dengan contoh pengaturan sebagai berikut:

The screenshot shows a configuration dialog for a database connection. The fields and their values are as follows:

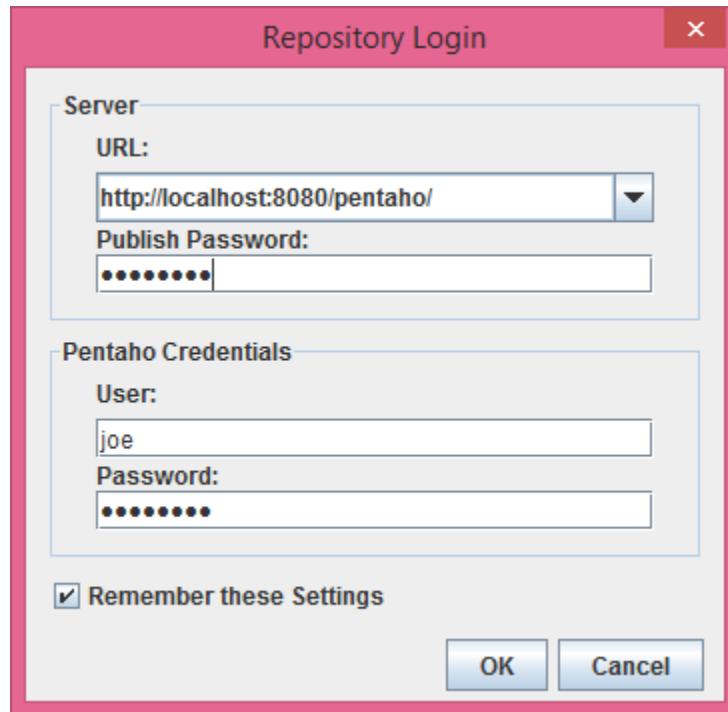
Name:	SakilaOlap
Driver Class:	com.mysql.jdbc.Driver
User Name:	root
Password:
URL:	jdbc:mysql://localhost:3306/sakila.olap

5.3.3. Publish Schema

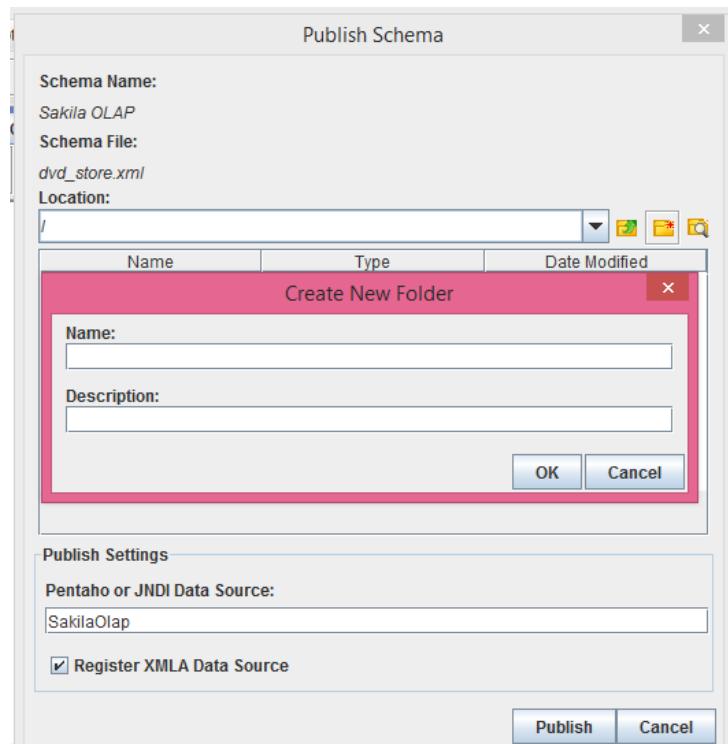
Untuk mem-publish skema yang telah dibuat pada BAB IV, pastikan password untuk mem-publish dalam file {Pentaho BI Server path}\Pentaho BI Server\biserver-ce\pentaho-solutions\system\publisher_config.x

ml sudah terisi.

Buka Schema Workbench, pilih menu File > Publish untuk mem-publish skema. Isikan kolom yang diperlukan, klik ok.

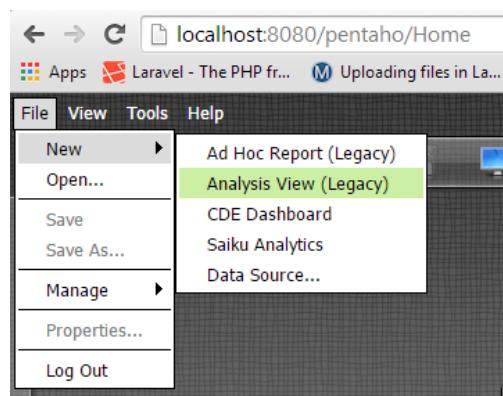


Buatlah file baru untuk menempatkan skema. Pada kolom Pentaho or JNDI Data Source, masukkan nama JNDI yang telah dibuat dalam Administrator Console.

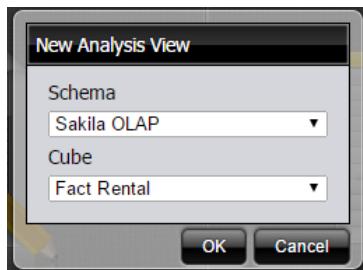


5.3.4. Analisis

Buka User Console pada browser, pilih menu File > New > Analysis View (Legacy).



Pilih skema yang ingin digunakan untuk analisis. Analisis dapat dilakukan dengan cara drill up dan drill down.



5.4. Latihan

Publish skema yang telah dibuat pada latihan BAB IV, kemudian lakukan analisis.

BAB VI

COMMUNITY TOOLS DAN SAIKU

6.1. Tujuan Pembelajaran

6.2. Dasar Teori

Community tools (CTools) merupakan perangkat *open source* yang disediakan sebagai *plug-in* untuk Pentaho *community edition*. Perangkat-perangkat yang tersedia pada *CTools* memiliki fungsi masing-masing sebagai pelengkap *client-side* Pentaho.

6.2.1. *Community Data Browser*

CDB merupakan CTools yang dibuat dengan tujuan untuk menyediakan repository yang terpusat untuk datasource yang tersedia berdasarkan Community Data Access (CDA). Repository ini memberikan kemampuan untuk mengeksplorasi, mengorganisir dan menyimpan data.

6.2.2. Saiku untuk Pentaho

CDB menggunakan Saiku didalamnya sebagai alat untuk mengeksplorasi cube yang telah dibuat. Penggunaan Saiku kurang lebih serupa dengan Analyzer Report pada Pentaho Enterprise Edition.

Sebelum masuk pada penggunaan Analyzer Report, Pentaho telah terlebih dahulu memperkenalkan penggunaan JPivot atau Analysis Report untuk mengeksplorasi cube yang telah dibuat. Dalam perkembangannya, untuk mempermudah pengguna dalam mengeksplorasi cube, dibuatlah Analyzer Report yang lebih interaktif dan user friendly.

6.2.3. *Community Data Access (CDA)*

CDA adalah *plugin* Pentaho yang didesain untuk mengakses data dengan fleksibilitas tinggi. Lahir sebagai perbaikan dari kritik berdasarkan implementasi sebelumnya, CDA memungkinkan pengguna untuk mengakses berbagai macam *data source* Pentaho dan juga:

- Menggabungkan *data source* yang berbeda dengan mengubah XML file;
- Menyimpan *cache* dari *query* sehingga dapat meningkatkan performa;
- Menyediakan data dalam berbagai format (cvs, xls, dan sebagainya) melalui Pentaho *User Console*;
- Mengurutkan dan memungkinkan pemberian halaman pada sisi *data server*.

CDA dapat digunakan di dalam Pentaho *BI Server* atau dikombinasikan dengan CDE/CDF sebagai *plugin* yang bersifat *standalone* tanpa perlu memedulikan detail dari CDA itu sendiri.

6.2.4. *Community Dashboard Framework*

CDF merupakan framework yang diperlukan untuk mengembangkan dashboard didalam pentaho.

6.2.5. *Community Dashboard Editor*

CDE lahir untuk mempermudah pembuatan, perubahan dan proses rendering dari CTools Dashboards. CDE merupakan tool yang powerful dan lengkap, dengan mengkombinasikan front-end dan data source serta modifikasi komponen secara cepat dan efektif.

Teknologi yang berhubungan oleh CDE antara lain adalah CDF, CDA dan CCC.

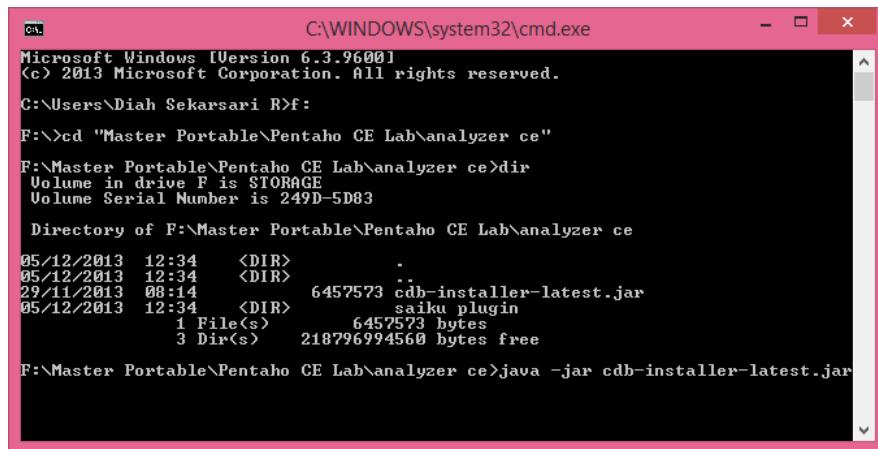
6.2.6. *Community Charting Components (CCC)*

CCC adalah library CTools untuk membuat grafik yang dibangun diatas Protovis. CCC sangat powerful, dapat digunakan secara gratis dan merupakan toolkit visualisasi yang open-source. CCC sudah termasuk dalam instalasi CDE sehingga tidak perlu lagi diinstal terpisah.

6.3. Skenario

6.3.1. Kegiatan 1: Instalasi *CTools*

1. Jalankan file berekstensi *jar* untuk menginstal CDB
 - a. Cara 1: Buka command prompt, masuk ke dalam *path* yang berisi file *jar* CDB (*cdb-installer-latest.jar*) dan jalankan perintah `java -jar cdb-installer-latest.jar` seperti pada contoh dibawah ini:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Diah Sekarsari R>f:
F:>\cd "Master Portable\Pentaho CE Lab\analyzer ce"
F:\Master Portable\Pentaho CE Lab\analyzer ce>dir
Volume in drive F is STORAGE
Volume Serial Number is 249D-5D83

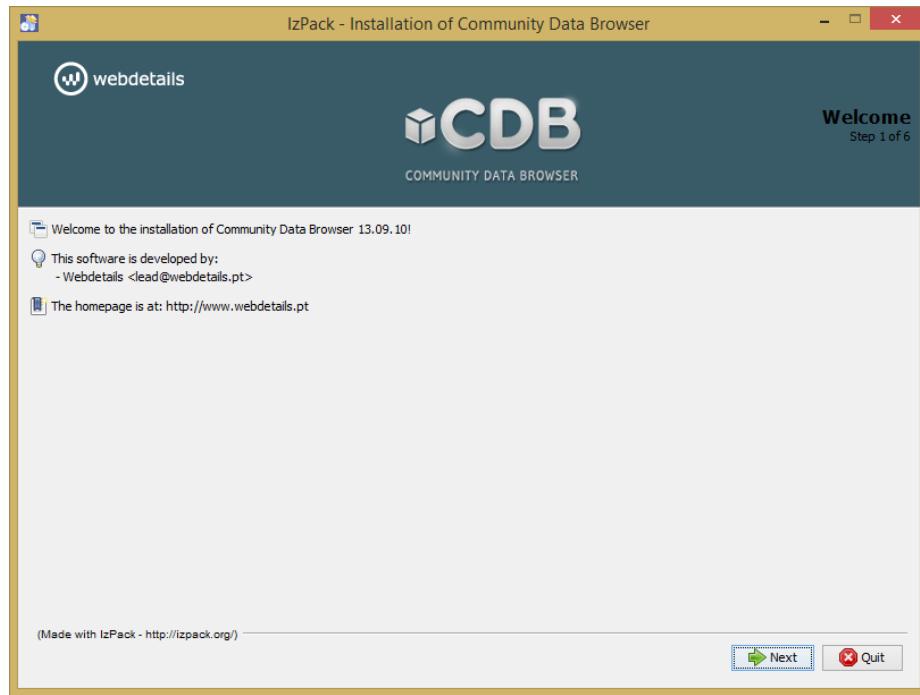
Directory of F:\Master Portable\Pentaho CE Lab\analyzer ce
05/12/2013 12:34 <DIR> .
05/12/2013 12:34 <DIR> ..
29/11/2013 08:14 6457573 cdb-installer-latest.jar
saiku plugin
05/12/2013 12:34 <DIR> 1 File(s) 6457573 bytes
3 Dir(s) 218796994560 bytes free

F:\Master Portable\Pentaho CE Lab\analyzer ce>java -jar cdb-installer-latest.jar
```

- b. Cara 2: Klik dua kali pada file *jar* berikut:

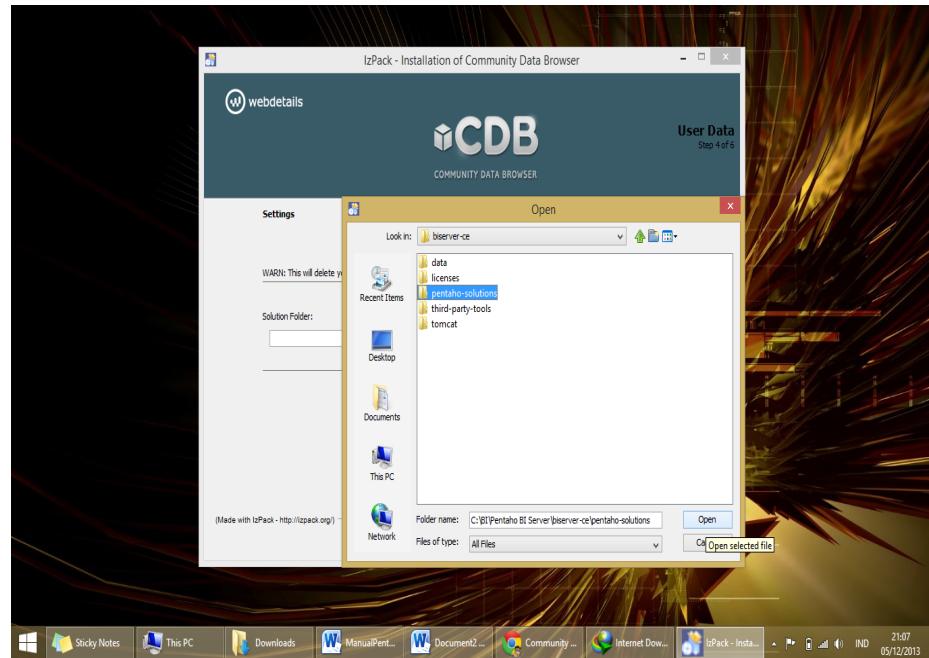


2. Jika file berhasil dijalankan, maka akan muncul kotak dialog seperti dibawah ini:

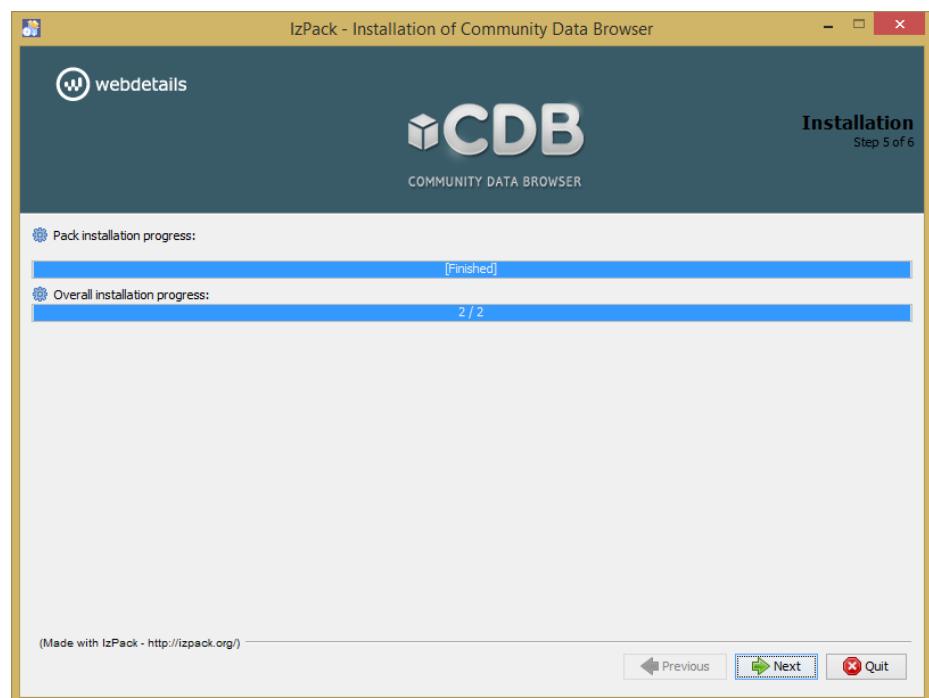


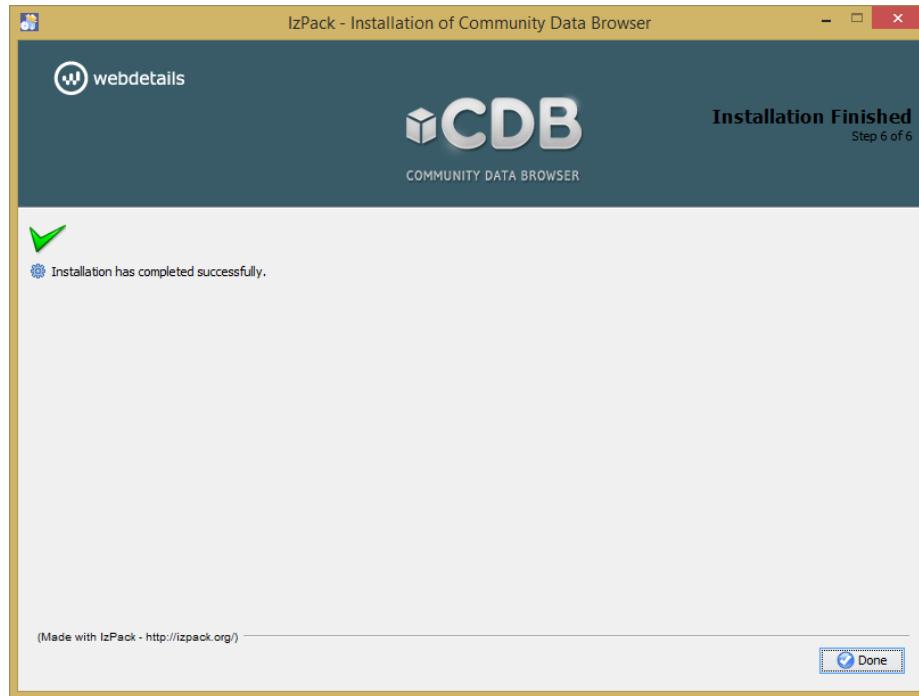
3. Klik tombol [Next] hingga halaman dimana Anda diminta untuk memasukkan *path* dari *Solution* pada Pentaho yang ingin Anda gunakan. Path untuk Solution untuk Pentaho Community Edition 4.X adalah

```
{Pentaho Path}\biserver-ce\pentaho-solutions.
```



4. Apabila instalasi sudah selesai tanpa *warning*, klik tombol [Next] pada kotak dialog seperti dibawah ini dan klik tombol [Done] pada kotak dialog berikutnya.

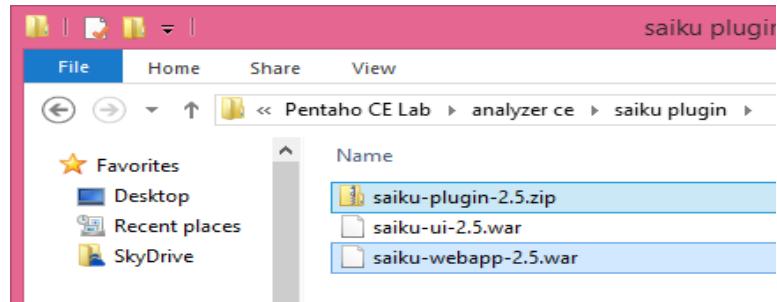




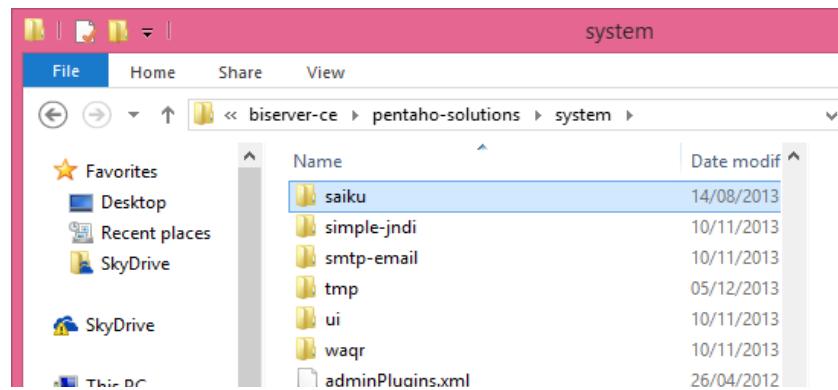
5. Lakukan langkah yang sama seperti pada **instalasi CDB**. File yang digunakan untuk instalasi CDA adalah: `cda-installer-latest.jar`.
6. Lakukan langkah yang sama seperti pada **instalasi CDB**. File yang digunakan untuk instalasi CDF adalah: `cdf-installer-latest.jar`.
7. Lakukan langkah yang sama seperti pada **instalasi CDB**. File yang digunakan untuk instalasi CDE adalah: `cde-installer-latest.jar`.

6.3.2. Kegiatan 2: Instalasi Saiku

1. Yang perlu diperhatikan adalah dua file Saiku berikut:
 - a. `saiku-plugin-2.5.zip`
 - b. `saiku-webapp-2.5.war`

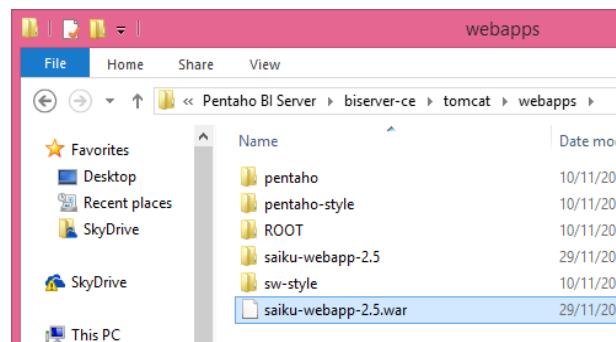


2. Lakukan ekstraksi file saiku-plugin-2.5.zip dan letakkan dalam *path*:



3. Letakkan file saiku-webapp-2.5.war pada *path*:

{Pentaho BI Server Path}\biserver-ce\tomcat\webapps



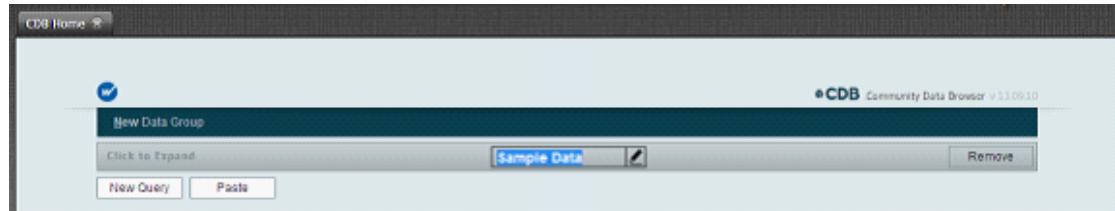
6.3.3. Kegiatan 3: Analisis Menggunakan CDB

1. Jalankan Pentaho BI Server
2. Login ke dalam Pentaho User Console

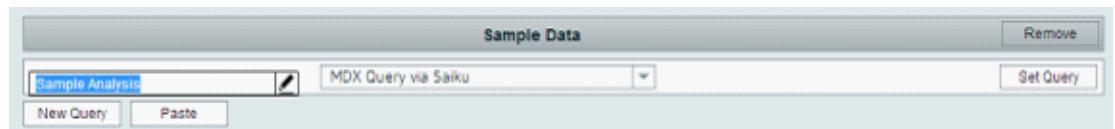
3. Klik pada lambang CDB



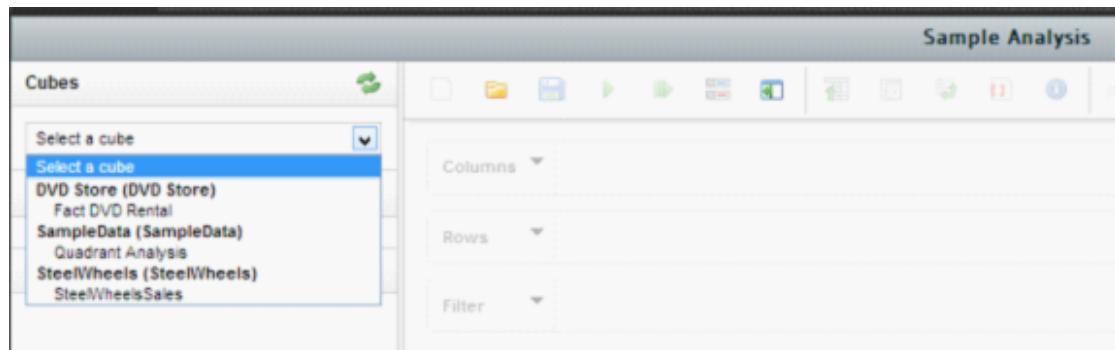
4. Isikan nama pada group yang akan dibuat



5. Klik tombol [New Query] dan masukkan nama query untuk sampel analisa



6. Klik tombol [Set Query] dan pilih cube yang ingin digunakan



7. Pilih cube SteelWheelSales, drag Dimensions>Time>Years pada Columns dan Dimensions>Markets>Territory pada Rows sehingga didapat tabel berikut:

Sample Analysis

Cubes
SteelWheelsSales

Dimensions

- Customers
- Markets
 - (All)
 - Territory
 - Country
 - State Province
 - City
- Order Status
- Product
- Time
 - (All)
 - Years
 - Quarters

Territory	2003	2004	2005
APAC	3.529	5.938	3.411
EMEA	16.711	23.630	9.237
Japan	2.851	1.692	380
NA	13.348	18.157	6.447

8. Tambahkan Measures>Sales pada Rows dan Dimensions>Product>Vendor pada Columns sehingga akan muncul hasil sebagai berikut:

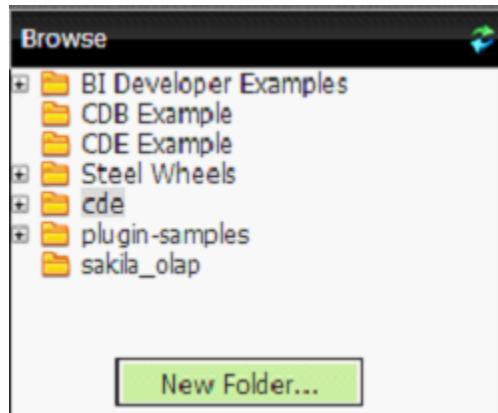
Columns
Years ▾ Sales ▾

Rows
Territory ▾ Vendor ▾

Filter ▾

Territory	Vendor	2003	2004	2005
		Sales	Sales	Sales
APAC	Autoart Studio Design	5.020	2.892	13.199
	Carousel DieCast Legends	2.672	12.354	4.381
	Classic Metal Creations	35.768	29.034	22.486

9. Buat folder baru pada repository: klik kanan > New Folder



10. Simpan hasilnya dengan cara klik simbol save dan simpan pada folder yang telah dibuat sebelumnya.

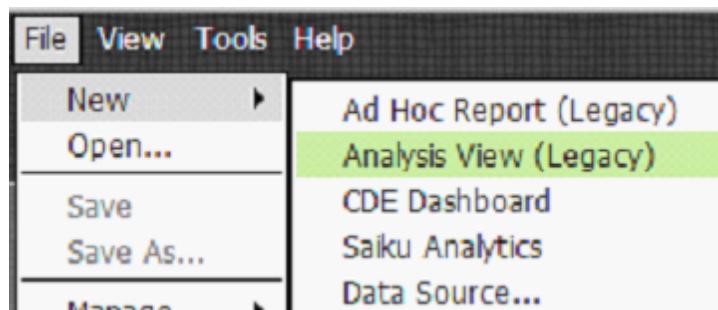
11. Klik tombol

6.4. Latihan

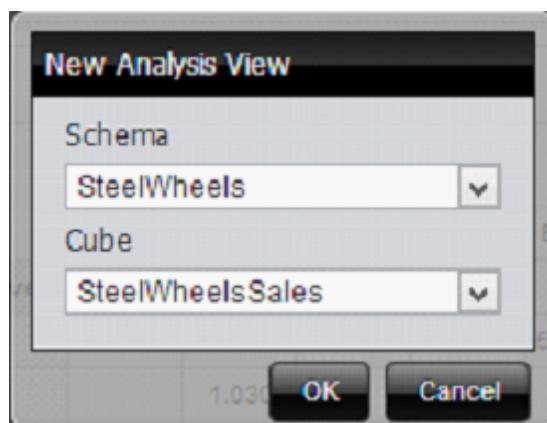
1. Lakukan langkah analisis pada cube yang telah Anda buat
2. Klik simbol pada bagian atas toolbar dan lakukan langkah-langkah berikut:
 - a. Copy MDX query

```
MDX
SELECT
NON EMPTY CrossJoin([Time].[Years].Members, {[Measures].[Sales]}) ON
COLUMNS,
NON EMPTY CrossJoin([Markets].[Territory].Members, [Product].[Vendor].Members) ON ROWS
FROM [SteelWheelsSales]
```

- b. Buka Analysis View



- c. Pilih cube yang telah Anda copy MDX query-nya



- d. Klik tombol **MDX**, kemudian paste MDX query pada MDX Query Editor yang tersedia dan klik tombol [Apply]

MDX Query Editor

The MDX Query Editor window shows the following MDX query:

```
select NON EMPTY CrossJoin([Time].[Years].Members, {[Measures].[Sales]}) ON COLUMNS,  
NON EMPTY CrossJoin([Markets].[Territory].Members, [Product].[Vendor].Members) ON ROWS  
from [SteelWheelsSales]
```

At the bottom right of the editor window are 'Apply' and 'Revert' buttons.

- e. Perhatikan tabel yang terbentuk dibawah Query Editor dan berikan pendapat Anda
3. Lakukan eksplorasi terhadap fitur-fitur yang tersedia pada toolbar dan capture hasilnya, jelaskan bagaimana fitur-fitur tersebut bekerja (tulisakan minimal 3 fitur)

BAB VII

COMMUNITY DASHBOARD EDITOR

7.1. Tujuan Pembelajaran

- Mahasiswa dapat membuat dashboard menggunakan CDE.

7.2. Dasar Teori

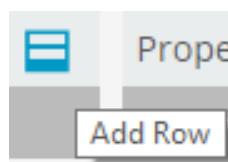
CDE terdiri dari tiga panel, yaitu Layout Panel, Components Panel dan Datasources Panel. Layout Panel bertanggung jawab pada tampilan, Components Panel bertanggung jawab pada penggunaan komponen-komponen dan Datasources Panel bertanggung jawab pada segala sesuatu yang berhubungan dengan basis data. Penjelasan mengenai CDE dapat merujuk pada pembahasan Community Tools pada BAB VI.

7.3. Skenario

Buatlah folder baru pada repository untuk menyimpan dashboard.

7.3.1. Header

- Pada Layout Panel, tambahkan row baru.



- Buat pengaturan row dan tambahkan html pada row untuk memberikan tag <h2> dan diisi title dari dashboard.

Property	Value
Name	dashboard_header
Height	-
BackgroundColor	<input checked="" type="checkbox"/> #cabcaf
Corners	Notch
Text Align	Center
Css Class	-

7.3.2. Parameter

- Tambahkan row baru pada Layout Panel.
- Buat dua kolom untuk row baru. Satu kolom sebagai tempat meletakkan nama parameter, kolom berikutnya untuk meletakkan komponen.

Property	Value
Name	select_param_year
Span size	6
Prepend size	0
Append size	-
Prepend gutter to top	False
Prepend gutter to bottom	False
Right border	False
Big right border	False
Height	-
BackgroundColor	<input type="color"/> #e0e0e0
Corners	Simple
Text Align	-
Css Class	-

- Pada Datasources Panel, pilih SQL Queries > sql over sqlJndi.

Property	Value
Name	select_year_query
Access Level	Public
Jndi	SampleData
Query	SELECT DISTINCT YEAR (...) ...
Parameters	[]
Output Mode	Include
Columns	[]
Calculated Columns	[]
Output Options	[]
Cache Duration	3600
Cache	True

- Isi Query-nya adalah sebagai berikut:

```
1 SELECT DISTINCT YEAR_ID FROM ORDERFACT
```

- Pada Components Panel pilih Select > Select Component untuk menambahkan dropdown list berdasarkan pada query yang telah dibuat.

Property	Value
Name	select_year
Parameter	parameter_year
Listeners	[]
Parameters	[]
Value as id	True
jQuery Plugin	-
Datasource	select_year_query
Values array	[]
HtmlObject	select_param_year

- Masih pada Component Panel, pilih Generic > Simple parameter untuk mendaftarkan parameter_year agar dapat digunakan sebagai parameter. Property value merupakan value default untuk parameter_year.

Property	Value
Name	parameter_year
Property value	2004
Bookmarkable	False

7.3.3. Objects

- Pada Layout Panel, buat row baru.
- Buat dua kolom untuk memasukkan dua object. Beri span size 12 pada kedua kolom.
- Pada kolom pertama, buat dua row. Row pertama untuk judul, row kedua untuk pie chart, beri nama piechart_object.
- Buat datasource baru.

Property	Value
Name	piechart_query
Access Level	Public
Jndi	SampleData
Query	SELECT C.TERRIT (...)
	...

- Masukkan Query:

```

1  SELECT
2      C.TERRITORY,
3      SUM(O.TOTALPRICE)
4  FROM
5      CUSTOMER_W_TER C
6      INNER JOIN ORDERFACT O ON C.CUSTOMERNUMBER = O.CUSTOMERNUMBER
7  WHERE
8      O.YEAR_ID = 2003
9  GROUP BY
10     C.TERRITORY

```

- Pada Components Panel, pilih Charts > CCC Pie Chart.

Property	Value
Name	piechart
Title	-
Listeners	['parameter_year']
Custom Chart Script	<input type="button" value="..."/>
Parameters	[["year","parameter_year"]]
Datasource	piechart_query
Width	450
Height	450
HtmlObject	piechart_object

- Pada kolom Listeners, pilih parameter_year.
- Pada kolom Parameters, argumen year dengan value didapat dari parameter_year.
- Pergi ke Datasource Panel, masukkan Parameter dengan nama sesuai argumen yang telah dibuat dan tipe data integer.

Property	Value
Name	piechart_query
Access Level	Public
Jndi	SampleData
Query	SELECT C.TERRIT (...) <input type="button" value="..."/>
Parameters	[["year","","Integer","",""]]
Output Mode	Include
Columns	<input type="button" value=""/>
Calculated Columns	<input type="button" value=""/>
Output Options	<input type="button" value=""/>
Cache Duration	3600
Cache	True

- Ubah Query dengan memasukkan \${year} dimana nilai tahun akan diambil dari nilai pada dropdown list yang telah dibuat sebelumnya.

```

1  SELECT
2      C.TERRITORY,
3      SUM(O.TOTALPRICE)
4  FROM
5      CUSTOMER_W_TER C
6      INNER JOIN ORDERFACT O ON C.CUSTOMERNUMBER = O.CUSTOMERNUMBER
7  WHERE
8      O.YEAR_ID = ${year}
9  GROUP BY
10     C.TERRITORY

```

- Preview dashboard dan lihat bagaimana hasilnya.

Contoh hasil akhir dashboard adalah sebagai berikut:



7.4. Latihan

1. Lakukan hal yang sama seperti skenario pembuatan pie chart. Buat bar chart pada kolom sebelah kanan Objects. Query untuk bar chart ditentukan sebagai berikut:

```

1  SELECT
2      P.PRODUCTLINE,
3      O.YEAR_ID AS YEAR,
4      SUM(O.TOTALPRICE) AS TOTAL
5  FROM
6      PRODUCTS P
7      INNER JOIN ORDERFACT O ON P.PRODUCTCODE = O.PRODUCTCODE
8      INNER JOIN CUSTOMER_W_TER C ON O.CUSTOMERNUMBER = C.CUSTOMERNUMBER
9  WHERE
10     O.YEAR_ID IN(${year}-1,${year})
11     AND C.TERRITORY = ${territory}
12  GROUP BY
13      P.PRODUCTLINE,
14      O.YEAR_ID

```

2. Dari seluruh proses latihan yang telah dilakukan, buatlah dashboard untuk model multidimensional Anda.