

Data Warehouse with Kettle - Open Source ETL



DΦLab

Daftar Isi

Bab 1.	Pendahuluan.....	6
Bab 2.	Pentaho - Open Source BI Suite.....	7
2.1	Pentaho Corporation	7
2.2	Pentaho – Community Open Source BI Suite	7
2.3	Skema Hubungan Aplikasi-aplikasi Pentaho.....	11
Bab 3.	Terminologi Dasar	12
3.1	Online Transaction Processing (OLTP)	12
3.2	Online Analytical Processing (OLAP).....	12
3.3	Data Warehouse.....	13
3.4	Data Mart.....	13
3.5	ETL (Extract, Transform, Load)	14
Bab 4.	Instalasi Aplikasi dan Contoh Data.....	15
4.1	Java Development Kit / Java Runtime Environment	15
4.2	MySQL – Paket XAMPP	16
4.3	7Zip	17
4.4	SQLYog – Community Edition	17
4.5	Pentaho Data Integration / Kettle.....	21
4.6	Import Database PHI-Minimart.....	21
Bagian 1		24
Pentaho Data Integration / Kettle		24
Bab 5.	Pentaho Data Integration (Kettle).....	25
5.1	Pendahuluan.....	25
5.2	Komponen Aplikasi Kettle.....	25
5.3	Arsitektur Kettle.....	26
5.4	Model Konseptual Job / Transformation	27
5.5	Organisasi Folder Kettle.....	28
Bab 6.	Spoon.....	29
6.1	Praktek: Menjalankan Spoon	29
6.2	Lingkungan Kerja Spoon	30
Bab 7.	Konsep dan Komponen Kettle	31

7.1	Koneksi Database.....	31
7.2	Step.....	31
7.3	Hop.....	31
7.4	Job.....	33
7.5	Praktek: Job Pertama – Message Box.....	34
7.6	Praktek: Message Box dengan Kondisi.....	36
7.7	Praktek: Merubah Kondisi Hop pada Job.....	39
7.8	Praktek: Folder Auto Creation dengan Menggunakan Schedule.....	39
7.9	Transformation.....	41
7.10	Row.....	41
7.11	Variable.....	42
7.12	Log	42
7.13	Praktek: Transformation Untuk Membaca 1 File Excel.....	42
7.14	Praktek: Merubah Meta Data.....	47
7.15	Praktek: Preview Data.....	50
7.16	Praktek: Membuat Koneksi Database ke MySQL.....	51
7.17	Praktek: Menulis Data ke Tabel MySQL.....	52
7.18	Praktek: Melihat Log.....	54
7.19	Praktek: Normalisasi Data.....	55
7.20	Email Step.....	56
7.21	Praktek: Menghasilkan dan Mengirimkan File Excel via Email.....	56
Bab 8.	Repositori Kettle.....	57
8.1	Penyimpanan Objek-objek Kettle.....	57
8.2	Praktek: Buat Database Repository.....	57
8.3	Praktek: Menyimpan Transformasi pada Repositori KettleRepo	59
8.4	Praktek: Eksplorasi Repositori.....	59
Bagian 2.....		60
Multi Dimensional Modelling.....		60
Bab 9.	Terminologi dan Konsep Dasar.....	61
9.1	Apa itu Multidimensional Modelling ?	61
9.2	OLAP : Cube, Dimension, Measure dan Member.....	61
9.3	Fact and Dimension Tables	62
9.4	Skema Bintang (<i>Star Schema</i>).....	62

9.5 Skema Butir Salju (<i>Snowflake Schema</i>).....	63
9.6 Surrogate Key.....	63
Bab 10. Multi Dimensional Modelling dengan Kettle.....	64
10.1 Petunjuk Umum Pembuatan Tabel Dimensi PHI-Minimart.....	64
10.2 Praktek: Membuat Database PHI_DW.....	64
10.3 Praktek: Membuat Koneksi database PHI_DW pada Spoon	64
10.4 Praktek: Dimensi Produk.....	65
10.5 Praktek: Dimensi Cabang	78
10.6 Praktek: Dimensi Karyawan	78
10.7 Praktek: Dimensi Waktu	80
10.8 Praktek: Fact Penjualan PHI-Minimart.....	82
10.9 Praktek: Menggabungkan Semua File Transformation dengan Job	83
10.10 Praktek: Penggunaan Mondrian sebagai OLAP Engine PHI-Minimart.....	84
Bagian 3.....	86
Staging Database	86
10.11 Pendahuluan.....	87
10.12 Latar Belakang : Physical I/O versus In-Memory Processing	87
10.13 Change Data Capture (CDC)	87
10.14 Demo: Change Data Capture pada SQL Server 2008.....	87
10.15 Praktek: Baca file dari 3 sumber	87
10.16 Praktek: Menggunakan Staging Database	88
Bab 11. Advance Controls.....	89
11.1 Environment Variable	89
11.2 Shared Object	90
11.3 Error Handling.....	92
11.4 Praktek: Error Handling pada Modified Javascript Value	92
11.5 Praktek: Setting Variable pada file kettle.properties	93
11.6 Praktek: Notifikasi Error via Email.....	93
11.7 Diskusi Kasus & Praktek: Changed Data Capture (CDC)	93
Bab 12. Slowly Changing Dimension.....	94
12.1 Apa itu Slowly Changing Dimension ?	94
12.2 Type 1 SCD.....	94
12.3 Type 2 SCD.....	94

12.4	Type 3 SCD.....	95
12.5	SCD Step : Dimension Lookup / Update	95
12.6	Praktek: SCD untuk Dimensi Produk.....	96
Bab 13.	Otomatisasi.....	99
13.1	Otomatisasi Kettle.....	99
13.2	Praktek: Script dan Penjadwalan Eksekusi.....	100

Bab 1. Pendahuluan

Dunia korporasi saat ini menghadapi permasalahan yang hampir sama, yaitu membengkaknya data akibat keberhasilan implementasi berbagai sistem komputer. Otomatisasi berjalan dengan baik, namun berbagai laporan yang ingin dihasilkan sebagai output dari sistem tersebut sebagian besar tidak tercapai.

Permasalahannya adalah 85 persen waktu yang diperlukan adalah mempersiapkan data (data preparation). Database yang dirancang untuk sistem entri ternyata sangat buruk digunakan menghasilkan berbagai laporan analisa. Berbagai solusi telah ditawarkan oleh para praktisi IT, dan pendekatan dengan membangun suatu data warehouse adalah yang terbaik.

Untuk membangun data warehouse yang baik diperlukan tool yang mampu mengambil, mengolah dan menyimpan data pada berbagai format dengan baik. Tool ini biasa disebut dengan ETL, singkatan dari Extract, Transform and Load.

Salah satu produk ETL adalah Kettle atau Pentaho Data Integration, suatu produk yang bersifat free open source ETL dan dapat berjalan di lingkungan multi-platform. Kettle memiliki lebih dari 140 built-in modules yang dapat digunakan hampir semua format data terpopuler dan biasa digunakan di dunia enterprise saat ini.

Dengan kemampuannya yang sangat baik dan didukung oleh komunitas open source, Kettle menjadi aplikasi favorit diantara para pengembang data warehouse.

Kursus berdurasi lima hari ini membahas pembangunan data warehouse dengan Kettle. Selama lima hari Anda akan mempelajari konsep dasar tentang data warehouse, Multidimensional Modelling, and penggunaan dari Kettle itu sendiri.

Bab 2. Pentaho - Open Source BI Suite

2.1 Pentaho Corporation

Pentaho adalah sebuah perusahaan *commercial open source BI* yang berpusat di Orlando, Amerika Serikat. Pentaho melakukan banyak akuisisi terhadap proyek *open source* yang telah stabil dan populer dan mengadopsi model pengembangannya, menjadikan produk-produk Pentaho banyak diterima di kalangan komunitas dan secara bertahap mulai populer di lingkungan *enterprise*.

Pentaho mendapatkan keuntungan dari penjualan produk Pentaho Enterprise Edition atau produk versi *non community*. Versi ini berbeda dari sisi *support*, jumlah produk, dan harga lisensi.

2.2 Pentaho – Community Open Source BI Suite

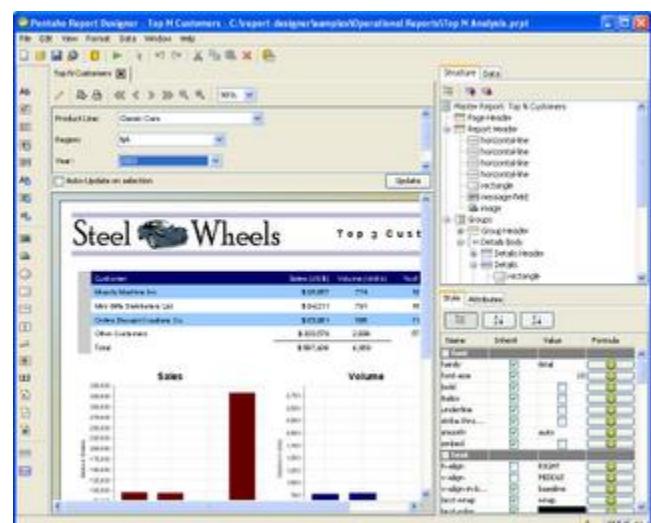
Pentaho community adalah kumpulan aplikasi Business Intelligence (BI) yang bersifat *free open source software (FOSS)* dan berjalan di atas platform Java.

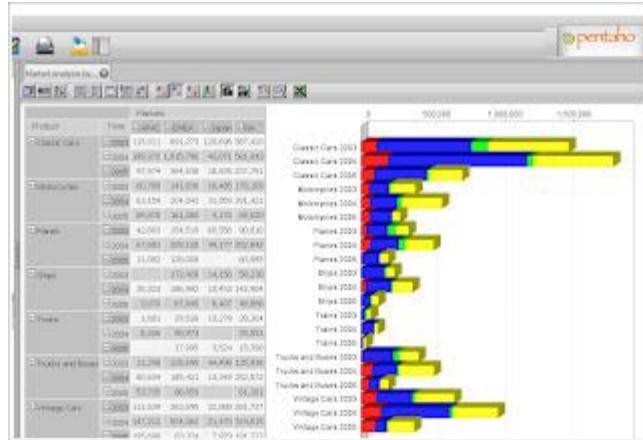
Pentaho community dikembangkan bersama oleh Pentaho Corporation dan komunitas yang berasal dari seluruh dunia.

Selain sifatnya gratis dan adopsi yang semakin hari semakin luas, dukungan Pentaho bisa didapatkan dalam bentuk *Service Level Agreement (SLA)* dan dipaketkan dalam versi Enterprise Edition yang sifatnya annual subscription atau perlu kontrak tahunan. Selain itu jika Anda tetap menggunakan *community edition* yang gratis, maka bisa mendapatkan *support* dari banyak system integrator Pentaho di seluruh dunia.

2.2.1 Pentaho Reporting

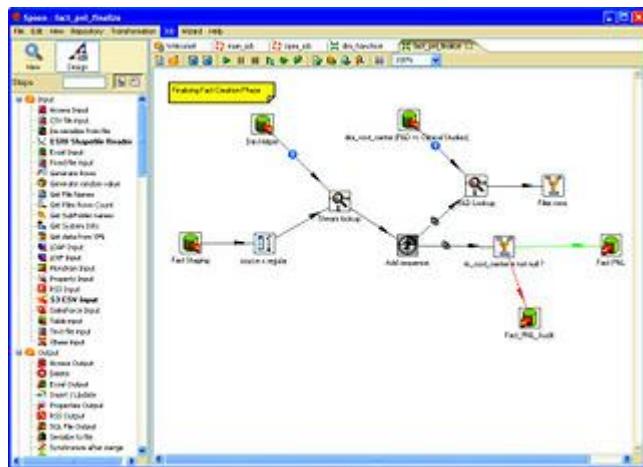
- Aplikasi perancangan *reporting* Pentaho dari sumber data relasional maupun OLAP.
- Dilengkapi designer yang intuitif dan koleksi chart yang lengkap untuk keperluan analisa.
- Output dalam format Format PDF, Excel, HTML dan CSV.
- Engine reporting yang bisa di-embed di aplikasi Java.
- Multi Platform.





2.2.2 Pentaho Analysis / Mondrian OLAP Engine

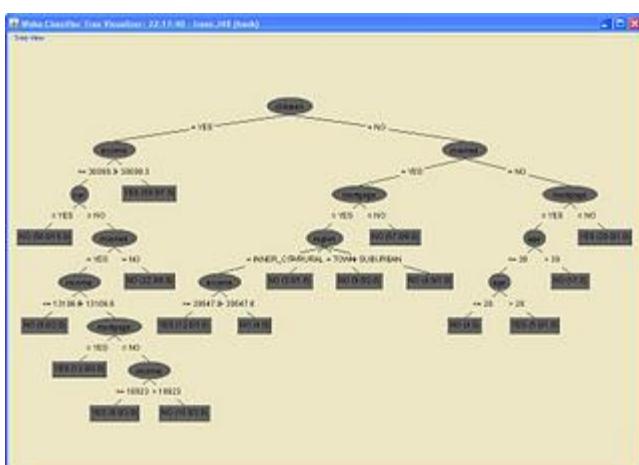
- OLAP engine *open source* terpopuler dan digunakan di berbagai produk BI *open source* (Pentaho dan Jasper).
- Multi Platform.
- Mondrian merupakan tipe ROLAP (Relational OLAP) dimana semua perintah data query diterjemahkan via SQL dan ditujukan kepada datamart.
- Mendukung datamart dengan rancangan multi dimensional Star Schema maupun Snowflake Schema.
- Mendukung query MDX (Multidimensional Expression) sebagai standar industri.
- Didukung oleh beberapa aplikasi interface baik web based maupun desktop seperti JPivot, Pentaho Reporting, Pentaho Analysis Tools, dan Pentaho Analyzer.
- Semua interface di atas memiliki kemampuan drill down / roll up serta drill through untuk melihat detil penyusun sel-sel nilai analisis.



2.2.3 Pentaho Data Integration / Kettle

- Utilitas ETL (Extract, Transform and Load) *open source* paling populer.
- Designer GUI yang intuitif dan sangat mudah digunakan.
- Multi Platform.

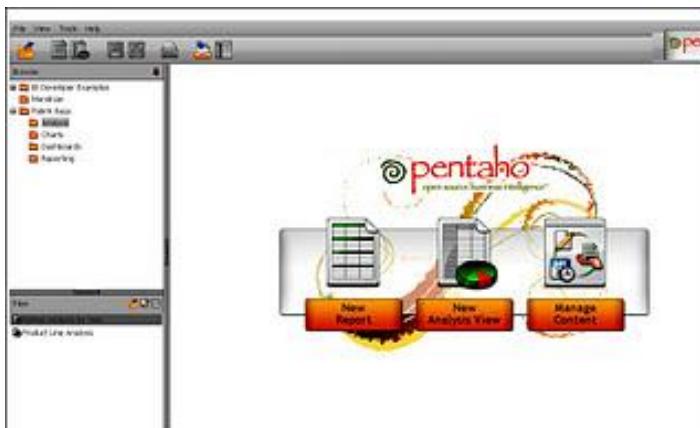
- Script ETL dapat disimpan dalam bentuk filesystem maupun repository.
- Mendukung multi pipelining sehingga load balance maupun optimasi pekerjaan data warehouse dapat dilakukan dengan mudah.
- Mendukung clustering (master-slave) engine ETL



- Terdiri atas lebih dari 200 step yang mencakup job (workflow kontrol) dan transformation (data workflow).
- Mendukung Apache Virtual Filesystem (Apache VFS) sehingga filesystem seperti HTTP Webdav, FTP, SFTP, dan lain sebagainya dapat dengan mudah diakses dengan konfigurasi yang minimal.

2.2.4 Pentaho Data Mining / Weka

- Utilitas *data mining* yang sangat populer dan digunakan untuk *predictive analysis*.
- Mendukung data sumber berformat ARFF (*Attribute-Relation File Format*), XRRF (*XML attribute Relation File Format*), CSV (*Comma Separated Value*), dan tabel-tabel dari database relasional.

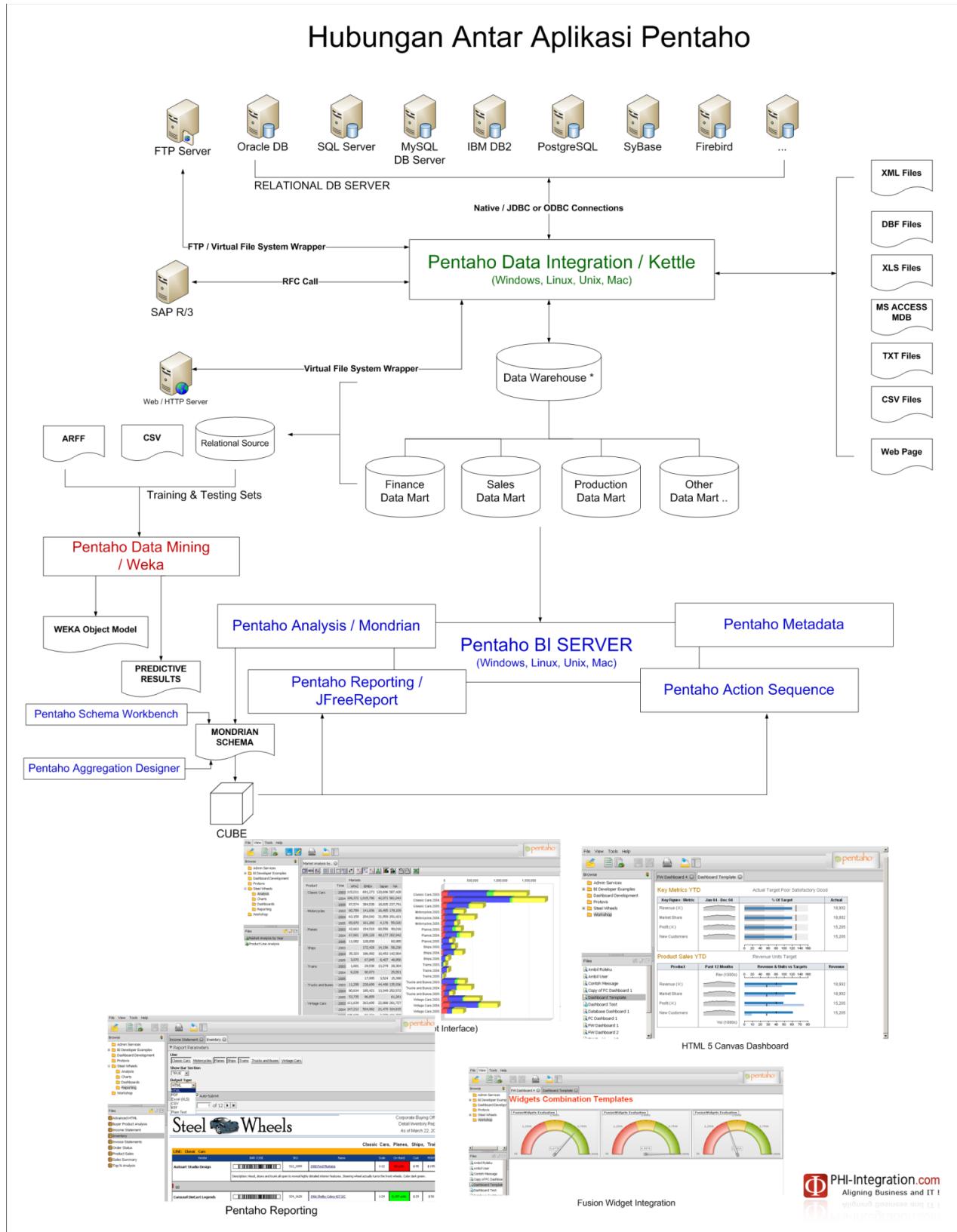


- Mendukung standar PMML (Predictive Model Markup Language).
- Mendukung data nominal dan angka diskrit maupun kontinu.
- Memiliki kumpulan algoritma yang cukup banyak untuk operasi dasar data mining seperti J48, M5P, SimpleCart, SimpleKMeans, Apriori, dan lain-lain.
- Utilitas grafis yang membantu pengolahan dan pemahaman data mulai dari *preprocessing*, *data visualization - plot* maupun *curve*, *tree visualization*, dan lain-lain.

2.2.5 Pentaho BI Server / Platform

- Server BI yang berjalan sebagai web application portal yang terdiri dari layanan web service, workflow pada space JVM (Java Virtual Machine), dan sebagai user interface untuk laporan operasional maupun analisis.
- Multi Platform.
- Workflow berupa integrasi dari produk Pentaho yang telah disebutkan sebelumnya (Pentaho Data Integration, Pentaho Reporting, dan Pentaho Analysis) dalam bentuk solution.
- *Scheduler* berjalan di atas platform BI ini untuk melakukan proses *batch*, misalkan mengirim laporan dalam bentuk PDF ataupun Excel secara periodik.
- Script JSP (Java Server Pages) dapat dengan mudah diintegrasikan ke dalam platform.
- User dapat memiliki space sendiri untuk menyimpan report dan tipe solution lainnya.
- Ad hoc report yang bisa digunakan untuk membuat rancangan report on the fly tanpa keterlibatan IT.

2.3 Skema Hubungan Aplikasi-aplikasi Pentaho



Bab 3. Terminologi Dasar

3.1 Online Transaction Processing (OLTP)

Sistem operasional bisnis biasanya berfokus pada mencatat transaksi-transaksi yang terjadi pada berbagai titik operasi bisnis, oleh karena itu mereka dikarakteristikkan sebagai sistem OLTP (*Online Transactional Processing*). Sebuah database OLTP biasanya mengandung data-data yang spesifik terhadap suatu proses bisnis seperti penjualan, produksi, finansial, dan lain-lain.

Beban kerja sistem informasi OLTP difokuskan pada memasukkan data baru, melakukan perubahan dan penghapusan data secara *real time*.

Sistem OLTP sangat *mission critical* artinya tidak boleh ada gangguan dalam sistem ini atau operasional tidak bisa berjalan dengan baik.

Beberapa contoh sistem informasi yang bisa dikategorikan sebagai OLTP adalah:

- **ERP (Enterprise Resource Planning)**

Beberapa contoh produk ERP: SAP, Compiere/Adempiere, Microsoft Dynamics, dan lain-lain.

- **HRM (Human Resource Management)**

Beberapa contoh produk HRM: OrangeHRM, PeopleSoft, SimpleHRM, dan lain-lain.

- **CRM (Customer Relationship Management)**

Beberapa contoh produk CRM: Microsoft CRM, SugarCRM, Siebel CRM, dan lain-lain.

3.2 Online Analytical Processing (OLAP)

Online Analytical Processing atau OLAP adalah sistem yang dirancang khusus untuk menghasilkan laporan analisa yang fleksibel, kompleks dan dapat dihasilkan dalam waktu yang cepat.

Database OLAP sangat berbeda dengan OLTP dari sisi beban kerja dimana OLAP dirancang dan difokuskan pada kecepatan pembacaan data sedangkan OLTP pada kecepatan perekaman dan perubahan data.

Umumnya database OLAP dihasilkan melalui suatu proses *batch* dan biasanya dilakukan dalam periode tertentu (tiap tengah malam, tiap minggu, dsbnya).

Database OLAP biasanya juga telah merupakan database yang sudah diperkaya dari berbagai sumber data OLTP, dan biasanya merupakan suatu *data warehouse*.

3.3 Data Warehouse

Data Warehouse adalah suatu database khusus yang digunakan sebagai “gudang data” atau data yang telah terkonsolidasi dari sumber-sumber data berbagai sistem informasi yang ada pada suatu organisasi / perusahaan.

Misalkan jika kita memiliki sistem pembelian / *purchasing*, sumber daya manusia (*human resources*), penjualan (*sales*), inventori, dan lain-lain. Maka data warehouse adalah hasil konsolidasi atau penggabungan data-data dari berbagai sistem OLTP tersebut dalam satu kesatuan sehingga kaitan antara satu dengan yang lainnya menjadi jelas dan kuat.

Dengan adanya konsolidasi tersebut, diharapkan banyak informasi berharga yang bisa didapatkan sehingga menciptakan nilai strategik yang tinggi dalam kaitan pengambilan keputusan.

Struktur data warehouse tentunya perlu dioptimalkan untuk berbagai kepentingan sistem pelaporan terutama yang bersifat analisa menyeluruh. Dan dengan kaitan perkembangan OLAP yang semakin penting, data warehouse saat ini lebih banyak mengadopsi pemodelan yang cocok untuk dikonsumsi sistem OLAP dengan baik.

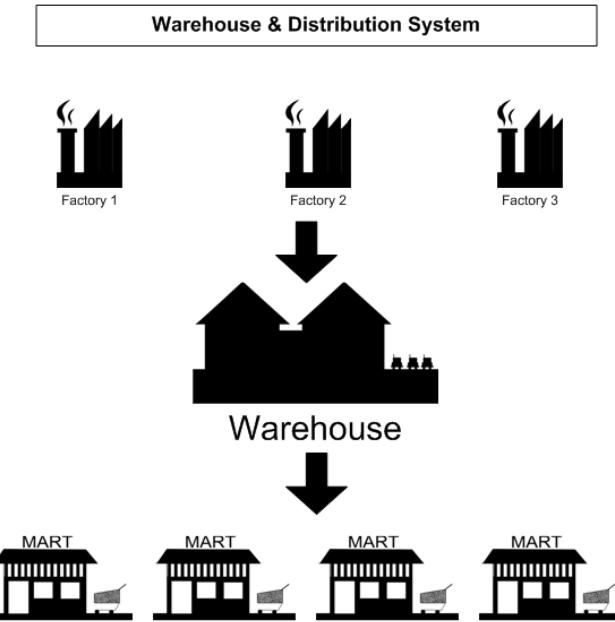
3.4 Data Mart

Setelah data warehouse terbentuk, informasi yang bisa diambil tentunya jauh lebih luas dan lengkap. Pada data warehouse kemungkinan besar terjadi proses pembersihan data (*data cleansing*) dan pengayaan data (*content enrichment*).

Sebagai contoh, pada sistem penjualan kita hanya mengetahui siapa saja tenaga penjual (*sales person*) dengan kinerja yang baik. Tetapi kita tidak dapat mengetahui latar belakang kenapa beberapa sales performanya baik dan yang lainnya tidak. Dan dengan integrasi data dari sistem SDM yang lebih lengkap profilnya, kita dapat memetakan kinerja dengan latar belakang pendidikan atau lebih luasnya ke data demografis dan hasil uji saringan masuk.

Dari data warehouse ini sering dibutuhkan subset data / informasi yang hanya dibutuhkan oleh divisi-divisi tertentu. Dan kadang jauh lebih baik apabila secara fisik database ini terpisah sendiri untuk divisi tersebut. Subset dari data warehouse ini disebut dengan istilah **Data Mart**.

Ilustrasi atau analogi akan hubungan sumber data (produsen / pabrik data), data warehouse dan data mart sangat mirip akan sistem distribusi seperti terlihat pada gambar di bawah ini.



3.5 ETL (Extract, Transform, Load)

Aplikasi ETL adalah aplikasi yang dirancang khusus untuk melakukan 3 proses pengolahan data:

- **Extract:** melakukan mengambil / ekstraksi data.
- **Transform:** melakukan transformasi struktur data agar dapat diolah lebih lanjut.
- **Load:** melakukan penyimpanan data sesuai format database yang digunakan.

Aplikasi ETL sangat membantu sekali dalam merancang dan membentuk suatu *data warehouse*.

Beberapa kemampuan yang harus dimiliki oleh suatu ETL adalah sebagai berikut:

- Dapat membaca dan mengirim data dari dan ke berbagai sumber seperti file teks, file Excel, table-table database relasional, web services, dan lain sebagainya.
- Mampu melakukan manipulasi (transformasi) data dari yang sederhana sampai ke tahap rumit. Beberapa pengolahan data yang harus bisa dilakukan seperti melakukan normalisasi dan denormalisasi data, memecah kolom, melakukan perhitungan yang terkait tanggal dan jam, dan lain-lain.
- Script dapat dirancang dan diimplementasikan dengan tingkat produktivitas yang tinggi.
- Menghasilkan informasi meta data pada setiap perjalanan transformasi.
- Memiliki audit log yang baik.
- Dapat ditingkatkan performanya dengan scale up dan scale out.

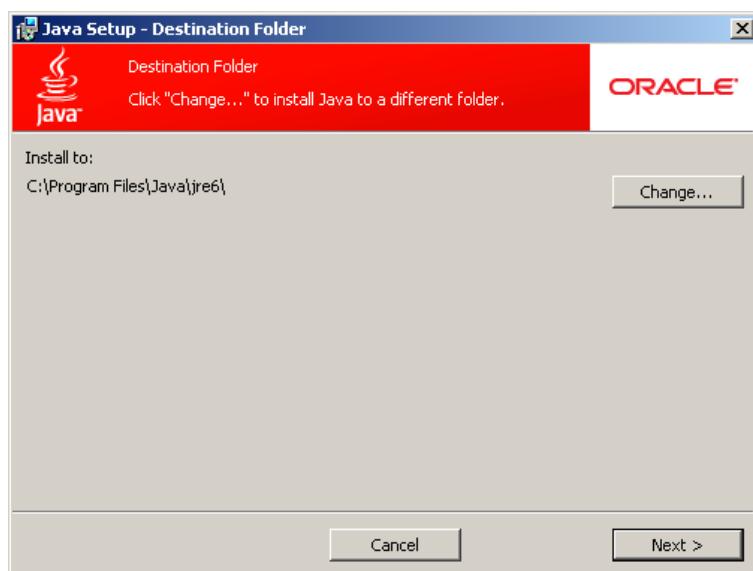
Walaupun membangun solusi ETL dengan programming biasa seperti *coding* dengan bahasa Java atau .NET sangat dimungkinkan, namun tidak disarankan karena produktivitasnya yang cukup rendah dan berakibat ke pemborosan waktu dan biaya.

Berbagai utilitas ETL telah tersedia di pasaran dan menunjukkan kinerja yang baik contohnya adalah Informatica, Data Transformation Service (DTS), Microsoft SQL Server Integration Service (SSIS), Pentaho Data Integration (Kettle), dan lain sebagainya.

Bab 4. Instalasi Aplikasi dan Contoh Data

4.1 Java Development Kit / Java Runtime Environment

Java adalah platform yang digunakan oleh seluruh aplikasi Pentaho dalam menjalankan fungsinya. Untuk instalasi Java ini jalankan file setup **java development kit (jdk)** versi 6, 7 atau 8 yang dapat Anda download dari website Oracle, dan kemudian tentukan konfigurasi instalasi yang Anda inginkan atau sesuai petunjuk dari instruktur untuk menyelesaikan setup.



Setelah selesai, lakukan pengujian instalasi melalui command prompt dengan langkah-langkah berikut :

- Pilih menu **Start | Run**, ketik **cmd**, klik tombol **OK**
- Ketik perintah **java -version**
- Jika instalasi berhasil dengan baik maka hasil outputnya akan tampak seperti pada gambar berikut.

```
C:\>java -version
java version "1.6.0_30"
Java(TM) SE Runtime Environment (build 1.6.0_30-b12)
Java HotSpot(TM) Client VM (build 20.5-b03, mixed mode, sharing)
C:\>
```

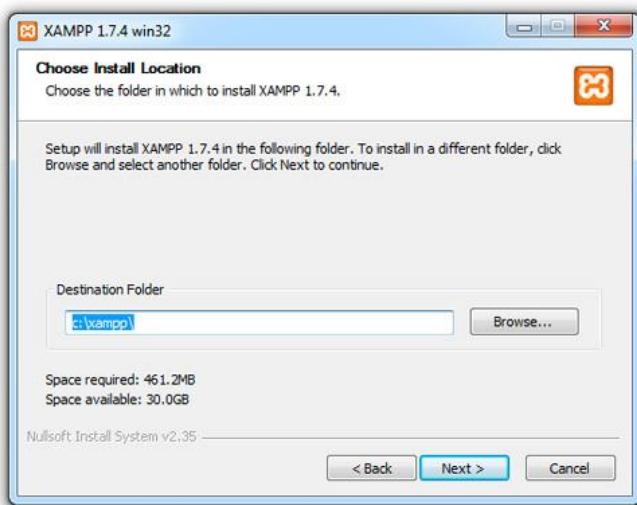
- Selesai.

4.2 MySQL – Paket XAMPP

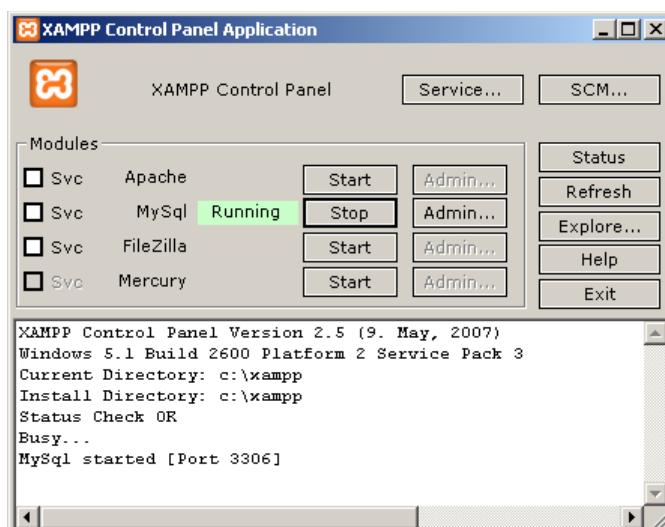
MySQL adalah produk database *open source* yang bisa dibilang paling populer saat ini. Penggunaan MySQL yang cukup mudah, kaya akan berbagai tools penunjang, dan integrasi yang baik dengan engine web PHP menjadikan menjadikan database ini sangat luas digunakan.

Untuk instalasi MySQL pada pelatihan ini, digunakan paket XAMPP yang di dalamnya memiliki database MySQL, web server Apache, engine PHP, dan aplikasi PHPMyAdmin yang dapat digunakan untuk melakukan administrasi database MySQL. XAMPP juga dilengkapi dengan aplikasi *control panel*.

Jalankan file setup **xampp-win32-1.7.7-VC9-installer.exe** yang dapat Anda download di <https://tinyurl.com/dqlab-xampp-download>, dan kemudian tentukan konfigurasi instalasi yang sesuai dengan sistem operasi Anda.



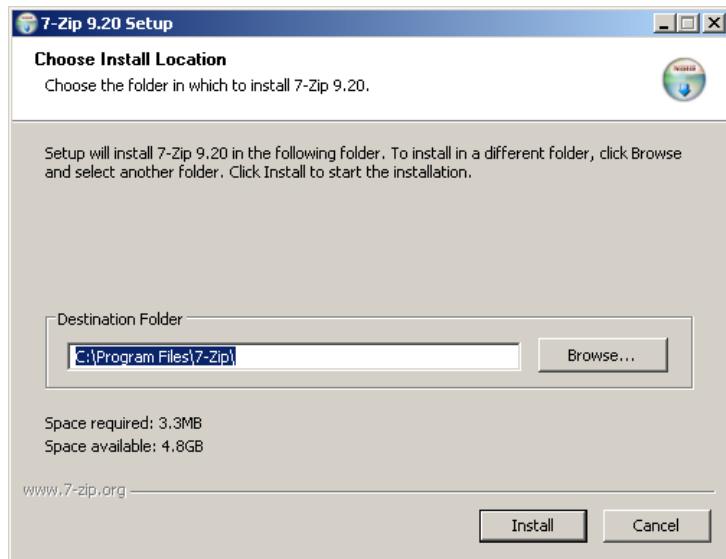
Setelah instalasi, jalankan XAMPP Control Panel seperti tampak pada gambar di bawah dimana Anda bisa melakukan administrasi seperti merubah konfigurasi, menyalakan dan mematikan server-server yang terinstalasi bersama paket XAMPP.



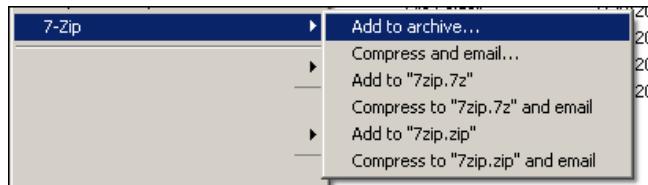
4.3 7Zip

7Zip adalah aplikasi *archiver free dan open source* yang dapat melakukan kompresi ataupun membuka berbagai file kompresi seperti zip, rar, tar.gz, dan lain-lain dengan performa yang cukup baik.

Untuk melakukan instalasi, jalankan file setup **7z920.exe** yang dapat Anda download di url <https://tinyurl.com/7zip-download-dqlab>.



Setelah selesai instalasi Anda dapat membuka file kompresi ataupun melakukan kompresi dari context menu pada aplikasi Windows Explorer.

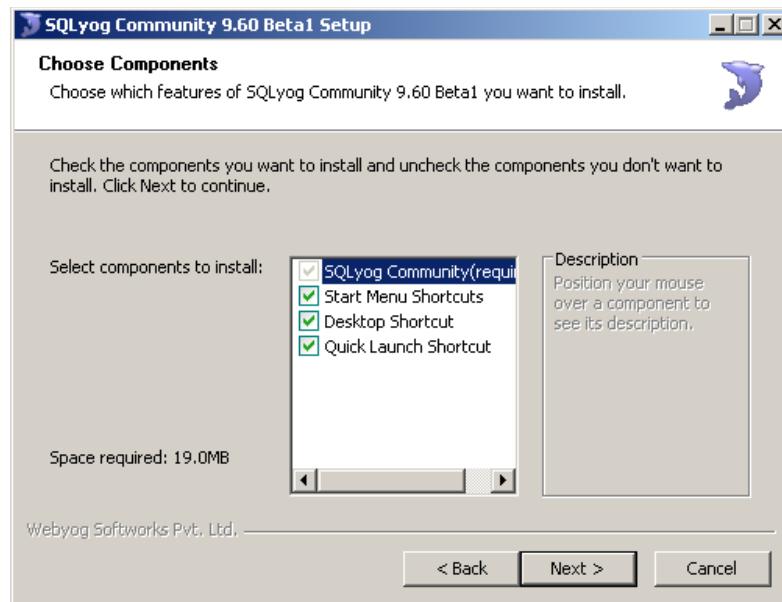


4.4 SQLyog – Community Edition

SQLyog adalah aplikasi desktop client untuk MySQL yang sangat populer digunakan. Aplikasi ini memiliki banyak fitur dengan GUI yang menarik sehingga memudahkan pengguna melakukan administrasi maupun melakukan pengolahan data MySQL.

Versi aplikasi ini ada yang gratis dan *open source* dan ada yang berbayar. Versi *open source* dari SQLyog ini dapat digunakan secara bebas namun dengan beberapa fitur *advanced* yang dibatasi.

Anda dapat melakukan instalasi aplikasi ini dengan menjalankan file setup **SQLyog-11.3.1-1.x86Community.exe** atau **SQLyog-11.3.1-1.x64Community.exe** yang terdapat pada folder MySQL/SQLYog pada DVD materi training Anda.



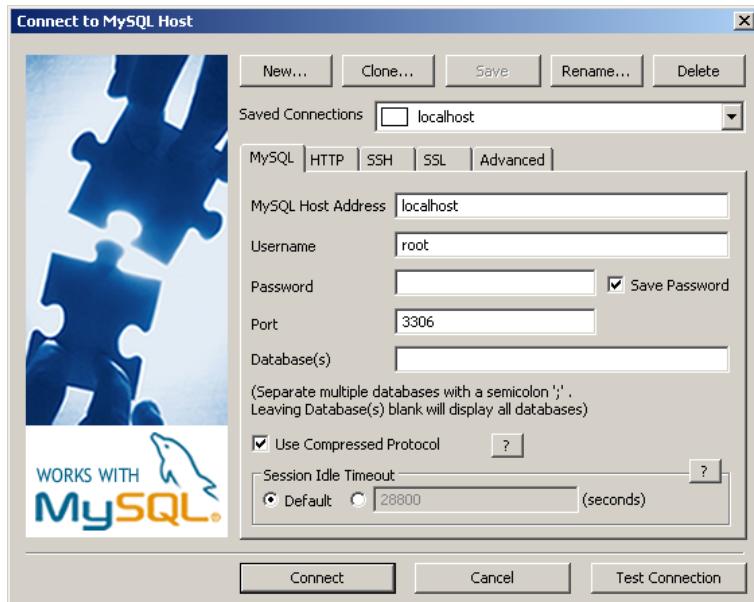
Pilihlah konfigurasi instalasi yang sesuai dengan kebutuhan dengan dibantu oleh instruktur Anda.

4.4.1 Membuat Koneksi

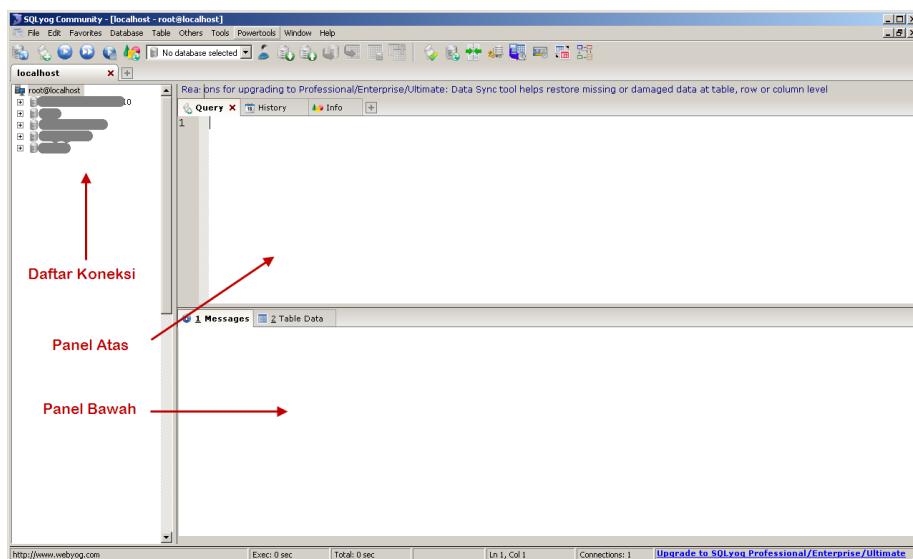
- Jalankan aplikasi **SQLYog**.
- Klik tombol **Continue**.



- Masukkan konfigurasi koneksi seperti terlihat pada gambar berikut ini.



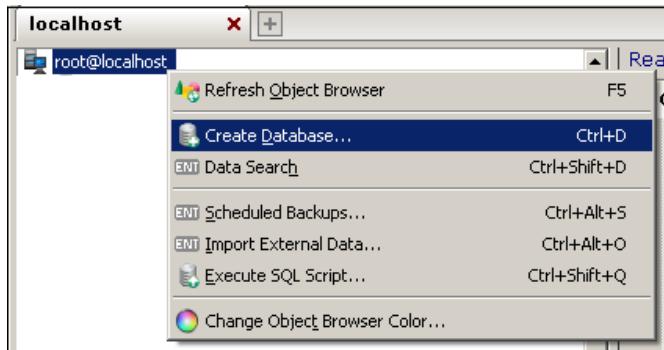
- Klik tombol **Connect**.
- Anda akan dibawa ke ruang kerja **SQLYog**.



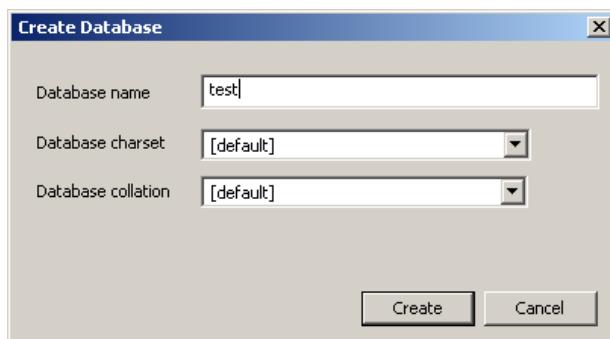
- Selesai.

4.4.2 Membuat Database

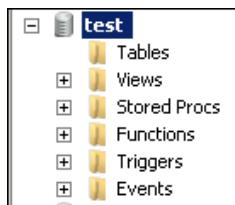
- Klik kanan node server – pada gambar terlihat sebagai **root@localhost** – dan pilih menu **Create Database**.



- Masukkan nama database yang kita inginkan pada input **Database name** dari dialog **Create Database**.



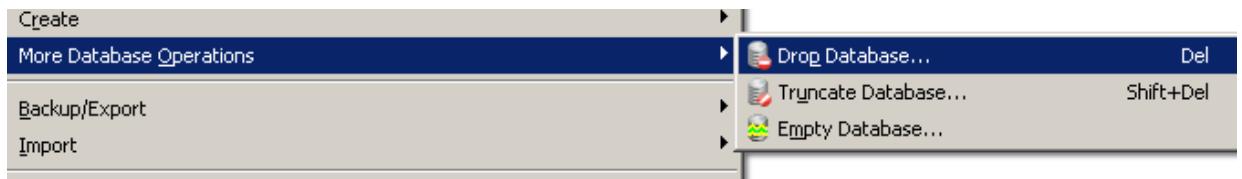
- Klik tombol **Create**, database kosong akan dihasilkan.



- Selesai.

4.4.3 Menghapus Database

- Klik kanan pada database yang ingin kita hapus.
- Klik menu **More Database Operations** → **Drop Database....**



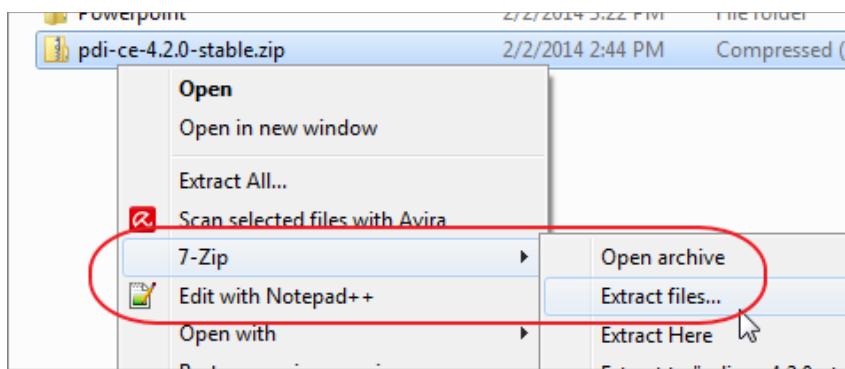
- Pilih **Yes** pada dialog konfirmasi.



- Selesai.

4.5 Pentaho Data Integration / Kettle

Instalasi aplikasi Pentaho Data Integration sangat sederhana. Buka file kompresi **pdi-ce-4.2.0-stable.zip** dengan aplikasi archiver seperti 7Zip yang telah diinstal sebelumnya, kemudian ekstrak ke folder yang Anda inginkan misalkan ke folder C:\.



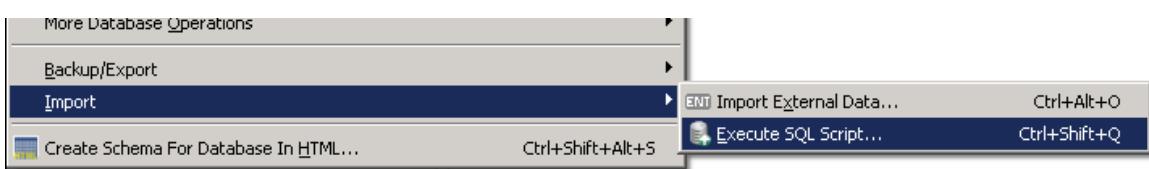
Anda akan mendapatkan folder data-integration sebagai hasil ekstraksi. Ubahlah nama folder tersebut menjadi **kettle4-2**.

4.6 Import Database PHI-Minimart

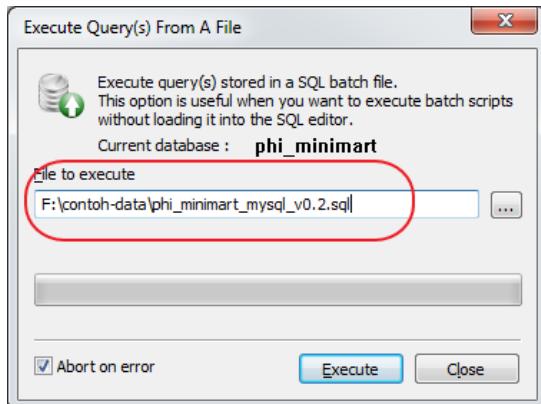
PHI-Minimart adalah sebuah contoh database relasional dengan table master dan transaksi yang sederhana yang akan digunakan sepanjang pelatihan data warehouse ini. Database ini dapat didownload di url <https://tinyurl.com/dqlab-ebook-data>.

Berikut adalah langkah-langkah untuk untuk mengimpor contoh database ini:

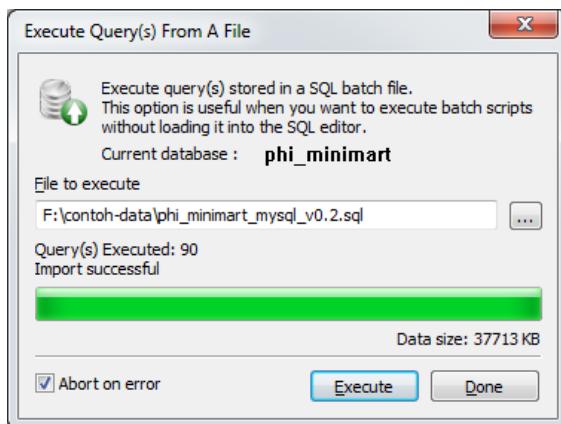
- Jalankan aplikasi SQLYog dan lakukan koneksi ke database server MySQL Anda.
- Buat database dengan nama **phi_minimart**.
- Klik kanan pada database tersebut, dan pilih menu **Import → Execute SQL Script ...**



- Pada dialog **Execute Query(s) From A File** pilih file **phi_minimart_mysql_v0.2.sql** dari file hasil download.



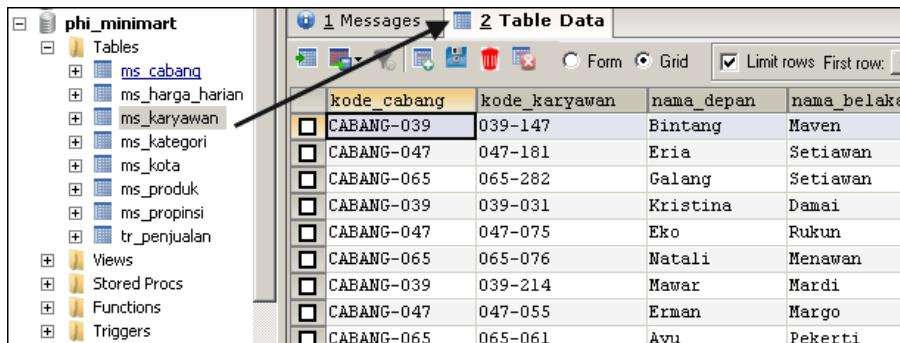
- Klik tombol **Execute**, kemudian klik tombol **Yes** untuk pesan lanjutan yang muncul. Proses import akan dimulai, tunggu sampai selesai.



- Klik tombol **Done**.
- Lakukan refresh dengan menekan tombol **F5** pada panel objek database. Terlihat database telah diimpor dengan sukses ditandai adanya tabel-tabel seperti pada gambar di bawah ini.



- Klik pada salah satu table, diikuti pemilihan menu tab **Table Data** pada panel bawah ruang kerja SQLYog untuk melihat isi data dari table tersebut.



The screenshot shows a database interface with a sidebar containing the schema 'phi_minimart'. Under 'Tables', several tables are listed: ms_cabang, ms_harga_harian, ms_karyawan, ms_kategori, ms_kota, ms_produk, ms_propinsi, and tr_penjualan. A black arrow points from the text '• Selesai.' to the 'ms_karyawan' table in the sidebar. The main area displays a grid titled '2 Table Data' with the following columns: kode_cabang, kode_karyawan, nama_depan, and nama_belaka. The data is as follows:

kode_cabang	kode_karyawan	nama_depan	nama_belaka
CABANG-039	039-147	Bintang	Maven
CABANG-047	047-181	Eria	Setiawan
CABANG-065	065-282	Galang	Setiawan
CABANG-039	039-031	Kristina	Damai
CABANG-047	047-075	Eko	Rukun
CABANG-065	065-076	Natali	Menawan
CABANG-039	039-214	Mawar	Mardi
CABANG-047	047-055	Erman	Margo
CABANG-065	065-061	Ayu	Pekerti

- Selesai.

Bagian 1

Pentaho Data Integration / Kettle



Bab 5. Pentaho Data Integration (Kettle)

5.1 Pendahuluan

Pentaho Data Integration (PDI) atau **Kettle** adalah utilitas ETL *open source* di bawah Pentaho Corp. Amerika.

Proyek ini awalnya merupakan inisiatif dari **Matt Casters**, seorang programmer dan konsultan Business Intelligence (BI) dari Belgia yang telah menangani berbagai proyek BI untuk perusahaan besar. Untuk informasi lebih lanjut, Anda dapat mengunjungi alamat blog <http://www.ibridge.be>.

Saat ini Kettle merupakan utilitas ETL yang sangat populer dan salah satu yang terbaik di pasaran. Beberapa kelebihannya adalah sebagai berikut:

- Memiliki koleksi modul pengolahan data yang cukup banyak. Lebih dari 100 modul atau *step*.
- Memiliki modul yang memudahkan perancangan model *data warehouse* seperti *Slowly Changing Dimension* dan *Junk Dimensions*.
- Performa dan kemampuan skalabilitas yang baik dan sudah terbukti.
- Dapat dikembangkan dengan berbagai *plugin* tambahan.

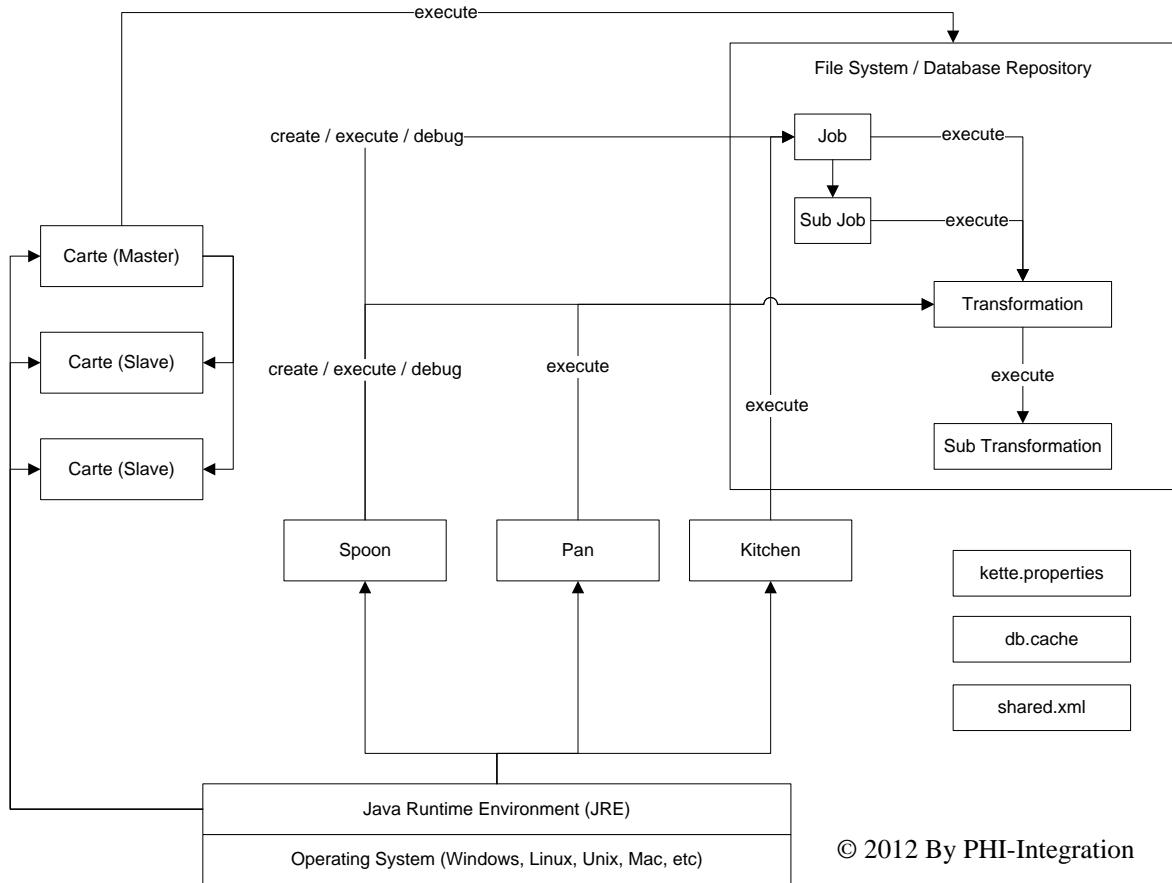
5.2 Komponen Aplikasi Kettle

Kettle terdiri dari 4 utilitas dalam bentuk *shell* dan *batch script* yaitu:

- **Spoon**
 - Merupakan utilitas yang digunakan untuk merancang dan menjalankan job / *transformation* secara visual (GUI)
 - Dipaketkan dengan nama file spoon.bat (batch script) dan spoon.sh (BASH shell script)
- **Pan**
 - Merupakan utilitas yang digunakan untuk mengeksekusi *transformation*
 - Umumnya dijalankan pada saat otomatisasi terjadwal (scheduled automation)
 - Dipaketkan dengan nama file pan.bat (batch script) dan pan.sh (BASH shell script)
- **Kitchen**
 - Merupakan utilitas yang digunakan untuk mengeksekusi job
 - Umumnya dijalankan pada saat otomatisasi terjadwal (scheduled automation)
 - Dipaketkan dengan nama file pan.bat (batch script) dan pan.sh (BASH shell script)
- **Carte**
 - Merupakan utilitas cluster web server yang digunakan untuk mengeksekusi job / *transformation*
 - Terutama digunakan untuk meningkatkan performa ETL dengan pembagian load kerja pada berbagai node Carte (master dan slave)

5.3 Arsitektur Kettle

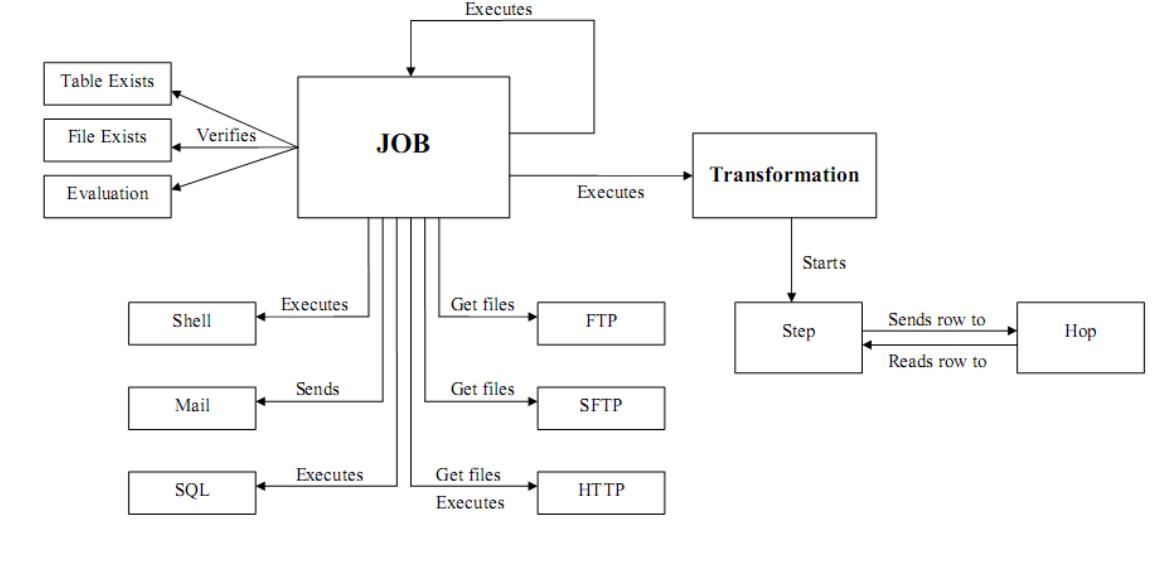
**Kettle 3 & 4
Components Architecture**



© 2012 By PHI-Integration

5.4 Model Konseptual Job / Transformation

Kettle Conceptual Model



Model Konseptual Kettle
(Gambar diambil dari dokumentasi distribusi Kettle)

5.5 Organisasi Folder Kettle

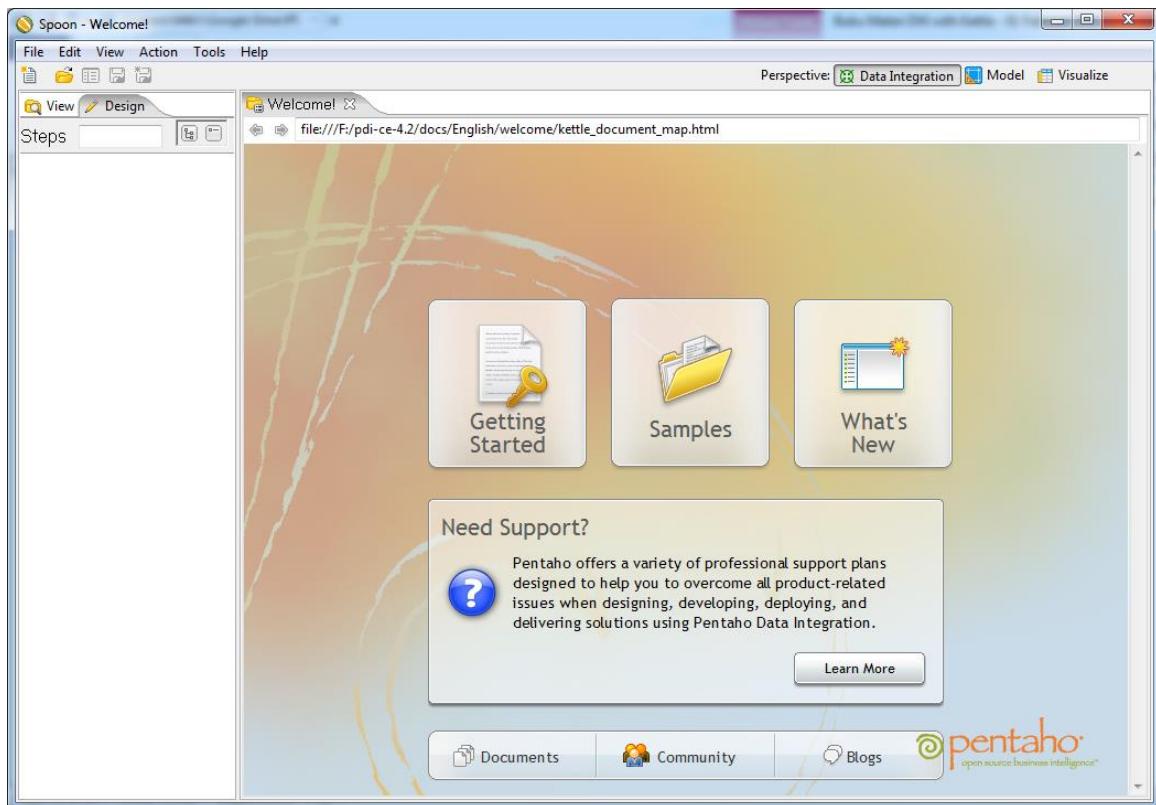
Nama Folder	Deskripsi
docs	Berisi dokumentasi penggunaan utilitas Kettle
extra	Berisi utilitas yang biasanya digunakan oleh programmer Java seperti menghasilkan source code <i>transformation</i> dalam Java
lib	Berisi file library Java (.jar) minimal yang diperlukan untuk menjalankan utilitas Kettle. Folder ini tidak berisi driver database.
libext	Berisi file library tambahan yang diperlukan untuk eksekusi job / transformasi Kettle
libext/JDBC	Merupakan lokasi jika Anda ingin menambahkan atau melakukan update driver database (JDBC)
libswt	Berisi file-file library user interface SWT
META-INF	Berisi file MANIFEST.MF dari hasil build ANT
plugins	Berisi folder / file plugin step tambahan
pwd	Berisi file password untuk Carte
samples	Berisi contoh-contoh job/ <i>transformation</i>
simple-jndi	Berisi file konfigurasi JNDI (Java Naming & Directory Interface)
ui	Berisi kumpulan file konfigurasi dan gambar <i>interface</i> visual Spoon

Bab 6. Spoon

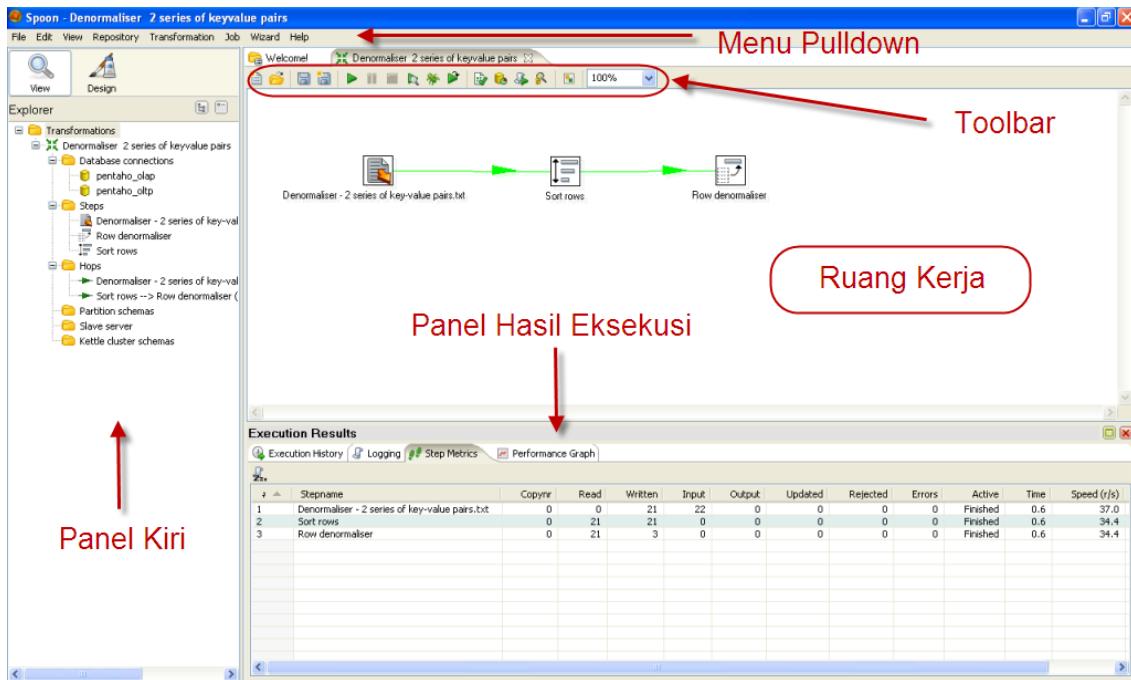
Spoon adalah utilitas grafik untuk merancang, mengeksekusi dan melakukan troubleshooting dari proses ETL melalui *job* dan *transformation*.

6.1 Praktek: Menjalankan Spoon

- Buka folder instalasi Kettle.
- Jalankan file **spoon.bat**.
- Pada saat muncul dialog **Select a repository**, klik tombol **No Repository**.
- Klik tombol **OK** untuk menutup jendela tips yang muncul.
- Akan muncul ruang kerja utama Spoon dengan halaman **Welcome Page**.



6.2 Lingkungan Kerja Spoon



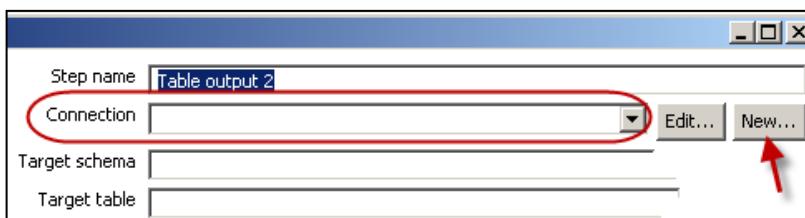
Lingkungan kerja Spoon terdiri dari beberapa bagian sebagai berikut:

- **Pulldown Menu:** koleksi menu dari Spoon yang terintegrasi dalam satu *toolbar* di bagian atas ruang kerja Spoon.
- **Welcome Screen:** merupakan halaman pembuka Kettle yang berisi informasi ke situs Pentaho. Untuk mengaktifkan welcome screen pilih menu Help | Show the Welcome Screen
- **Toolbar,** terdiri dari job / transformation toolbar
- Panel Hasil Eksekusi atau ***Execution Results***, terdiri dari:
 - **Execution History:** Data historis eksekusi
 - **Logging:** Berisi log dari eksekusi job/transformation
 - **Job Metrics:** Berisi detil dari step-step yang telah dieksekusi
 - **Step Metrics:** Berisi detil jumlah pembacaan data (write, update, dll) per satuan waktu detik dari step-step yang telah dieksekusi
 - **Performance Graphs:** Tampilan grafis dari pembacaan data dari Step Metrics

Bab 7. Konsep dan Komponen Kettle

7.1 Koneksi Database

- Kettle mendukung banyak sekali jenis koneksi database terpopuler saat ini seperti Oracle, MS SQL Server, MongoDB, MySQL, PostgreSQL, dan lain-lain.
- Pastikan bahwa driver database JDBC (*Java DataBase Connectivity*) yang diperlukan terdapat di folder libext/JDBC.
- Koneksi database yang dapat dikenali oleh Job/*Transformation* dapat dibuat di panel kiri bagian **Database connections** atau melalui tombol **New...** pada kotak entri *Connection*.



7.2 Step

- **Step** adalah komponen / modul Kettle yang mengerjakan tugas tertentu pada suatu job / *transformation*.
- Kettle terdiri dari 100 lebih *step* bawaan yang tersebar pada 2 kategori:
 - **Job Steps** : *step* yang berjalan secara sekuensial dan lebih berfokus kepada control flow secara keseluruhan dari tugas ETL
 - **Transformation Steps** : *step* yang berjalan secara parallel dan lebih menitikberatkan I/O data
- Referensi penggunaan berbagai *step* dapat ditemukan di dokumen PDI / Kettle *user guide*.
- Step pada Kettle dapat ditambahkan sebagai *plugin*.

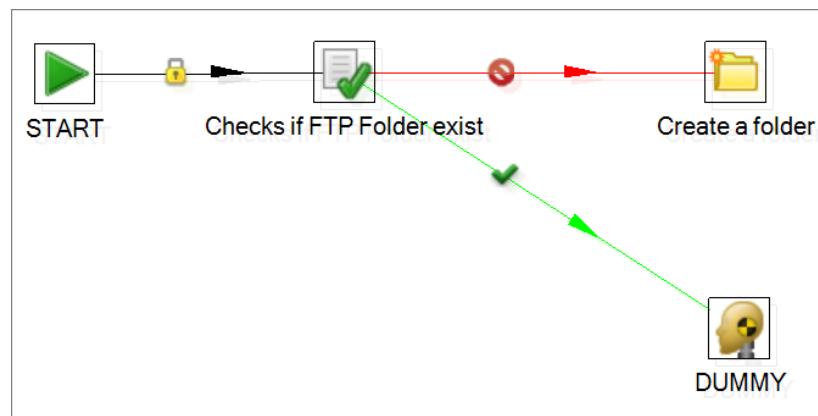
7.3 Hop

- Suatu **Hop** menggabungkan 2 *step* baik sebagai penentu aliran / *flow* dari job atau penghantar data di *transformation*
- Pada *transformation*, hop memiliki berbagai warna yang memiliki arti tersendiri.
- Tabel pada halaman berikut ini berisi daftar lengkap dari bentuk hop pada Spoon – Kettle 4.2.

Bentuk Hop	Arti
	Distribusi baris data secara merata (Distribute Rows)
	Referensi kepada step lain
	Membagi data dengan melakukan duplikasi (Copy Rows)
	Warna biru, hop yang masih dalam keadaan edit (belum final)
	Warna abu-abu, hop dinonaktifkan (disabled)
	Hop memiliki data yang error

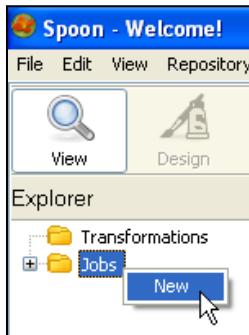
- Sedangkan pada job, hop memiliki berbagai warna yang memiliki arti tersendiri. Tabel berikut adalah daftar lengkap dari arti tersebut pada Spoon – Kettle 4.2.

Warna Hop	Arti
Hijau	Jika evaluasi kondisi benar maka lanjutkan ke <i>step</i> berikutnya
Merah	Jika evaluasi kondisi salah maka lanjutkan ke <i>step</i> berikutnya
Hitam	Lanjutkan ke <i>step</i> berikutnya dalam kondisi apapun

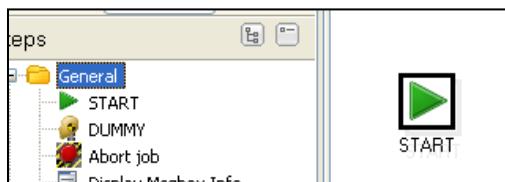


7.4 Job

- **Job** adalah komponen dari Kettle yang menangani kontrol atas aliran tugas (*flow control*). Job tidak memproses aliran data secara langsung. Job bisa terdiri satu atau beberapa *transformation* dan *job* lain.
- Cara membuat job:
 - Pilih menu **File | New | Job** – atau tekan tombol **CTRL+ALT+N**.
 - Melalui panel kiri **View**, klik kanan pada entri **Jobs**, pilih menu **New**.



- Setiap job harus dimulai dari suatu *step Start*.



- Job berjalan secara sekuensial dan dapat dijadwalkan (*scheduled*).
- Simbol job dalam Spoon adalah



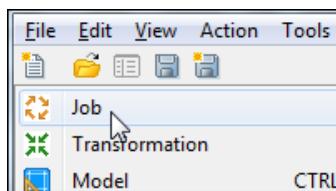
7.5 Praktek: Job Pertama – Message Box

Untuk latihan pertama menggunakan Spoon, kita akan membuat suatu *job* sederhana yang akan menampilkan satu kotak pesan ketika dijalankan.

- Buat file job baru:
 - Klik menu icon **New file**.



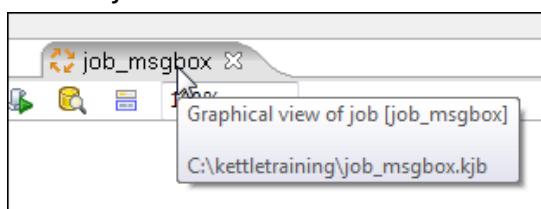
- Pilih **Job**.



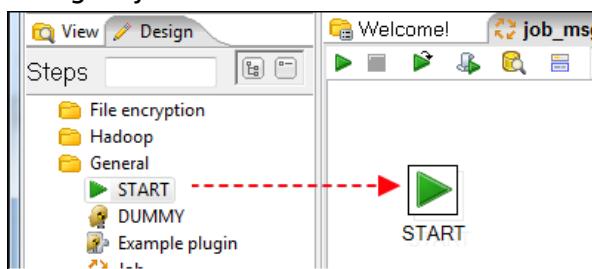
- Simpan file tersebut dengan nama **job_msgbox.kjb**.



- Arahkan kursor mouse pada tab **job_msgbox** untuk memunculkan informasi nama dan lokasi file dari job kita.



- Pada pilihan **Design** di panel kiri, buka folder **General** dan klik step **Start** dan diseret (drag) ke ruang kerja.



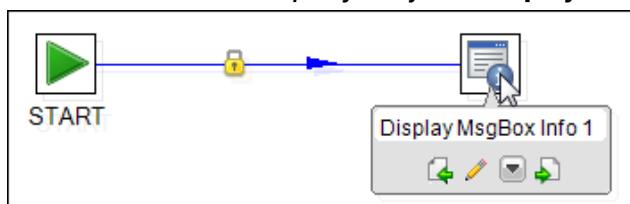
- Dengan cara yang sama, masukkan step **Display MsgBox Info** dari folder **Utility**.



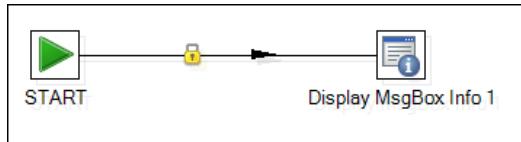
- Hubungkan kedua step yang telah ada di ruang kerja dengan salah satu dari dua cara berikut:
 - CTRL + Click** pada step awal / sumber yaitu **Start** kemudian **CTRL+Click** pada step tujuan yaitu **Display MsgBox Info 1**, klik kanan dan pilih **New Hop**.



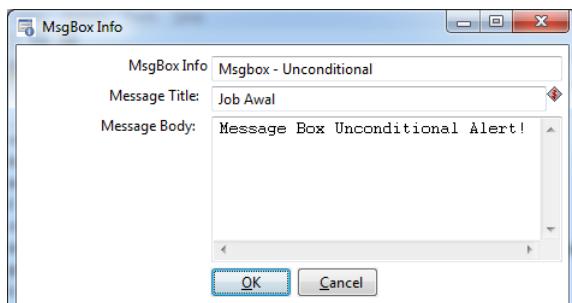
- Dengan mouse, klik roda tengah mouse kita (*roller mouse*) pada step **Start**. Sambil ditahan arahkan cursor ke step tujuan yaitu **Display MsgBox Info 1**.



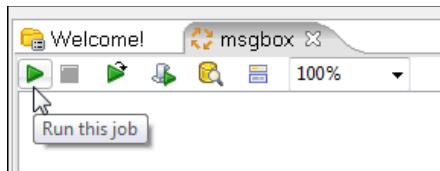
- Kedua Step akan terhubung dengan hop berwarna hitam. Warna hitam menunjukkan kondisi tidak bersyarat (*unconditional*).



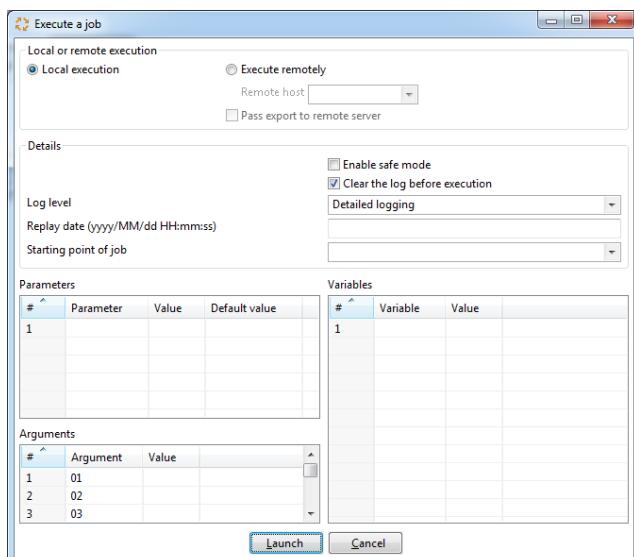
- Klik dua kali pada step **Diplay MsgBox Info 1**, dan masukkan konfigurasi seperti pada gambar berikut ini.



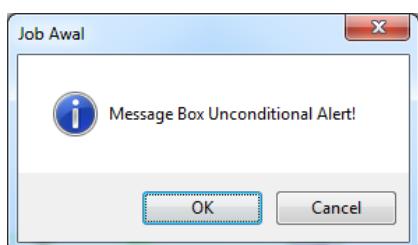
- Klik tombol **OK**, perhatikan bahwa nama *step* tersebut telah berubah dari **Display MsgBox Info 1** menjadi **MsgBox – Unconditional**.
- Job kita sudah siap dijalankan. Simpan file kita, dan klik tombol **Run this job** pada bagian menu *toolbar*.



- Klik tombol **Launch** pada dialog **Execute a job**.

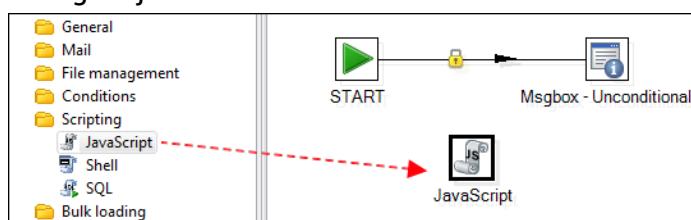


- Hasil eksekusi dari *job* ini adalah munculnya sebuah kotak pesan seperti terlihat pada gambar berikut.



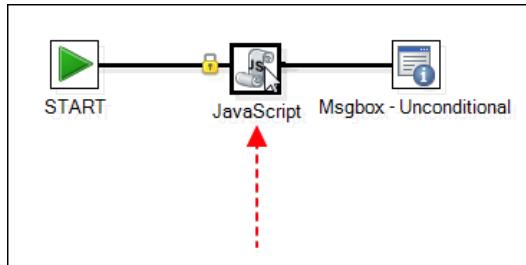
7.6 Praktek: Message Box dengan Kondisi

- Masih dengan melanjutkan job sebelumnya, sekarang tambahkan *step Javascript* ke dalam ruang kerja kita.

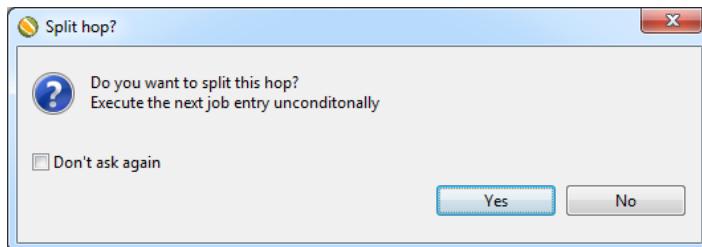


- Klik dan tahan *step Javascript* serta arahkan *step* tersebut ke hop penghubung antara *step*

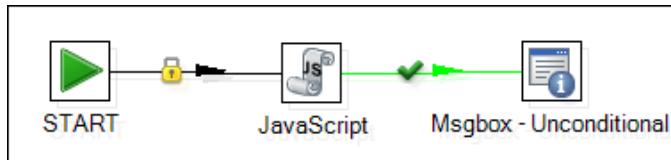
Start dan **MsgBox – Unconditional** sampai hop tersebut kelihatan lebih tebal. Lepaskan klik dari *step* tersebut.



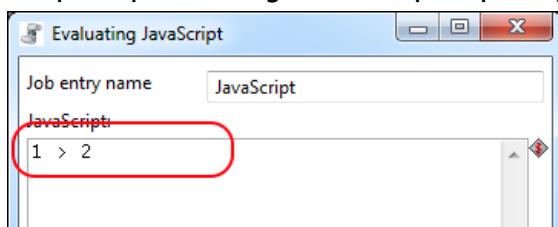
- Klik **Yes** pada dialog **Split hop?**.



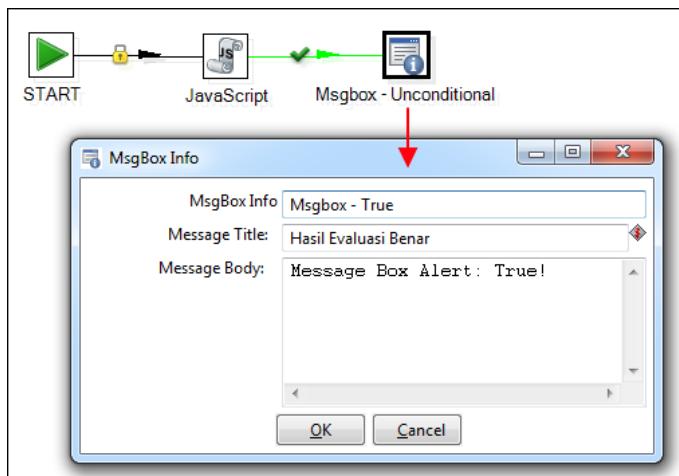
- Perhatikan bahwa *step Javascript* tersebut akan menjadi *step intermediate* baru. Hop yang menuju **step MsgBox –Unconditional** berwarna hijau, yang memiliki arti bahwa *step* tersebut akan tereksekusi jika *step Javascript* sukses dijalankan.



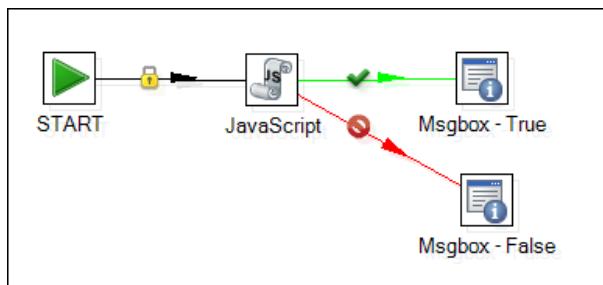
- Klik 2 kali pada *step Javascript* untuk membuka editor **Evaluating JavaScript**. Masukkan ekspresi perbandingan nilai seperti pada gambar di bawah. Klik tombol **OK** setelah selesai.



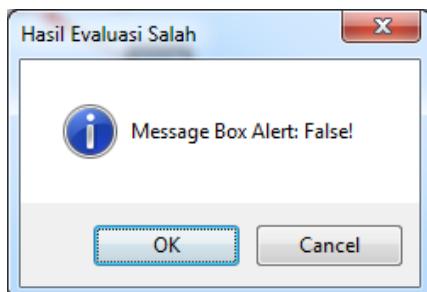
- Rubah nama dan isi dari step **MsgBox – Unconditional** sebagai berikut.



- Tambahkan step **Display MessageBox Info** baru pada ruang kerja kita dengan konfigurasi berikut:
 - Msgbox Info:** Msgbox – False
 - Message Title:** Hasil Evaluasi Salah
 - Message Body:** Message Box Alert : False !
- Tarik *hop* baru dari step **Evaluasi** ke **MsgBox – False**. Perhatikan bahwa *hop* yang baru terbentuk tersebut akan diberi warna merah secara otomatis.



- Jalankan *job* kita dan hasil yang kita harapkan adalah munculnya dialog sebagai berikut. Hasil ini artinya evaluasi ekspresi matematika dari step Javascript memang salah. Klik tombol **OK** untuk menyelesaikan eksekusi job.



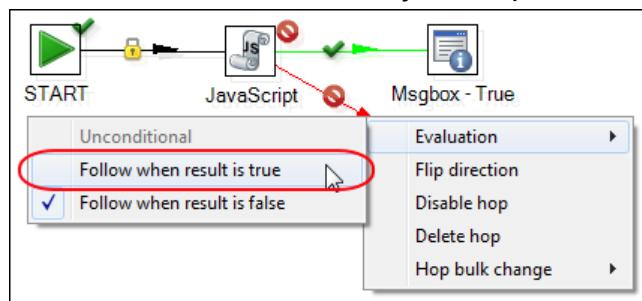
- Pada panel **Execution Results** yang terdapat di bawah ruang kerja kita, coba analisa perjalanan dari eksekusi job kita.

Execution results			
Job / Job Entry	Comment	Result	Reason
job_msgbox			
Job: job_msgbox	Start of job execution	start	start
START	Start of job execution	start	
START	Job execution finished	Success	
JavaScript	Start of job execution		Followed unconditional link
JavaScript	Job execution finished	Failure	
Msgbox - False	Start of job execution		Followed link after failure
Msgbox - False	Job execution finished	Success	
Job: job_msgbox	Job execution finished	Success	finished

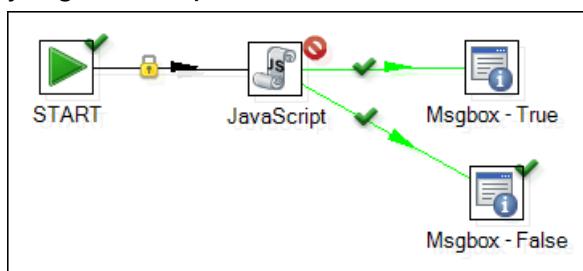
- Sekarang, cobalah ganti isi ekspresi pada step **Evaluasi** kita menjadi **1 < 2**.
- Simpan job tersebut dan coba eksekusi kembali. Apa kesimpulan yang bisa Anda ambil?

7.7 Praktek: Merubah Kondisi Hop pada Job

- Setiap hop pada Job dapat memiliki aturan / kondisi sehingga dengan kondisi tertentu alur hop akan dijalankan atau tidak.
- Klik kanan pada hop antara step **Evaluasi** dan **MsgBox – False**, pilih **Evaluation** dan perhatikan bahwa ada 2 entri yang dapat dipilih, yaitu:
 - Follow when result is false**: lanjutkan apabila nilai kondisi dari step sebelumnya salah.
 - Follow when result is true**: lanjutkan apabila nilai kondisi dari step sebelumnya benar.



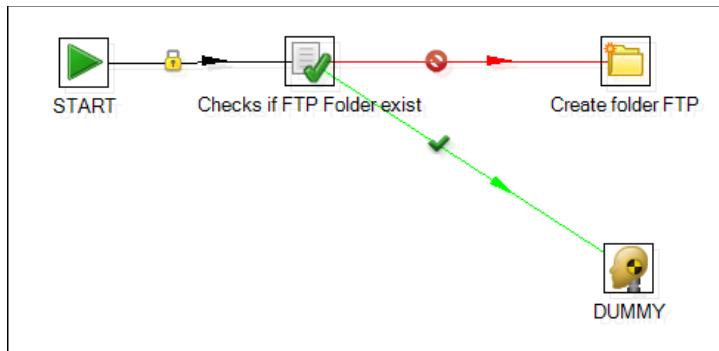
- Ambil pilihan **Follow when result is true**, simpan dan kembali jalankan job tersebut. Apa hasil yang Anda dapatkan?



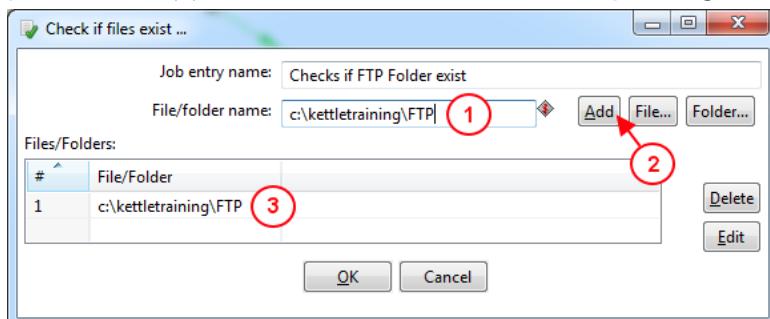
7.8 Praktek: Folder Auto Creation dengan Menggunakan Schedule

- Buatlah *job* baru dengan yang akan melakukan hal berikut ini:
 - melakukan pengecekan eksistensi dari folder **c:\kettletraining\FTP** dengan durasi berulang per dua detik.

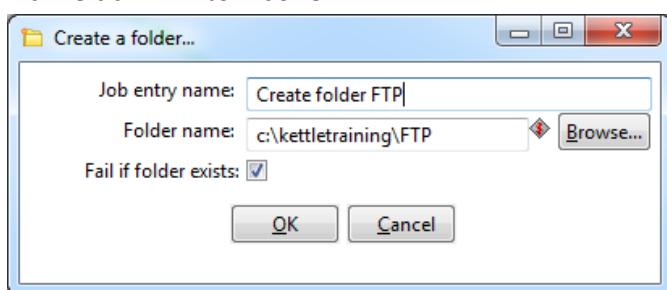
- jika folder tersebut belum ada maka folder tersebut akan dibuat, jika folder tersebut sudah ada maka tidak ada tindakan apapun yang dilakukan.
- Masukkan *step-step* berikut ini pada ruang kerja Spoon :
 - General → Start
 - Conditions → Checks if files exist
 - File Management → Create a folder
 - General → DUMMY
- Susun *flow* dari *step-step* tersebut seperti pada gambar di bawah ini



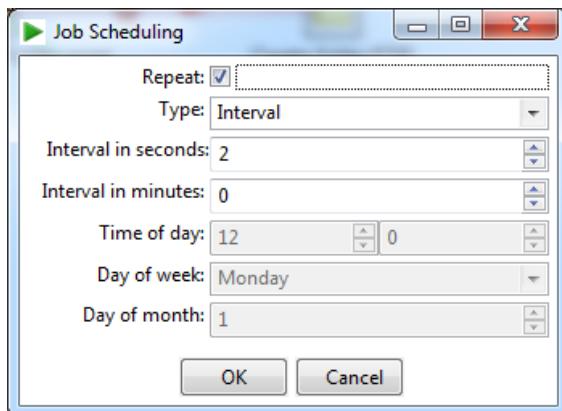
- Klik dua kali pada step **Check if FTP folder exist**. Pada kotak **File/folder name**, masukkan nama folder **c:\kettletraining\FTP** (1). Klik tombol **Add** (2), nama folder tersebut akan ditambahkan pada daftar (3). Klik tombol **OK** untuk menutup dialog.



- Klik dua kali pada step **Create folder FTP**, masukkan nama folder yang sama pada kotak **Folder Name** dan klik tombol **OK**.



- Lakukan konfigurasi berikut pada step **Start**, digunakan agar seluruh *job* ini diulang per tiap dua detik.



- Simpan *job* tersebut dengan nama file **c:\kettletraining\cek_folder.kjb**.
- Eksekusi *job* tersebut dan pantau folder **c:\kettletraining** melalui aplikasi Windows Explorer. Terlihat bahwa folder FTP akan dibuat oleh job ini. Dan jika kita menghapus folder tersebut, dengan *job* ini folder FTP tetap akan dibuat kembali.

7.9 Transformation

- **Transformation** adalah komponen dari Kettle yang melakukan proses aliran data sebenarnya. Semua proses pengambilan (*extract*), perubahan (*transform*) dan penyimpanan (*load*) data dilakukan disini.
- Cara membuat *transformation*
 - Pilih menu File | New | Transformation atau tekan CTRL+ALT+N.
 - Dari Panel Kiri - View, klik kanan entri **Transformations** | klik **New**
- Simbol *transformation* dalam Spoon adalah

7.10 Row

- **Row** atau baris data adalah aliran data antar sumber input ke *step-step* lainnya pada *transformation*.
- **Row** terdiri dari 2 bagian :
 - **Meta Data** dari row, yaitu berisi deskripsi dan tipe, panjang, format data.
 - **Isi Data** dari row itu sendiri.
- Tipe data untuk row yang didukung oleh Kettle adalah sbb :
 - **Integer** : Bilangan Bulat
 - **Number** : Bilangan Pecahan
 - **String** : Rangkaian Karakter
 - **Date** : Tanggal
 - **Boolean** : Nilai benar / salah
 - **BigNumber** : Bilangan Pecahan besar
 - **Binary** : Data biner

7.11 Variable

- **Variable** adalah penyimpan data di memori dengan penggunaan nama tertentu.
- Beberapa kategori *variable* di Kettle adalah:
 - **Environment Variables**: Berlaku secara global
 - **Kettle Variables**: Berlaku dinamis, variable ini dibuat menggunakan step **Set Variable**
 - **Internal Variables**: Variable-variable yang sudah didefinisikan sebelumnya dan punya nilai tertentu
- Dapat diakses menggunakan dua notasi yaitu **`$(NamaVariable)`** atau **`%%NamaVariable%%`**.
- Pada beberapa dialog di Spoon, jika terdapat tanda gambar tanda tanya (?) maka nilainya bisa digantikan dengan variable.

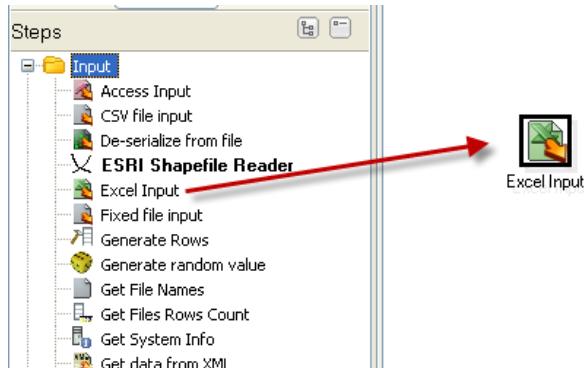
7.12 Log

- **Log** pada Kettle terdapat pada berbagai level pad job, *transformation* dan *step* dapat diarahkan ke berbagai output:
 - Layar Monitor
 - File system
 - Database relasional
- Rincian dari log juga dapat dikonfigurasi sesuai kebutuhan dan keinginan kita mulai dari tanpa log sama sekali sampai dengan sangat detil (*row details*).

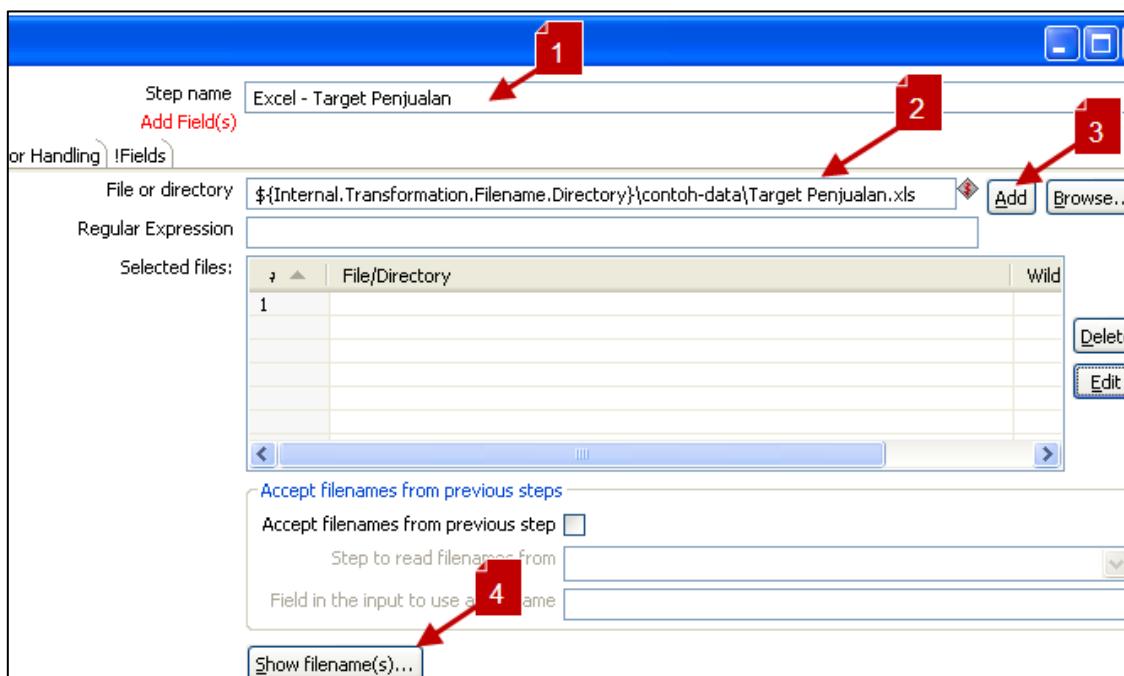
7.13 Praktek: Transformation Untuk Membaca 1 File Excel

A	B	C	D	E	F	G	H	I
1 Kode Cabang	Kode Produk	Tahun	Jan	Feb	Mar	Apr	May	Jun
2 CABANG-039	PROD-0000001	2008	32637383	29651096	55384720	42483638	30497420	4822516
3 CABANG-039	PROD-0000002	2008	18434034	16562378	19087829	18493675	14282259	1288988
4 CABANG-039	PROD-0000003	2008	19058922	15804388	20328401	17887601	24052992	2435183
5 CABANG-039	PROD-0000004	2008	35538965	28002486	25064300	28644736	33397134	3302438
6 CABANG-039	PROD-0000005	2008	26668799	20019024	20209767	34388468	21688292	3050644
7 CABANG-039	PROD-0000006	2008	39401262	35996178	32824821	24069108	37727115	2341347
8 CABANG-039	PROD-0000007	2008	16702554	11239313	17632157	15674109	16086168	1604741
9 CABANG-039	PROD-0000008	2008	4516056	3754410	3804348	2821245	2873151	368550
10 CABANG-039	PROD-0000009	2008	13529753	17908166	19962346	20082202	14655992	1880787
11 CABANG-039	PROD-0000010	2008	3077235	4554995	4869763	5210387	5061722	440941
12 CABANG-039	PROD-0000011	2008	5213289	3801475	5697202	3765894	2922297	516429
13 CABANG-039	PROD-0000012	2008	10052644	10466095	16275752	10705405	14667357	1183384

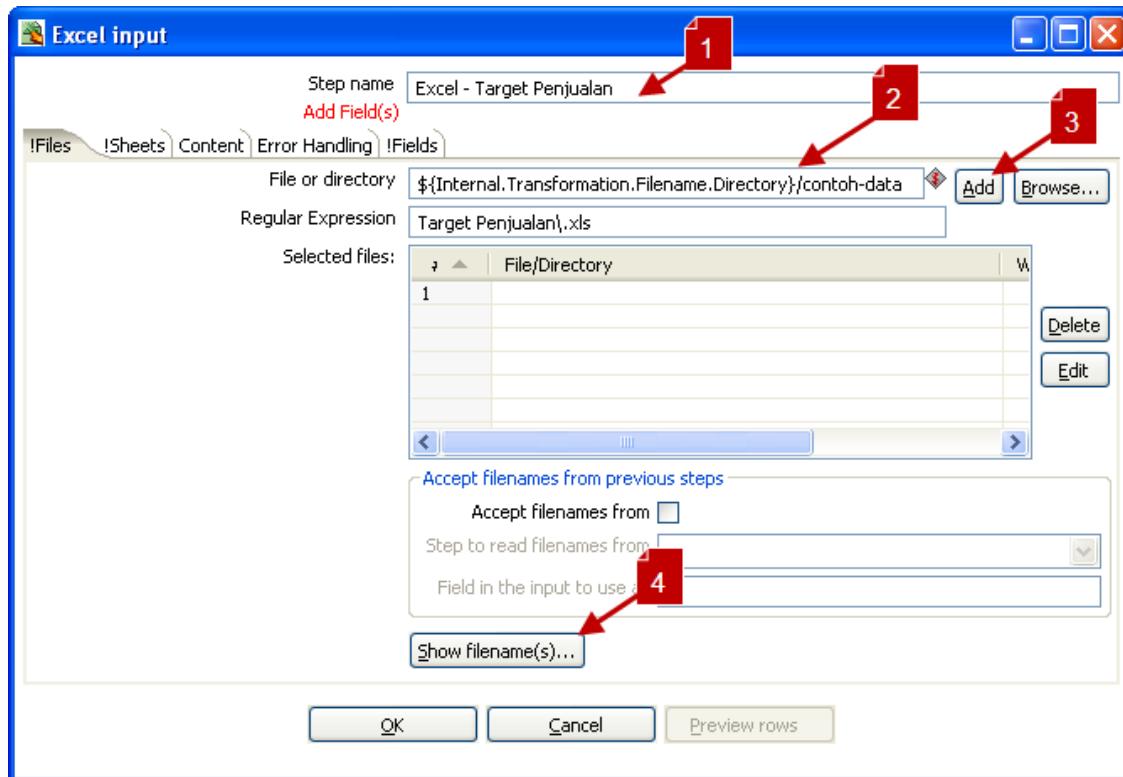
- Rancang suatu *transformation* dengan nama **c:\kettletraining\transformasi_target.ktr**.
- Masukkan **step Excel Input** yang akan digunakan untuk membaca file spreadsheet Excel **c:\kettletraining\contoh-data\Target Penjualan.xls**.



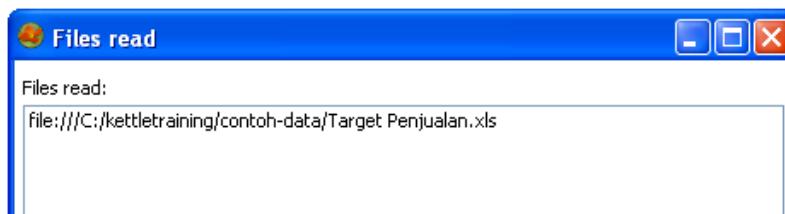
- Klik 2 kali pada *step* tersebut. Namakan *step* tersebut dengan **Excel - Target Penjualan**.
- Masukkan konfigurasi dengan petunjuk salah satu dari 2 gambar berikut di bawah ini.



atau



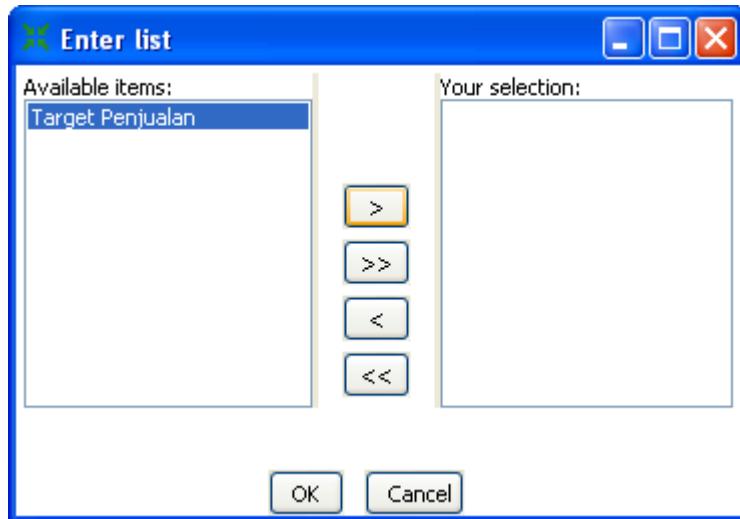
- Klik tombol **Add** sehingga pola nama file yang kita masukkan di atas tersebut masuk dalam daftar entri **Selected files**.
- Klik tombol **Show filename(s)...** untuk melihat apakah nama pola yang kita masukkan dikenali. Apabila berhasil maka akan muncul dialog dengan daftar file Excel seperti berikut ini.



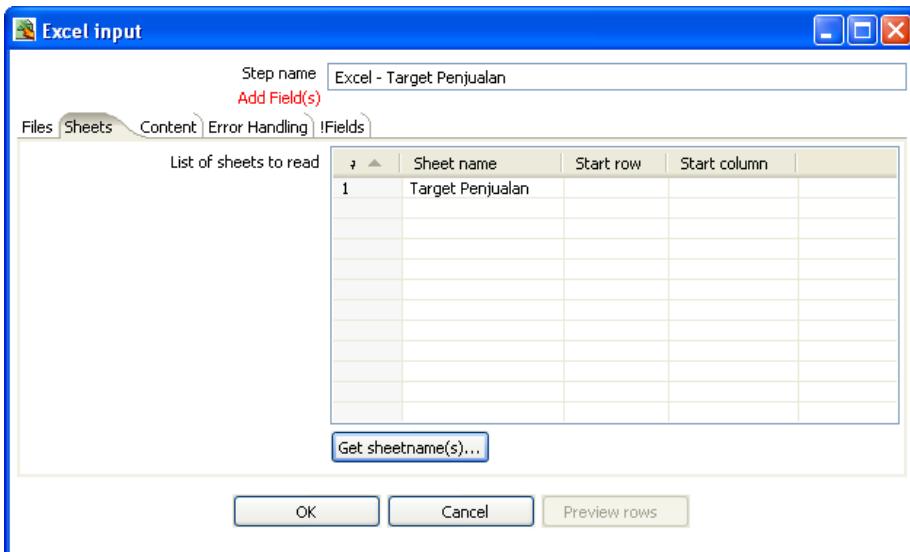
- Klik tombol **Close**.
- Klik tab **Sheets**, lanjutkan dengan klik tombol **Get sheetname(s)**.



- Pada dialog "Enter List", masukkan sheet "Target Penjualan" dari list "Available Items" ke "Your Selection". Klik tombol "OK".



- Hasil pada tab **Sheets** akan tampak sebagai berikut.

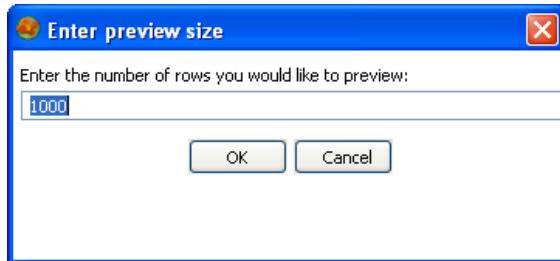


- Pada tab **!Fields** klik tombol **Get fields from header row....**
- Dalam beberapa saat semua fields dan meta datanya akan coba dikenali oleh Kettle. Perhatikan meta data (tipe) dari semua *field* tersebut.

	Name	Type	Length	Precision	Trim type	Repeat	Format
1	Kode Cabang	String			none	N	
2	Kode Produk	String			none	N	
3	Tahun	Number			none	N	
4	Jan	Number			none	N	
5	Feb	Number			none	N	
6	Mar	Number			none	N	
7	Apr	Number			none	N	
8	May	Number			none	N	
9	Jun	Number			none	N	
10	Jul	Number			none	N	
11	Aug	Number			none	N	

Get fields from header row...

- Klik tombol **Preview rows** untuk melihat data yang dibaca dari Excel.
- Masukkan jumlah preview sebanyak 1000 baris data pada dialog **Enter preview size** dan klik tombol **OK**.



- Akan muncul jendel **Examine preview data** yang merupakan *preview* dari data Excel kita. Klik tombol **Close** untuk menutup dialog tersebut.

#	Kode Cabang	Kode Produk	Tahun	Jml
1	CABANG-039	PROD-0000001	2008.0	32637383.0
2	CABANG-039	PROD-0000002	2008.0	18434034.0
3	CABANG-039	PROD-0000003	2008.0	19058922.0
4	CABANG-039	PROD-0000004	2008.0	35538965.0
5	CABANG-039	PROD-0000005	2008.0	26668799.0
6	CABANG-039	PROD-0000006	2008.0	39401262.0
7	CABANG-039	PROD-0000007	2008.0	16702554.0
8	CABANG-039	PROD-0000008	2008.0	4516056.0
9	CABANG-039	PROD-0000009	2008.0	13529753.0
10	CABANG-039	PROD-0000010	2008.0	3077235.0
11	CABANG-039	PROD-0000011	2008.0	5213289.0

- Klik tombol "OK".

- Selesai.

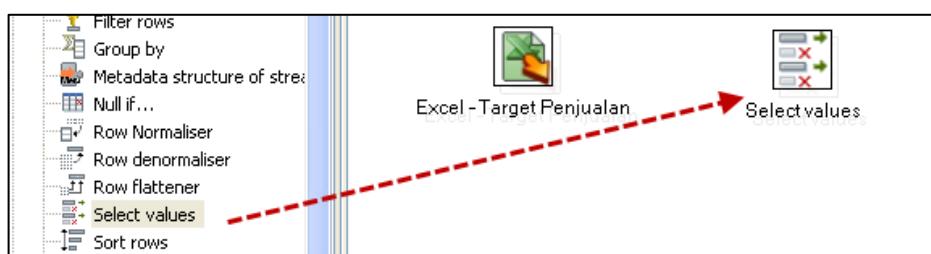
7.14 Praktek: Merubah Meta Data

Apabila kita perhatikan transformasi dari latihan praktek sebelumnya, terlihat bahwa tipe data dari tahun dan nilai target tidak seperti yang kita harapkan. Harusnya semua data tersebut bertipe ***bilangan bulat*** atau ***integer***, bukannya ***number*** atau ***bilangan angka pecahan*** seperti yang dikenali oleh Spoon secara otomatis.

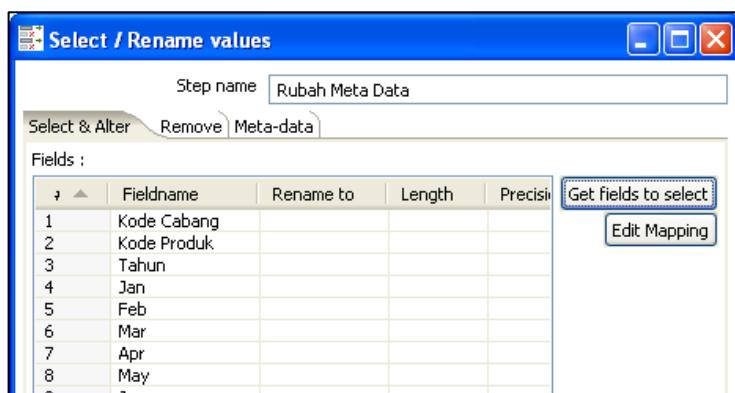
Untuk memperbaiki hal ini, kita dapat merubahnya langsung melalui konfigurasi pada step **Excel Input** tersebut, atau melalui penggunaan step **Select Values**.

Berikut adalah langkah-langkah untuk merubah meta data menggunakan cara terakhir yang disebutkan :

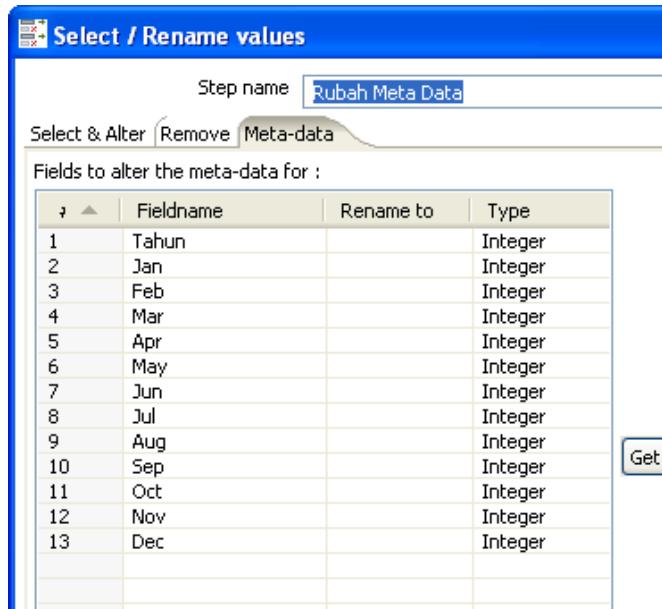
- Masukkan step **Select Values** pada ruang kerja kita.



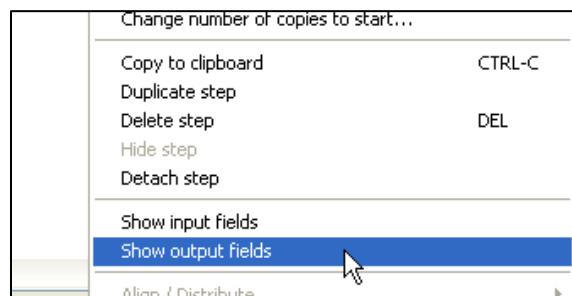
- Buat hop antara step **Excel – Target Penjualan** dan **Select values**.
- Klik 2 kali pada step **Select Values**.
- Ganti nama step tersebut ke **Rubah Meta Data**.
- Pada tab **Select & Alter** klik tombol **Get fields to select**.



- Pada tab **Meta Data** klik tombol **Get fields to change**.
- Hapus **field Kode Cabang** dan **Kode Produk** dari daftar dengan cara mengklik kolom paling kiri yang berdekatan dengan nama **field** tersebut – tekan tombol **Delete**.
- Pada kolom **Type**, ambil pilihan **Integer** dari daftar tipe-tipe data yang tersedia. Lakukan hal tersebut untuk semua nama **field** yang tersisa - seperti terlihat pada gambar berikut.



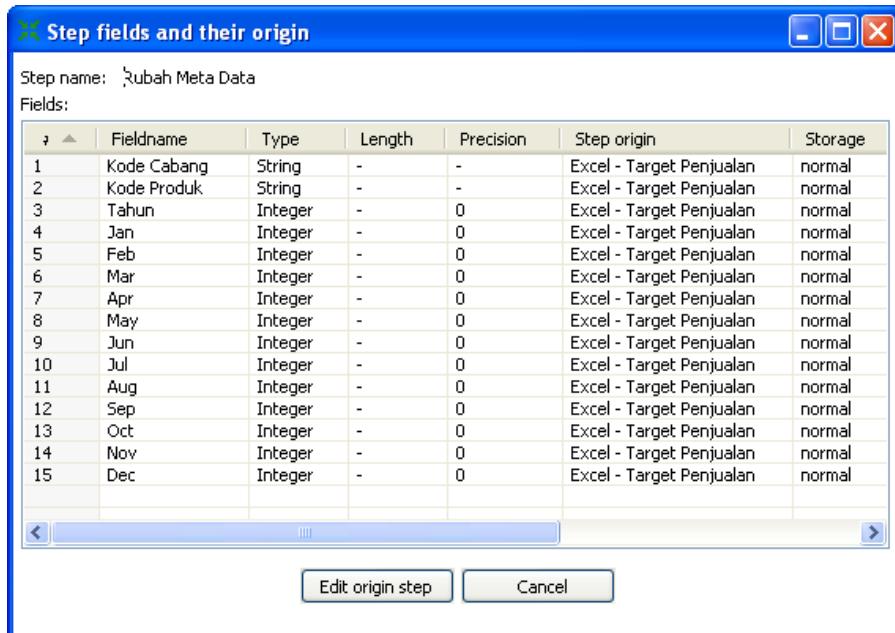
- Klik tombol **OK** setelah selesai.
- Untuk melihat perubahan meta data pada tiap step, klik kanan pada step **Excel – Target Penjualan**, kemudian pilih menu **Show output fields**.



- Akan muncul window yang berisi informasi output metadata dari step tersebut.

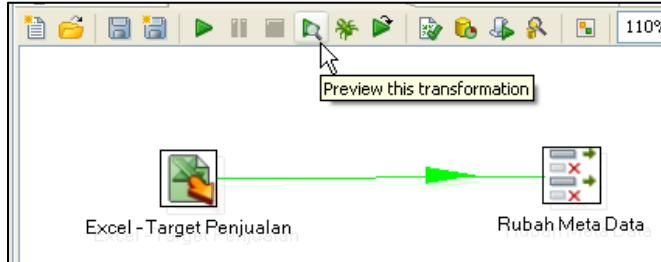
Step fields and their origin						
Step name: Excel - Target Penjualan						
Fields:						
#	Fieldname	Type	Length	Precision	Step origin	
1	Kode Cabang	String	-	-	Excel - Targ	
2	Kode Produk	String	-	-	Excel - Targ	
3	Tahun	Number	-	-	Excel - Targ	
4	Jan	Number	-	-	Excel - Targ	
5	Feb	Number	-	-	Excel - Targ	
6	Mar	Number	-	-	Excel - Targ	
7	Apr	Number	-	-	Excel - Targ	
8	May	Number	-	-	Excel - Targ	
9	Jun	Number	-	-	Excel - Targ	
10	Jul	Number	-	-	Excel - Targ	
11	Aug	Number	-	-	Excel - Targ	
12	Sep	Number	-	-	Excel - Targ	
13	Oct	Number	-	-	Excel - Targ	
14	Nov	Number	-	-	Excel - Targ	
15	Dec	Number	-	-	Excel - Targ	

- Lakukan hal yang sama untuk step **Rubah Meta Data**. Terlihat bahwa field Tahun & Bulan sudah berubah tipe datanya dari **Numeric** menjadi **Integer**.

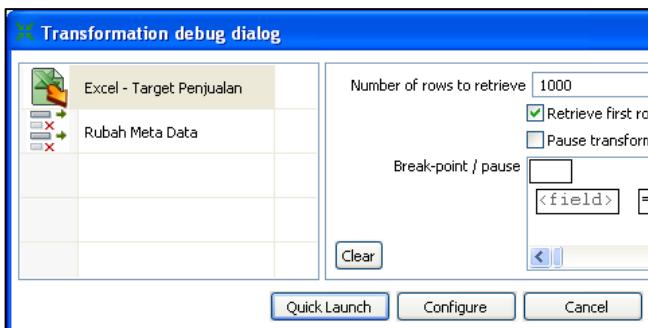


7.15 Praktek: Preview Data

- Klik step **Select Values**, kemudian klik tombol **Preview this transformation**.



- Klik tombol **Quick Launch**.



- Akan keluar dialog **preview data**. Klik tombol **Close** untuk menutup dialog ini.

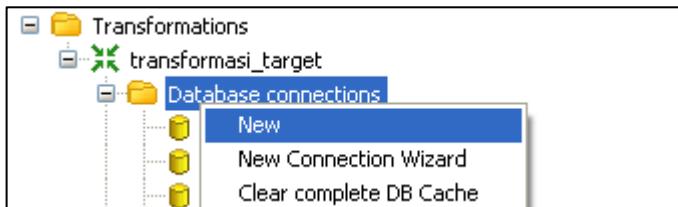
This screenshot shows the 'Examine preview data' dialog. It displays a table with 7 rows of data. The columns are labeled '#', 'Kode Cabang', 'Kode Produk', 'Tahun', 'Jan', and 'Feb'. All rows show the same values: Kode Cabang is 'CABANG-039', Kode Produk is 'PROD-0000001', Tahun is '2008.0', Jan is '32637383.0', and Feb is '29'. The dialog has a blue header bar with the title 'Examine preview data'.

#	Kode Cabang	Kode Produk	Tahun	Jan	Feb
1	CABANG-039	PROD-0000001	2008.0	32637383.0	29
2	CABANG-039	PROD-0000002	2008.0	18434034.0	16
3	CABANG-039	PROD-0000003	2008.0	19058922.0	19
4	CABANG-039	PROD-0000004	2008.0	35538965.0	24
5	CABANG-039	PROD-0000005	2008.0	26668799.0	20
6	CABANG-039	PROD-0000006	2008.0	39401262.0	35
7	CABANG-039	PROD-0000007	2008.0	16702554.0	1

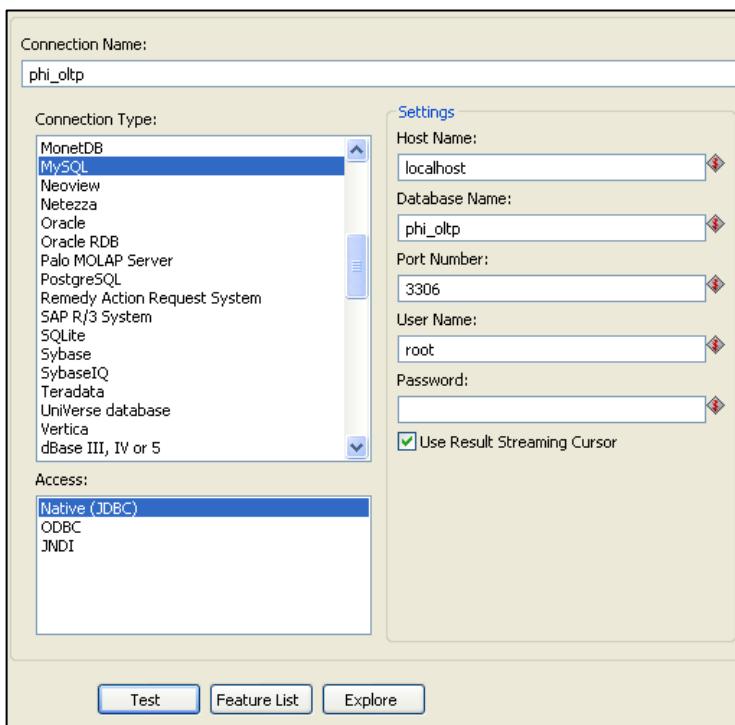
- Lakukan hal yang sama untuk step **Rubah Data** dan perhatikan perbedaannya.

7.16 Praktek: Membuat Koneksi Database ke MySQL

- Pada bagian **View** di panel kiri Spoon terdapat titik **node Database Connections**, klik kanan pada **node** tersebut dan pilih **New**.



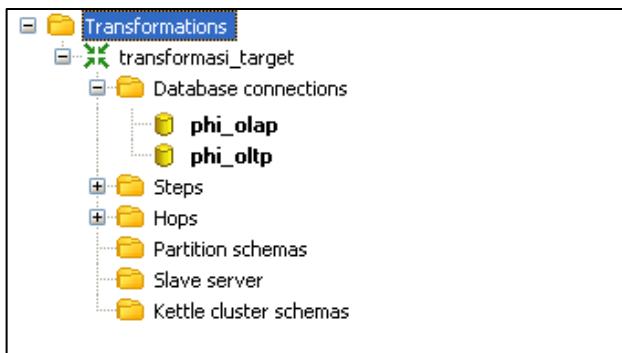
- Masukkan konfigurasi koneksi ke database MySQL sebagai berikut. Database sudah harus ada sebelumnya.



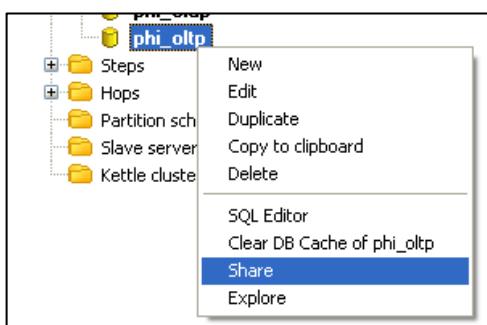
- Klik tombol **Test**, apabila koneksi berhasil dilakukan akan muncul *window* seperti gambar berikut.



- Klik tombol **OK** untuk menutup *window* tersebut.
- Koneksi tersebut akan muncul pada daftar **Database Connections**.



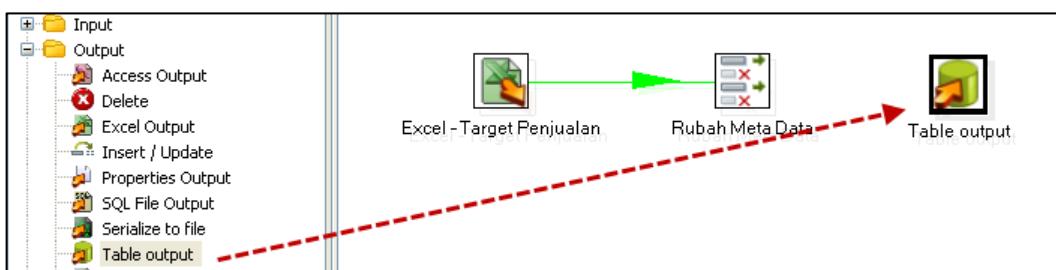
- Klik kanan pada koneksi yang baru kita buat tadi, dan klik **Share**. Opsi ini memungkinkan koneksi dapat digunakan oleh seluruh *job / transformation* kita tanpa harus dideklarasikan ulang.



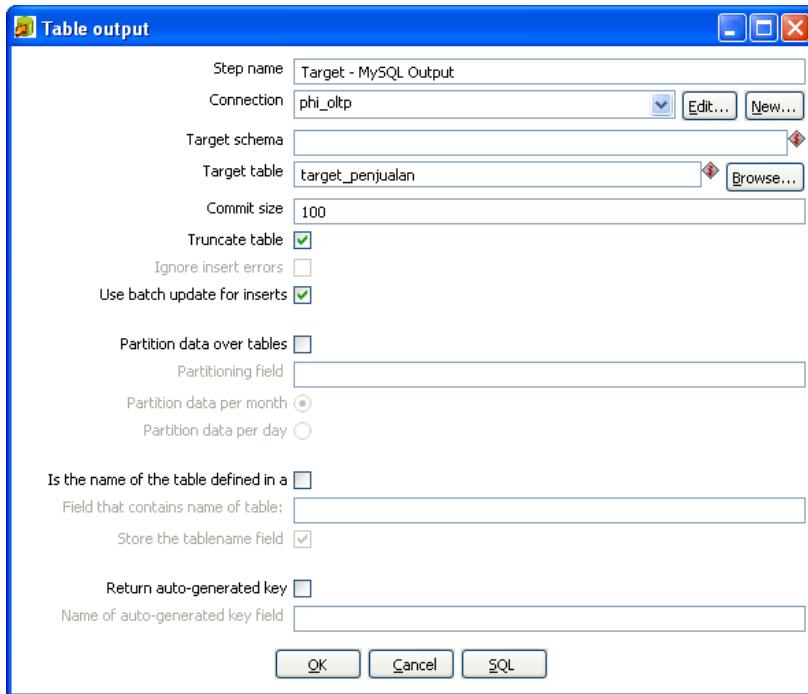
- Simpan *transformation* kita dengan cara mengklik tombol **Save**.
- Selesai.

7.17 Praktek: Menulis Data ke Tabel MySQL

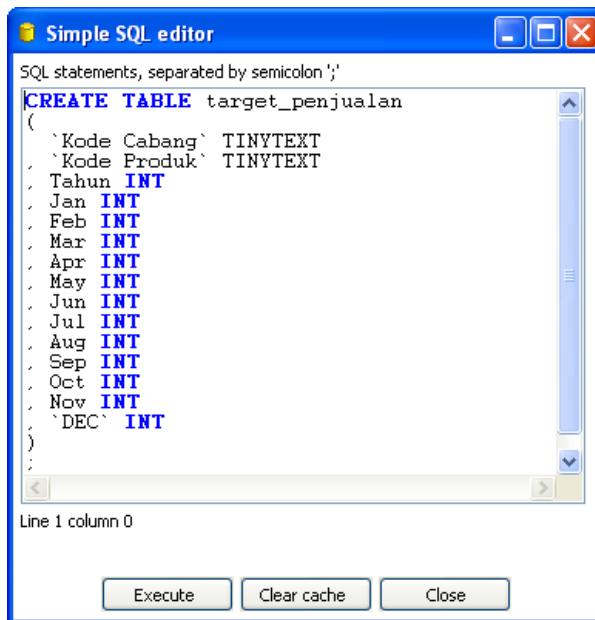
- Tambahkan step **Table Output** pada ruang kerja *transformation* kita



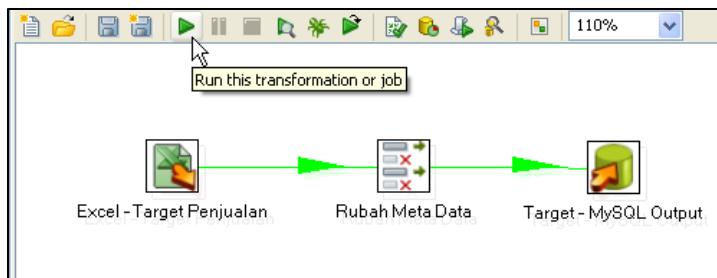
- Hubungkan step **Rubah Meta Data** dengan **Table Output**.
- Klik 2 kali pada step **Table Output**.
- Masukkan konfigurasi untuk dialog step tersebut sebagai berikut :
 - Step Name** : Target - MySQL Output
 - Connection** : phi.oltp
 - Target table** : target_penjualan
 - Commit Size** : 100
 - Truncate table** : yes



- Karena tabel **target_penjualan** belum ada di database kita maka kita perlu membuatnya terlebih dahulu. Tapi jangan kuatir, pada step ini terdapat tombol "SQL" yang dapat digunakan untuk menghasilkan perintah DDL SQL untuk membuat / merubah tabel output.
- Klik tombol **SQL**, akan terlihat perintah DDL **CREATE TABLE target_penjualan....**



- Klik tombol **Execute** untuk melakukan mengeksekusi perintah tersebut pada database kita. Klik tombol **OK** pada dialog yang muncul kemudian.
- Pada tahap ini tabel **target_penjualan** akan dibentuk di database **phi_oltp**.
- Klik **OK** beberapa kali untuk menutup semua dialog dan kembali ke ruang kerja kita.
- Jalankan *transformation* kita dengan cara klik tombol **Run this transformation or job**.



- Klik tombol **Launch** pada dialog **Execute a transformation**.



- Transformasi data akan berlangsung, tunggu beberapa saat sampai selesai – ditandai dengan status **Finished** pada tab **Step Metrics** yang berada pada bagian panel bawah ruang kerja.

	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active
0	0	123	0	0	0	0	0	0	Finished
0	123	123	0	0	0	0	0	0	Finished
0	123	123	0	123	0	0	0	0	Finished

Event	Date
2009/06/01 05:36:18 - org.pentaho.di.ui.core.database.dialog.SQLEditor -)	2009/06/01 05:36:18
2009/06/01 05:36:19 - phi_lotp - Connection to database closed!	2009/06/01 05:36:19
2009/06/01 05:36:21 - phi_lotp - New database connection defined	2009/06/01 05:36:21
2009/06/01 05:36:21 - phi_lotp - Connected to database.	2009/06/01 05:36:21
2009/06/01 05:36:21 - phi_lotp - Connection to database closed!	2009/06/01 05:36:21
2009/06/01 05:36:50 - Spoon - Save to file or repository...	2009/06/01 05:36:50
2009/06/01 05:36:50 - User preferences - Save properties for Spoon...	2009/06/01 05:36:50
2009/06/01 05:36:50 - SharedObjects - Reading the shared objects file [file:///C:/Documents and Settings/Feris/.kettle/sharedobjects]	2009/06/01 05:36:50
2009/06/01 05:36:50 - DBCache - We wrote 12 cached rows to the database cache!	2009/06/01 05:36:50
2009/06/01 05:40:01 - SharedObjects - Reading the shared objects file [file:///C:/Documents and Settings/Feris/.kettle/sharedobjects]	2009/06/01 05:40:01
2009/06/01 05:40:01 - org.pentaho.di.ui.spoon.trans.TransGraph - Transformation opened.	2009/06/01 05:40:01
2009/06/01 05:40:01 - Spoon - Launching transformation [transformasi_target]...	2009/06/01 05:40:01
2009/06/01 05:40:01 - Spoon - Started the transformation execution.	2009/06/01 05:40:01

- Dengan **SQLYog**, cobalah lihat isi data dari table **target_penjualan**.

7.18 Praktek: Melihat Log

- Lihat log dari hasil eksekusi *transformation* kita di atas pada tab **Logging** di panel **Execution Results**.
- Terdapat tiga tombol pada tab **Logging**:
 - Show error lines : menunjukkan hanya pesan kesalahan (jika ada)
 - Clear log : membersihkan layar log dari tampilan sebelumnya
 - Log Settings : memasukkan filter dan rincian log yang diinginkan
- Ubahlah log setting ke pilihan **LogLevel = Rowlevel (very detailed)**.
- Klik tombol **Clear log** untuk membersihkan tampilan.

- Jalankan kembali transformasi sampai selesai. Bandingkan perbedaan log saat ini dengan hasil eksekusi sebelumnya.

The screenshot shows the 'Execution Results' panel with the following log output:

```

Execution Results
Execution History Logging Step Metrics Performance Graph

2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000031], [2008], [5402331]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000032], [2008], [1884771]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000033], [2008], [5392660]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000034], [2008], [6985108]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000035], [2008], [2386077]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000036], [2008], [6834351]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000037], [2008], [1670484]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000038], [2008], [2847817]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000039], [2008], [1776465]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000040], [2008], [4235263]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Written row: [CABANG-065], [PROD-0000041], [2008], [5219932]
2009/06/01 05:55:29 - phi_oltp - Commit on database connection [phi_oltp]
2009/06/01 05:55:29 - Target - MySQL Output.0 - Signaling 'output done' to 0 output rowsets.
2009/06/01 05:55:29 - phi_oltp - Commit on database connection [phi_oltp]
2009/06/01 05:55:29 - phi_oltp - Connection to database closed!
2009/06/01 05:55:29 - Target - MySQL Output.0 - Finished processing (I=0, O=123, R=123, W=123, U=123, E=0)
2009/06/01 05:55:29 - Spoon - The transformation has finished!

```

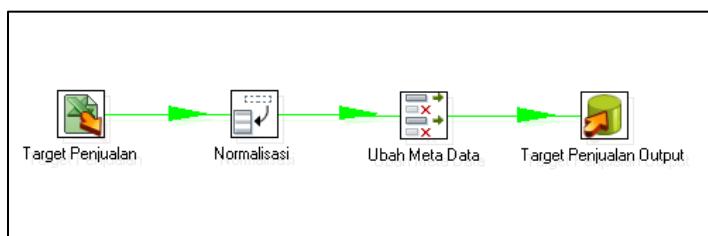
7.19 Praktek: Normalisasi Data

- Tambahkan step **Row Normaliser** pada ruang kerja kita.
- Namakan step tersebut **Normalisasi** - sehingga jika kita preview data pada step tersebut akan tampak seperti di bawah ini. (Instruktur perlu menerangkan cara penggunaan dari step ini terlebih dahulu).

The screenshot shows the 'Examine preview data' window with the following preview data:

#	Kode Cabang	Kode Produk	Tahun	Bulan	Total
1	CABANG-039	PROD-0000001	2008	1	32637383.0
2	CABANG-039	PROD-0000001	2008	2	29651096.0
3	CABANG-039	PROD-0000001	2008	3	55384720.0
4	CABANG-039	PROD-0000001	2008	4	42483638.0
5	CABANG-039	PROD-0000001	2008	5	30497420.0
6	CABANG-039	PROD-0000001	2008	6	48225162.0
7	CABANG-039	PROD-0000001	2008	7	33749703.0
8	CABANG-039	PROD-0000001	2008	8	39221638.0
9	CABANG-039	PROD-0000001	2008	9	37832077.0
10	CABANG-039	PROD-0000001	2008	10	41541044.0

- Ubah meta data untuk kolom Total sehingga menjadi Integer tanpa angka pecahan dengan step **Select Values**. Namakan step tersebut dengan **Ubah Meta Data**.
- Simpan data hasil transformasi ke dalam tabel **target_penjualan** pada database **phi_minimart** di MySQL.
- Jalankan *transformation* tersebut dan coba analisa output yang muncul panel "Execution Results". Apa yang dapat Anda simpulkan ?



7.20 Email Step

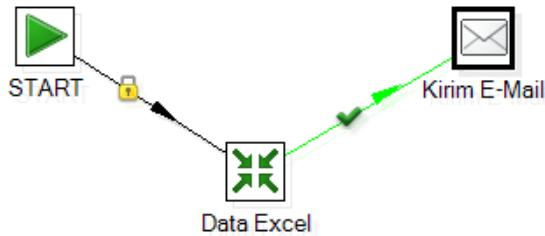
Step yang berkaitan dengan email ada 2 pada Kettle yaitu:

- **Mail:** digunakan untuk mengirimkan pesan dengan atau tanpa *file attachment*. Step ini sangat berguna untuk membantu kita melakukan kontrol atas tugas ETL kita. Kita dapat menggabungkan *error handling* ataupun *conditional hop* dengan notifikasi email.
- **Get Mails From POP3:** digunakan untuk mendapatkan pesan email melalui protokol POP3.

Pada praktik berikutnya, kita akan coba hasilkan file Excel yang langsung dikirimkan melalui email.

7.21 Praktek: Menghasilkan dan Mengirimkan File Excel via Email

- Buka file **kirim_email.kjb** pada folder contoh-file yang ada pada DVD training Anda.



- Lakukan konfigurasi email sesuai petunjuk instruktur dan jalankan transformation tersebut. Apabila semua berjalan baik maka transformation ini akan mengirimkan data master produk berupa file Excel dengan nama **data_excel_yyyy-MM-dd.xls**, dimana yyyy mewakili tahun, MM mewakili bulan, dan dd mewakili hari.

Bab 8. Repozitori Kettle

8.1 Penyimpanan Objek-objek Kettle

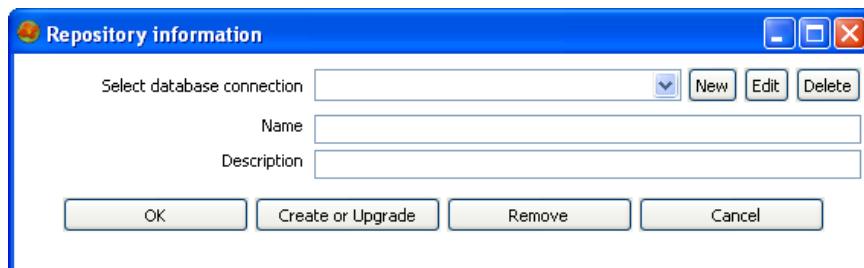
Penyimpanan job, *transformation* dan *object-object* lainnya di Kettle sebenarnya dapat dilakukan pada dua media, yaitu:

- Pada filesystem dengan format file berupa text xml. File tersebut disimpan dengan ekstensi **.kjb** untuk job dan **.ktr** untuk *transformation*.
- Pada table-table database relasional yang didukung oleh Java (JDBC).

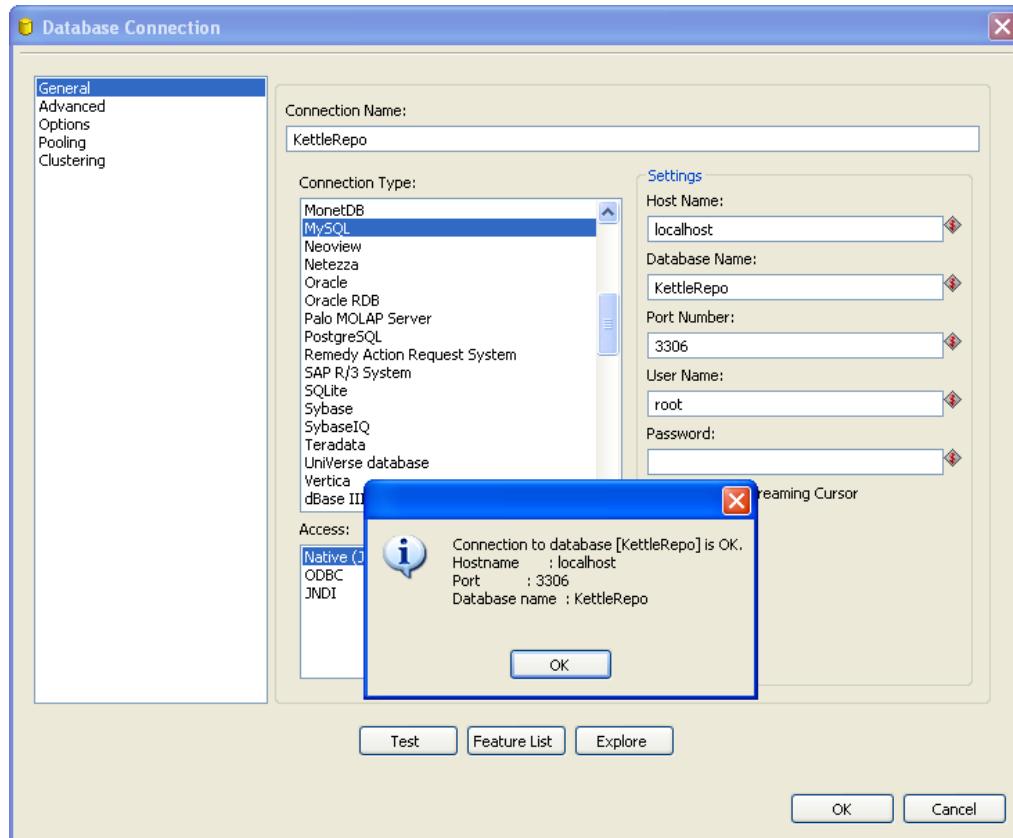
Untuk penyimpanan dengan filesystem dengan format XML, sebenarnya telah kita lakukan pada semua latihan sebelumnya. Jika kita ingin menyimpan dalam bentuk file xml, maka pada saat menjalankan Kettle kita memilih opsi **No Repository**.

8.2 Praktek: Buat Database Repository

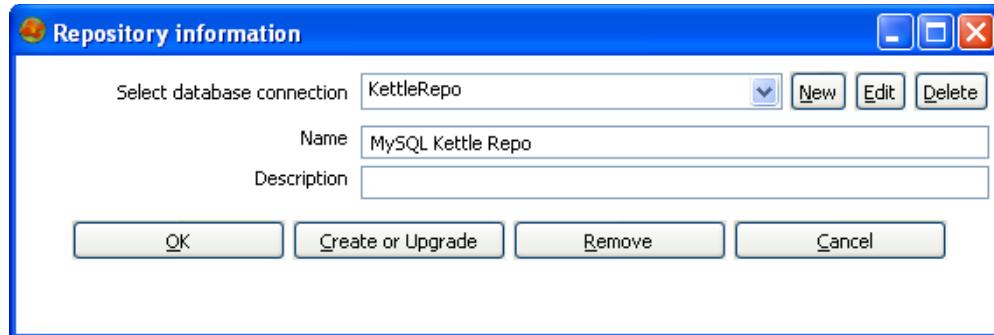
- Buat database **KettleRepo** pada MySQL.
- Jalankan Spoon, pilih opsi **New** pada dialog **Select a Repository** atau ...
- Jika sudah dalam lingkungan Spoon, maka pilih menu **Repository | Connect to repository** dan pilih opsi **New** pada dialog **Select a Repository**.



- Pada dialog **Repository Information** klik tombol **New**.
- Isi konfigurasi dari dialog **Database Connection** dan tes konesinya apakah berhasil atau tidak. Klik **OK** setelah semuanya berhasil.



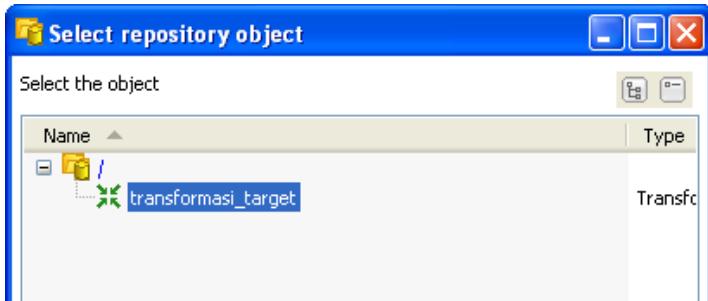
- Isi dialog **Repository Information** dengan entri berikut ini. Klik tombol **Create or Upgrade**.



- Klik **Yes** untuk pertanyaan **Are you sure you want to create the repository on the specified database connection ?**.
- Tunggu beberapa saat sampai muncul dialog **Kettle created the repository on the specified connection.**
- Jalankan kembali Spoon dan pilih repository **MySQL Kettle Repo**.
- Login dengan pasangan user / password berikut: **admin / admin**.
- Selesai.

8.3 Praktek: Menyimpan Transformasi pada Repository KettleRepo

- Buka file transformasi **transformasi_target.ktr** yang sebelumnya telah kita buat dengan cara pilih menu **File | Import from an XML file**.
- Tutup *transformation* tersebut.
- Buka kembali transformasi tersebut, namun kali ini melalui repositori.



- Eksekusi *transformation* tersebut, apakah ada error yang terjadi ?

8.4 Praktek: Eksplorasi Repositori

- Pada Spoon cobalah melakukan eksplorasi repository dengan menu **Repository → Explore Repository**.
- Lakukan navigasi dan munculkan contextual menu dari tiap object yang diamati.
- Cobalah edit user dan password melalui dialog tersebut.

Bagian 2

Multi Dimensional Modelling



Bab 9. Terminologi dan Konsep Dasar

9.1 Apa itu Multidimensional Modelling ?

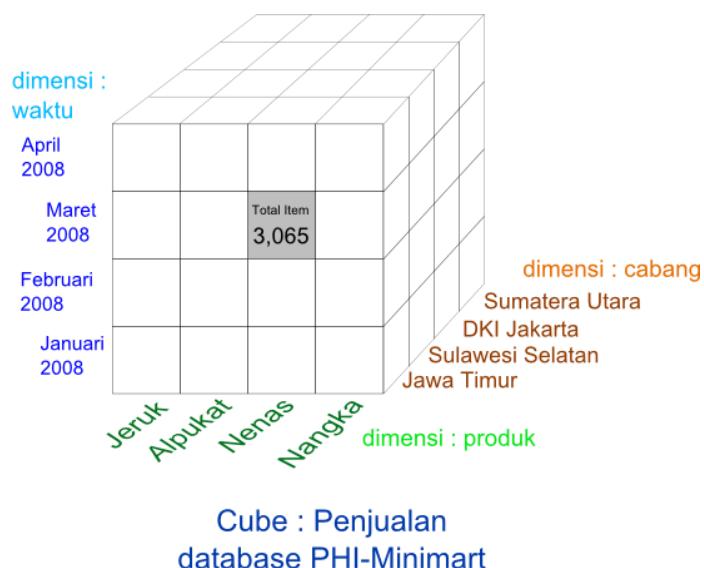
Multidimensional Modelling (MDM) adalah suatu kumpulan konsep dan teknik yang digunakan dalam merancang data warehouse dengan penekanan pada dua jenis tabel konseptual yaitu **fact** – yang berisi nilai pengukuran – dan **dimension** – yang memberikan konteks. Pemodelan ini dipopulerkan oleh Ralph Kimball.

9.2 OLAP : Cube, Dimension, Measure dan Member

Teknologi OLAP dewasa ini secara umum sangat *bersahabat* dengan perancangan *multi dimensional modelling*.

Di dalam OLAP ini kita perlu mengenal berbagai peristilahan dan konsep sebagai berikut:

- **Cube**: adalah struktur data dalam bentuk array multi dimensional, terdiri dari *dimension* dan *measure*. Tiap cube biasanya mencakup domain atau aspek bisnis tertentu seperti penjualan.
- **Dimension / Dimensi**: adalah view / sudut pandang yang menyusun cube. Dimensi dapat terdiri dari berbagai level. Pada gambar di bawah terdapat 3 dimensi yaitu dimensi waktu (April, May, June, July), dimensi area (Boston, Benson, Seattle, Wichita) dan dimensi produk (Sardines, Anchovies, Herring, Pilchards).
- **Measure**: Nilai pengukuran.
- **Member**: Isi / anggota dari suatu dimension / measure tertentu.



9.3 Fact and Dimension Tables

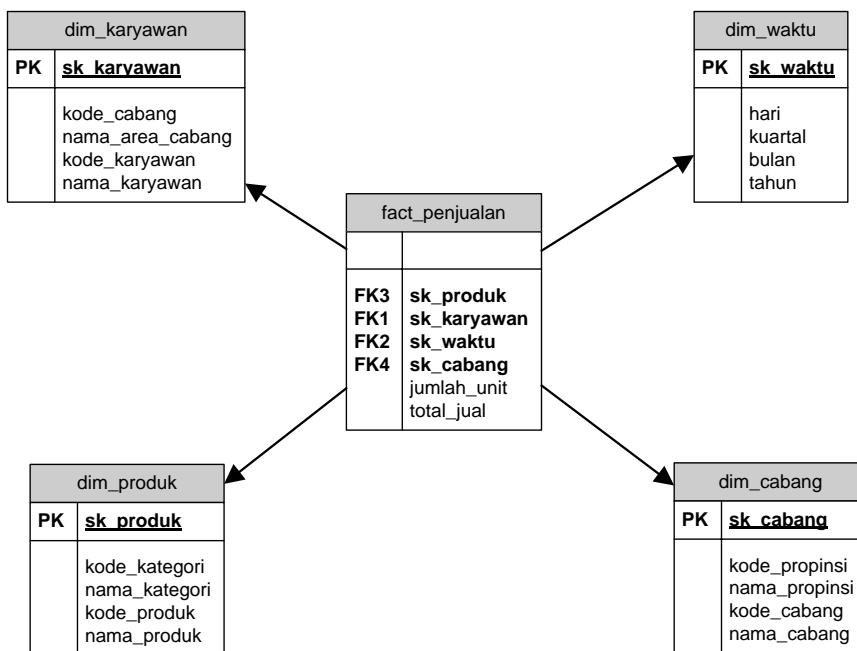
Di dalam model multidimensional, database terdiri dari beberapa **tabel fakta (fact tables)** dan **tabel dimensi (dimension tables)** yang saling terkait. Suatu tabel fakta berisi berbagai nilai agregasi yang menjadi dasar pengukuran (*measure*) serta beberapa key yang terkait ke tabel dimensi yang akan menjadi sudut pandang dari *measure* tersebut.

Dalam perkembangannya, susunan fact table dan dimension table ini memiliki standar perancangan atau schema karena terbukti meningkatkan performa dan kemudahan dalam penerjemahan ke sistem OLAP.

Schema inilah yang menjadi dasar untuk melakukan data warehousing. Dua schema yang paling umum digunakan oleh berbagai OLAP engine adalah skema bintang (***Star Schema***) dan skema butir salju (***Snowflake Schema***).

9.4 Skema Bintang (***Star Schema***)

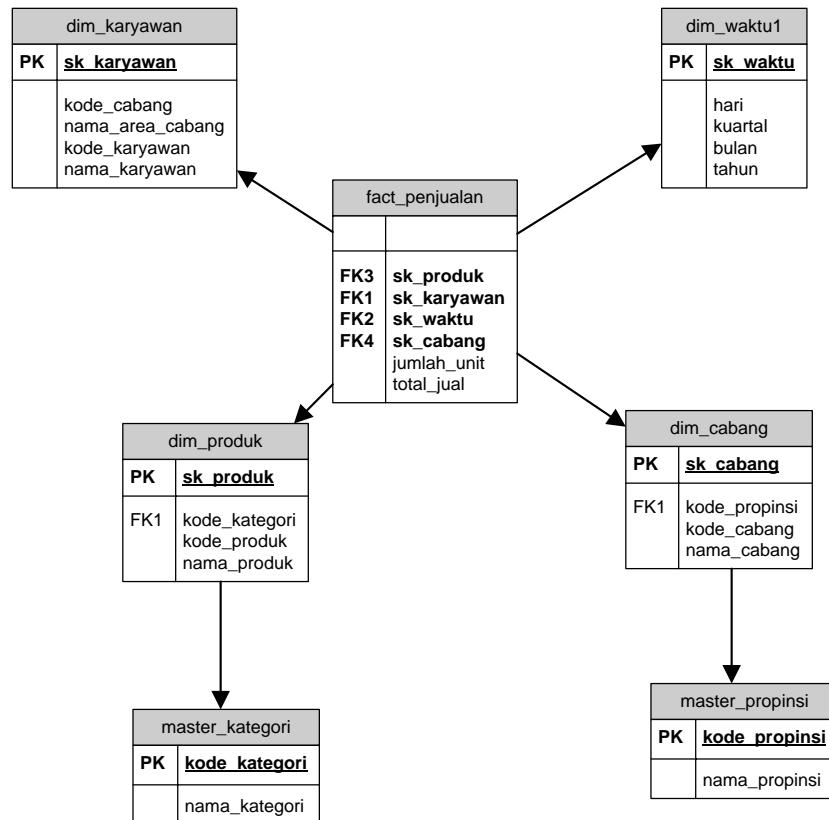
Star schema berpusat pada satu tabel fakta yang dikelilingi oleh satu atau beberapa tabel dimensi sebagai 'cabang'-nya sehingga kelihatan seperti bintang. Setiap percabangan berhenti pada satu tabel dimensi. Atau dengan kata lain tabel dimensi dengan schema ini semuanya berupa *leaf* atau daun dan tidak memiliki percabangan lain.



Star Schema dari PHI-Minimart

9.5 Skema Butir Salju (Snowflake Schema)

Berbeda dengan *Star*, skema *Snowflake* memiliki percabangan pada tabel dimensinya dan biasanya dioptimalkan karena pertimbangan ruang penyimpanan. Namun seiring berkembangnya waktu dan struktur yang kurang bersahabat bagi kebanyakan OLAP engine, maka skema ini kurang diadopsi oleh perancang *data warehouse*.



Snow Flake Schema dari PHI-Minimart

9.6 Surrogate Key

Surrogate key adalah key / kolom data di tabel dimensi yang menjadi primary key dari tabel tersebut.

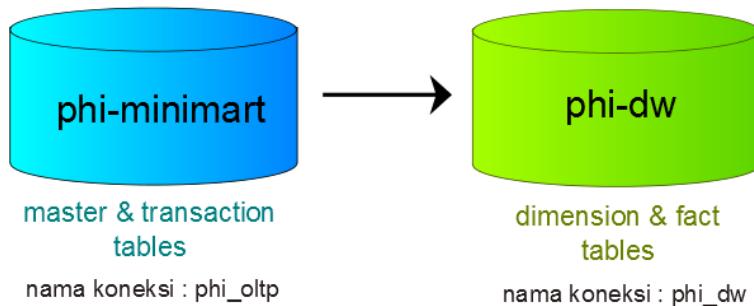
Nilai dari key ini biasanya berupa nilai sekuensial atau random unik dan tidak memiliki arti jika dilihat dari proses bisnis dari mana sumber data berasal. Ini kebalikan dari OLTP dimana kolom primary key dari sumber data OLTP memiliki nilai yang mempengaruhi proses bisnis.

Namun demikian, surrogate key ini sangat penting untuk konsistensi analisis, terutama pada saat terkait transaksi historis.

Bab 10. Multi Dimensional Modelling dengan Kettle

10.1 Petunjuk Umum Pembuatan Tabel Dimensi PHI-Minimart

- Objektif dari latihan ini adalah data warehouse **phi_dw** dengan pemodelan skema bintang dengan database sumber **phi_minimart**.



- Buat satu *job* c:\kettletraining\main.kjb yang terdiri dari beberapa file *transformation* dengan tugas berikut:
 - Melakukan populasi data dimensi produk, cabang dan karyawan dengan strategi repopulasi ulang.
 - surrogate key dari tiap dimensi bersifat sekuensial dengan menggunakan step **Add Sequence** pada Kettle atau **auto sequence key** pada database.



10.2 Praktek: Membuat Database PHI_DW

- Jalankan aplikasi **SQLYog**.
- Buatlah satu database MySQL dengan nama **phi_dw** yang akan berisi rancangan data warehouse dengan konsep multidimensional modelling (berisi table fact dan dimension).
- Selesai.

10.3 Praktek: Membuat Koneksi database PHI_DW pada Spoon

- Pada Spoon buatlah dua koneksi database yang bernama **phi_dw** yang merujuk ke database MySQL **phi_dw**.
- Share koneksi database tersebut sehingga dapat digunakan sebagai referensi koneksi secara global oleh seluruh *job* / *transformation* kita.
- Selesai.

10.4 Praktek: Dimensi Produk

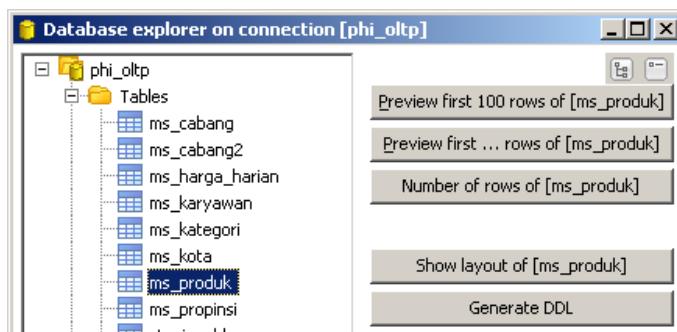
- Buat satu *transformation* baru dan simpan dengan nama **c:\kettletraining\dim_produk.ktr**.
- Masukkan step **Table Input** pada ruang kerja kita. Klik dua kali pada step tersebut.



- Namakan step tersebut dengan nama **ms_produk** dan pilih koneksi **phi_oltp**.



- Klik tombol **Get SQL Select Statement**.
- Akan muncul dialog **Database Explorer** yang terdiri dari dua bagian:
 - Pada panel kiri / object explorer kita bisa melakukan navigasi dan pemilihan objek-objek database (table, view dan lain-lain).
 - Tombol-tombol pada bagian kanan panel yang dapat kita gunakan untuk melihat/mengolah table / view yang kita pilih.

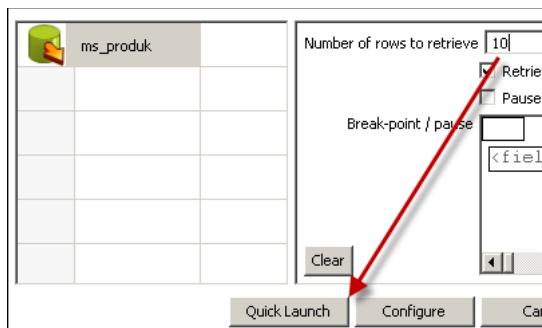


- Pilih objek table **ms_produk** dan klik tombol **OK**.
- Kita akan dibawa kembali ke dialog konfigurasi dari "Table Input" dengan input SQL terisi dengan perintah "Select * from ms_produk". Spoon akan menanyakan apakah kita ingin lebih rinci dengan memasukkan semua *field* melalui prompt "**Do you want to include the field-names in the SQL?**". Klik tombol "**Yes**" dan perhatikan perubahan pada input SQL kita.

```
SQL
SELECT *
FROM ms_produk
```

Question?
Do you want to include the field-names in the SQL?
Yes No Cancel

- Klik tombol **OK** untuk menutup dialog **Table Input** tersebut.
- Kembali ke ruang kerja kita, klik **step** yang telah kita namakan dengan **ms_produk** tersebut, diikuti dengan mengklik icon / tombol preview yang ada pada menu **toolbar** ().
- Masukkan nilai **10** pada input **Number of rows to retrieve** kemudian klik tombol **Quick Launch**.



- Dialog **Examine preview data** akan muncul dengan jumlah **10 row data** seperti terlihat pada gambar berikut.

Examine preview data				
Rows of step: ms_produk (10 rows)				
#.	kode_item	kode_produk	kode_kategori	nama_produk
1	ITM-001	PROD-0000001	KTG-003	bawang merah 1
2	ITM-002	PROD-0000002	KTG-003	bawang putih 1
3	ITM-003	PROD-0000003	KTG-003	buncis 1 kg
4	ITM-004	PROD-0000004	KTG-003	kacang panjang
5	ITM-005	PROD-0000005	KTG-003	kacang hijau 1 kg
6	ITM-006	PROD-0000006	KTG-003	kentang 1 kg
7	ITM-007	PROD-0000007	KTG-003	ketimun 1 kg
8	ITM-008	PROD-0000008	KTG-003	kobis 1 kg
9	ITM-009	PROD-0000009	KTG-003	tomat 1 kg
10	ITM-010	PROD-0000010	KTG-003	ubi kayu basah 1

Tips Spoon

Setiap hasil output dari *preview data* dapat dimasukkan ke dalam ruang kerja / *worksheet* Excel dengan mekanisme *copy paste*.

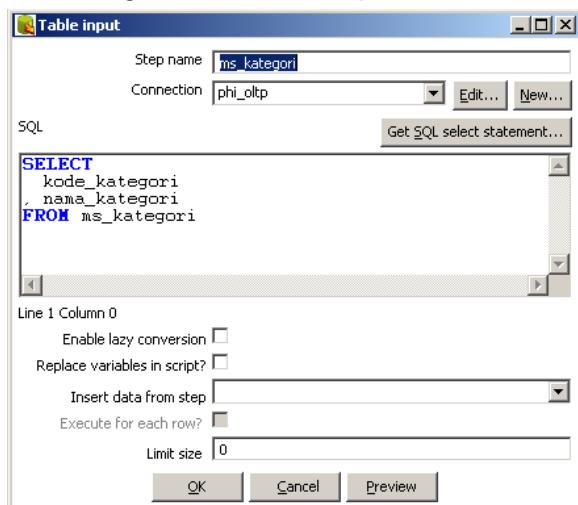
Contoh *copy paste* dari hasil output preview contoh ms_produk terlihat seperti pada gambar berikut.

	A	B	C	D	E
1	kode_item	kode_produk	kode_kategori	nama_produk	unit
2	ITM-001	PROD-0000001	KTG-003	bawang merah 1000000 mg	
3	ITM-002	PROD-0000002	KTG-003	bawang putih 1 kg	
4	ITM-003	PROD-0000003	KTG-003	buncis 1 kg	
5	ITM-004	PROD-0000004	KTG-003	kacang panjang 1 kg	
6	ITM-005	PROD-0000005	KTG-003	kacang hijau 1 kg	
7	ITM-006	PROD-0000006	KTG-003	kentang 1 kg	
8	ITM-007	PROD-0000007	KTG-003	ketimun 1 kg	
9	ITM-008	PROD-0000008	KTG-003	kobis 1 kg	
10	ITM-009	PROD-0000009	KTG-003	tomat 1 kg	
11	ITM-010	PROD-0000010	KTG-003	ubi kayu basah 1 kg	

Sebenarnya hal ini tidak terbatas pada *data preview*, Anda dapat melakukan hal yang sama pada setiap dialog yang memiliki *data grid* atau *table*.

Dan jika dialog grid berupa *input list*, kita dapat melakukan hal sebaliknya yaitu melakukan *copy paste* dari table Excel ke table entri pada dialog Spoon.

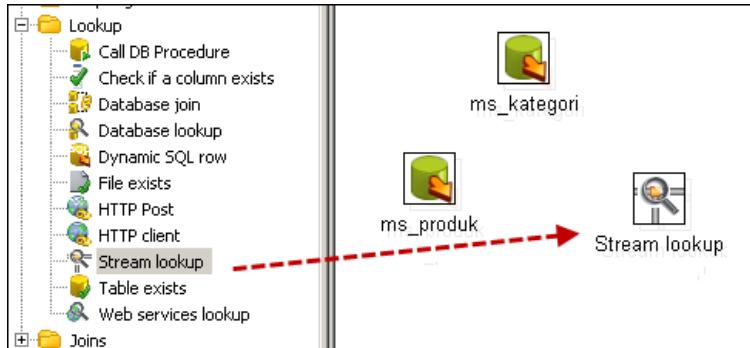
- Masukkan kembali satu **step Table Input**, kali ini dengan mengambil semua *field* dari table ms_kategori. Namakan step tersebut sesuai input tablenya, yaitu **ms_kategori**.



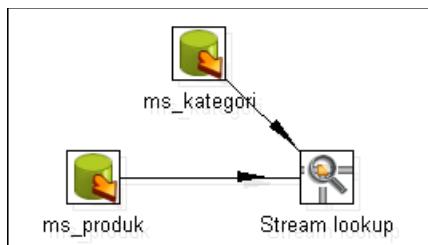
- Lakukan *preview* pada step tersebut.

Rows of step: ms_kategori (3 rows)		
#.	kode_kategori	nama_kategori
1	KTG-001	Buah
2	KTG-002	Makanan & Minuman
3	KTG-003	Sayur-Sayuran

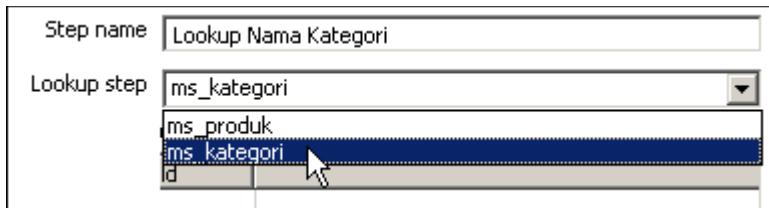
- Masukkan step **Stream Lookup** dari kategori **Lookup** ke dalam ruang kerja kita.



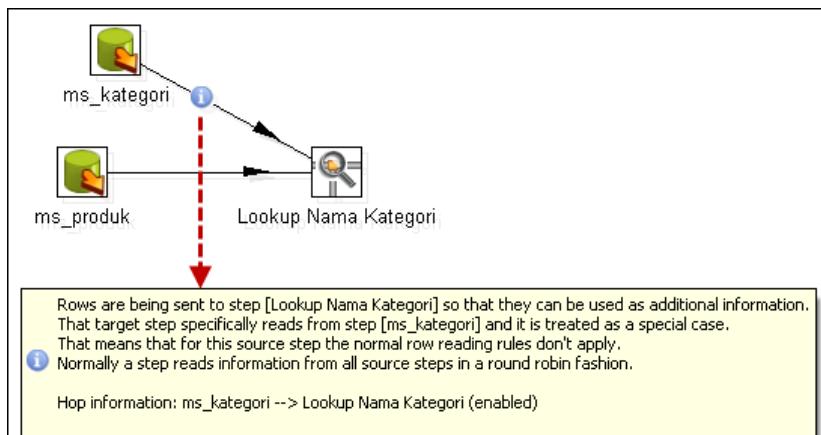
- Buat dua hop untuk menghubungkan ketiga step yang sudah dimasukkan, yaitu dari step **ms_produk** menuju **Stream Lookup** dan dari step **ms_kategori** menuju **Stream Lookup**.



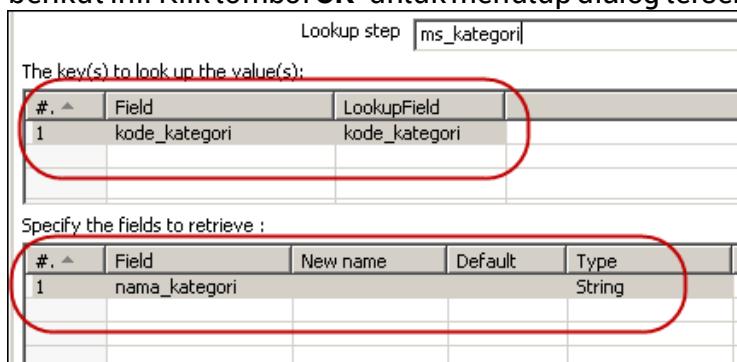
- Klik dua kali pada step **Stream Lookup** dan masukkan konfigurasi berikut :
 - Step name : Lookup Nama Kategori
 - Lookup step : **ms_kategori**



- Klik tombol **OK** untuk menutup dialog **Stream Lookup**.
- Kembali ke ruang kerja, perhatikan perubahan yang terjadi. Terlihat pada hop antara step **ms_kategori** dan **Lookup Nama Kategori** terdapat **icon referensi / information** (info icon). Arahkan kursor mouse Anda ke icon tersebut dan baca keterangannya. Apa yang dapat Anda simpulkan?



- Klik dua kali pada step **Lookup Nama Kategori** dan masukkan konfigurasi seperti pada gambar berikut ini. Klik tombol **OK** untuk menutup dialog tersebut.



Tips Spoon

Pada beberapa step Kettle seperti **Stream Lookup** terdapat tombol **Get Fields**, **Get Lookup Fields** yang digunakan untuk mengisi entri nama field pada input grid sehingga memudahkan kita memilih dengan menghapus yang tidak perlu.



Gambar berikut menunjukkan daftar nama field yang dihasilkan pada step **Lookup Nama Kategori** apabila kita mengklik tombol **Get Fields** dan **Get lookup fields**.

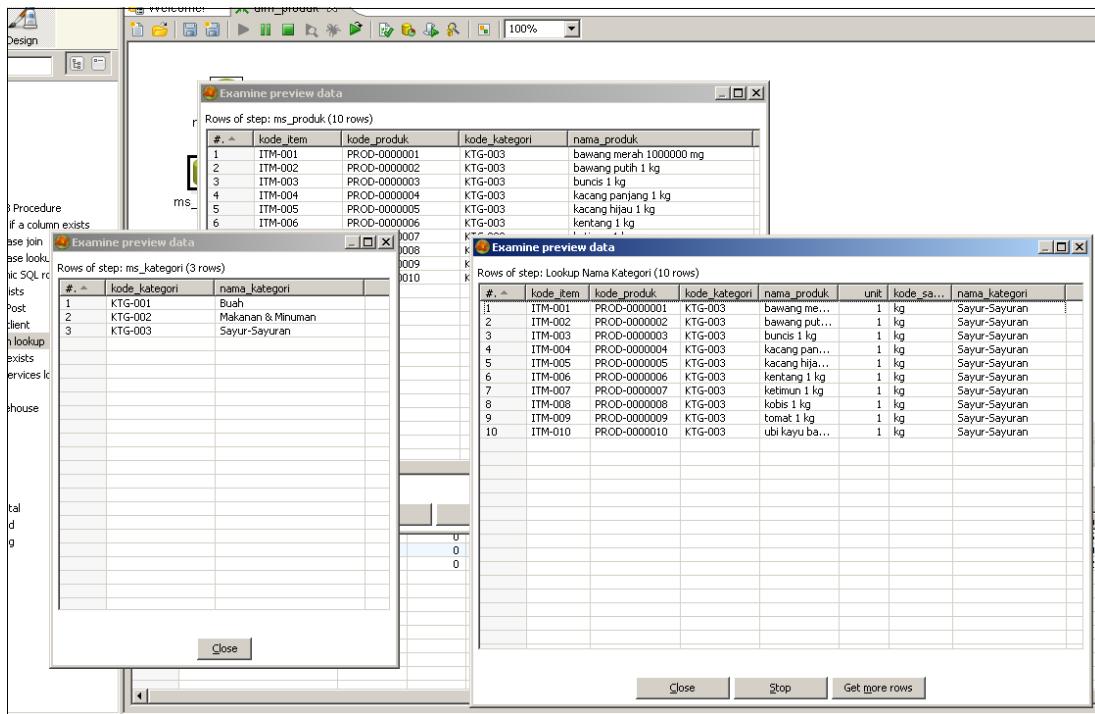
The key(s) to look up the value(s):			
#.	Field	LookupField	
1	kode_item	kode_item	
2	kode_produk	kode_produk	
3	kode_kategori	kode_kategori	
4	nama_produk	nama_produk	
5	unit	unit	
6	kode_satuan	kode_satuan	

Specify the fields to retrieve :				
#.	Field	New name	Default	Type
1	kode_kategori			String
2	nama_kategori			String

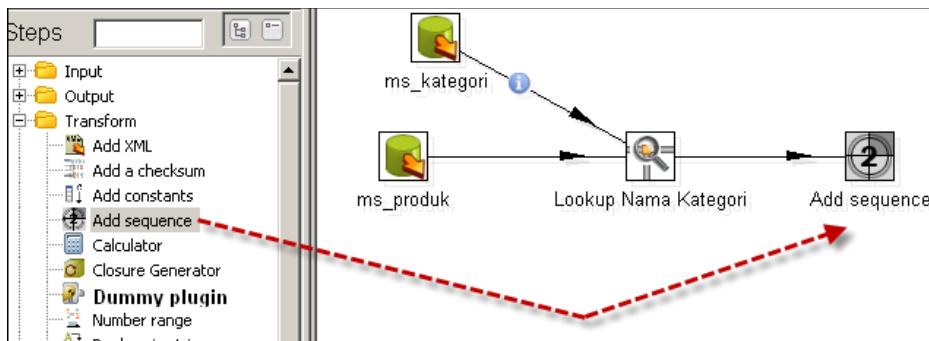
- Lakukan preview data pada step **Lookup Nama Kategori**.

Examine preview data							
Rows of step: Lookup Nama Kategori (10 rows)							
#.	kode_item	kode_produk	kode_kategori	nama_produk	unit	kode_sa...	nama_kategori
1	ITM-001	PROD-0000001	KTG-003	bawang me...	1	kg	Sayur-Sayuran
2	ITM-002	PROD-0000002	KTG-003	bawang put...	1	kg	Sayur-Sayuran
3	ITM-003	PROD-0000003	KTG-003	buncis 1 kg	1	kg	Sayur-Sayuran
4	ITM-004	PROD-0000004	KTG-003	kacang pan...	1	kg	Sayur-Sayuran
5	ITM-005	PROD-0000005	KTG-003	kacang hija...	1	kg	Sayur-Sayuran
6	ITM-006	PROD-0000006	KTG-003	ketimun 1 kg	1	kg	Sayur-Sayuran
7	ITM-007	PROD-0000007	KTG-003	ketimun 1 kg	1	kg	Sayur-Sayuran
8	ITM-008	PROD-0000008	KTG-003	kobis 1 kg	1	kg	Sayur-Sayuran
9	ITM-009	PROD-0000009	KTG-003	tomat 1 kg	1	kg	Sayur-Sayuran
10	ITM-010	PROD-0000010	KTG-003	ubi kayu ba...	1	kg	Sayur-Sayuran

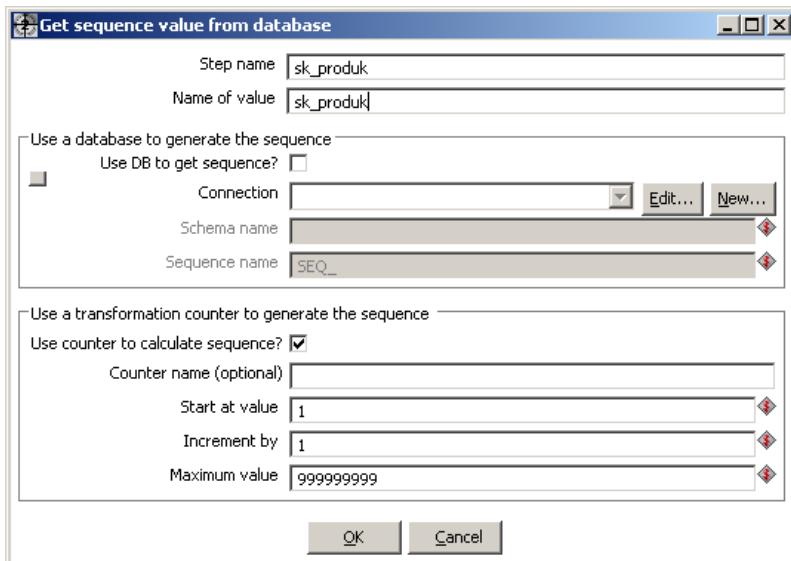
- Kembali lakukan *preview* pada step **ms_produk** dan **ms_kategori** tanpa menutup dialog *preview* di atas. Jika tombol *preview* tidak bisa digunakan, maka klik tombol **stop** pada toolbar. Apa hasil yang Anda dapatkan? Apa yang dapat Anda simpulkan?



- Simpan file *transformation* Anda dengan menggunakan tombol **CTRL + S**. Disarankan untuk sesering mungkin menyimpan file *job* maupun *transformation* Anda pada saat masa perancangan dan pengembangan.
- Tambahkan *step Add Sequence* pada ruang kerja kita. Step ini akan digunakan menghasilkan angka sekuensial / nomor urut sesuai pertambahan row yang didapatkan dari *step* sebelumnya.



- Klik dua kali pada *step Add sequence* dan lakukan konfigurasi seperti pada gambar berikut. Konfigurasi termasuk merubah nama *step* menjadi **sk_produk**.

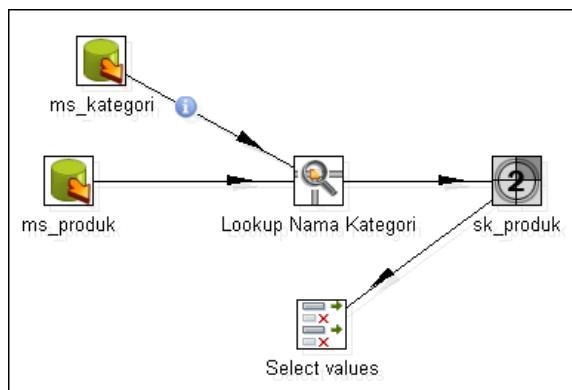


- Klik tombol **OK** untuk menutup dialog.
- Lakukan preview pada *step sk_produk* tersebut. Terlihat kolom **sk_produk** ada pada posisi terakhir dari data preview dengan nilai integer yang terurut.

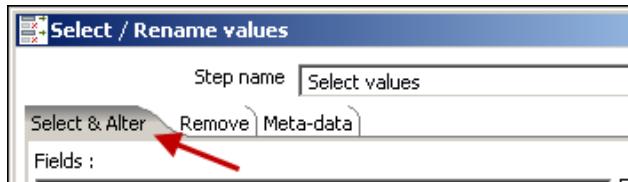
gori	sk_produk
iran	1
iran	2
iran	3
iran	4
iran	5
	6

Langkah kita selanjutnya adalah mengurutkan *field-field* tersebut dengan **sk_produk** dalam posisi paling atas.

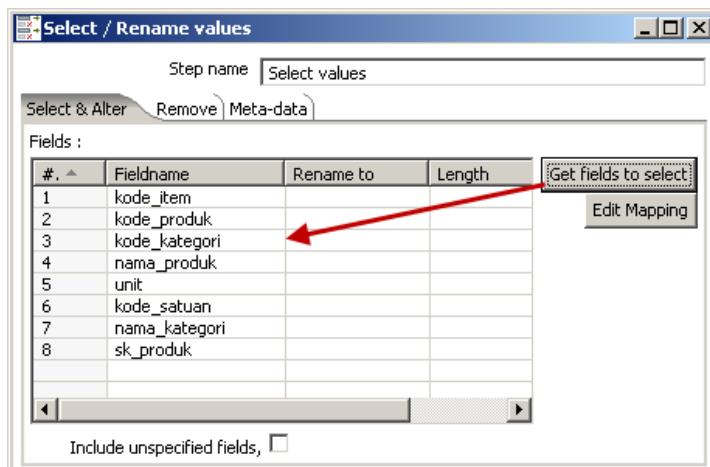
- Tambahkan *step Select Values* dan hubungkan dengan *step sk_produk* seperti terlihat pada gambar berikut.



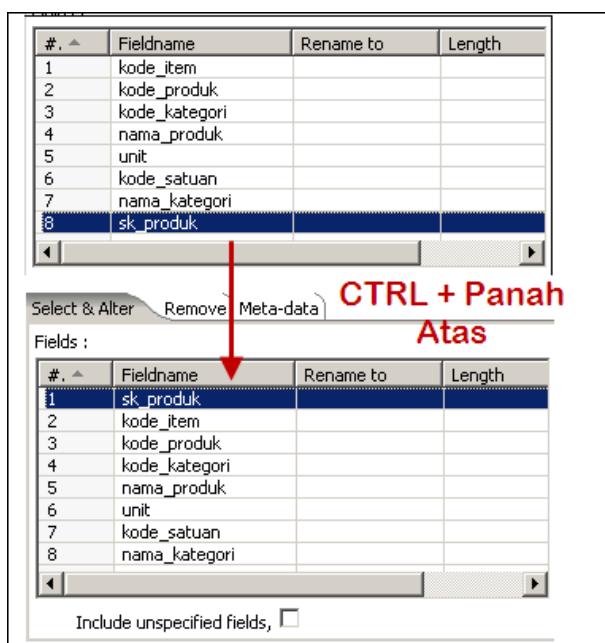
- Klik dua kali pada *step Select values*, pada dialog yang muncul pastikan Anda ada pada tab **Select & Alter**.



- Klik tombol **Get Fields** sehingga nama-nama *field* yang berasal dari step **sk_produk** akan terpopulasi pada tab **Select & Alter**.



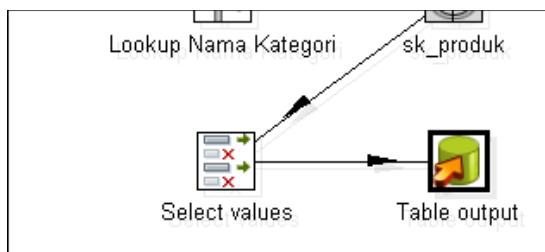
- Klik pada cell pertama pada baris ke delapan sehingga seluruh baris ter-highlight.
- Dengan menekan **CTRL + Panah Atas** berulang kali pada *keyboard* geser *field* **sk_produk** sehingga menempati posisi paling atas.



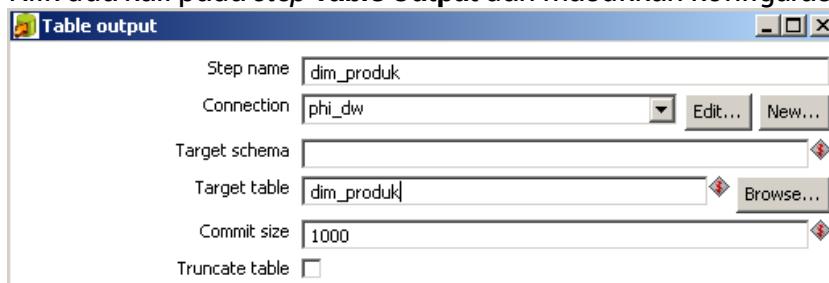
- Susunlah semua *field* sehingga urutannya tampak seperti pada gambar berikut.

#	Fieldname	Rename to	Length	Pr
1	sk_produk			
2	kode_kategori			
3	kode_produk			
4	kode_item			
5	nama_kategori			
6	nama_produk			
7	unit			
8	kode_satuan			

- Klik tombol **OK** untuk menutup dialog.
- Kembali ke ruang kerja, masukkan *step Table Output* dan sambungkan dengan *step Select Values*.



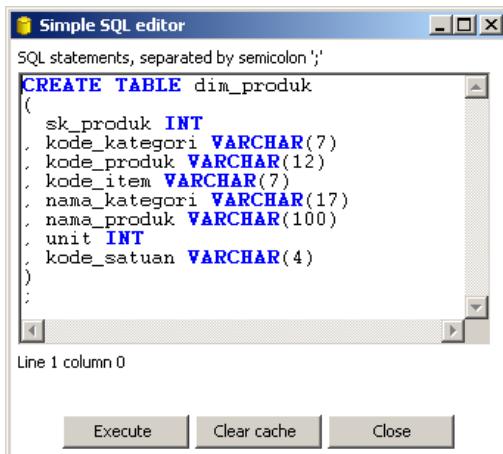
- Klik dua kali pada *step Table Output* dan masukkan konfigurasi seperti berikut.



- Klik tombol **SQL** pada bagian bawah dialog untuk melakukan sinkronisasi antara data yang terdapat pada Kettle dengan **table** yang ada pada database **phi_dw**. Table **dim_produk** belum ada pada database kita.



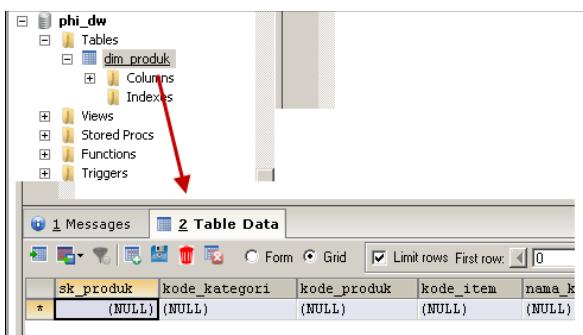
- Akan muncul perintah DDL *Create Table* pada dialog **Simple SQL Editor** yang siap digunakan membuat table **dim_produk**. Klik tombol **Execute** untuk menjalankan perintah SQL tersebut.



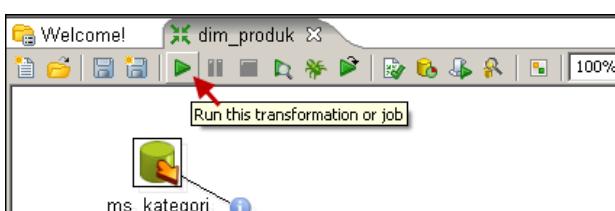
- Tutup dialog dengan mengklik tombol **Close**.
- Klik kembali tombol **SQL**, namun kali ini pesan yang muncul adalah "**No SQL needs to be executed to make this step function properly**". Ini artinya struktur rancangan data flow ETL dan database sudah sinkron sehingga tidak perlu adanya eksekusi DDL lain.



- Tutup semua dialog yang ada.
- Jika kita perhatikan database **phi_dw** dengan menggunakan SQLYog, maka table **dim_produk** sekarang sudah ada namun belum memiliki data.



- Kembali ke Spoon, simpan *transformation* Anda dan kemudian lakukan eksekusi dengan mengklik tombol **Run this transformation or job** pada menu *toolbar*.



- Perhatikan pada panel **Execution Results | Step Metrics** untuk entri **dim_produk** terdapat output bernilai 41. Ini artinya pada step tersebut telah berhasil terjadi penulisan data ke database **phi_dw** sejumlah 41 baris data.

Perhatikan juga, selain output terdapat juga input yang menyatakan berapa banyak row yang dibaca dari table database atau data external. Sedangkan read dan written yang menyatakan jumlah row yang terjadi pada internal Kettle.

#	Stepname	Copynr	Read	Written	Input	Output
1	ms_produk	0	0	41	41	0
2	Lookup Nama Kategori	0	44	41	0	0
3	ms_kategori	0	0	3	3	0
4	sk_produk	0	41	41	0	0
5	Select values	0	41	41	0	0
6	dim_produk	0	41	41	0	41

Jumlah Row yang dimasukkan ke table

- Kembali ke SQLYog, perhatikan bahwa table **dim_produk** telah memiliki data.

sk_produk	kode_kategori	kode_produk	kode_item
1	KTG-003	PROD-0000001	ITM
2	KTG-003	PROD-0000002	ITM
3	KTG-003	PROD-0000003	ITM
4	KTG-003	PROD-0000004	ITM
5	KTG-003	PROD-0000005	ITM
6	KTG-003	PROD-0000006	ITM

- Masih pada SQLYog, ketikkan perintah berikut pada tab “Query” untuk menghitung jumlah row yang saat ini ada pada table **dim_produk**.

```
1 SELECT COUNT(*) FROM dim_produk;
```

- Tekan tombol **F9** untuk mengeksekusi perintah tersebut, dan pada tab **Result** akan kelihatan hasil nilai **41**.

count(*)
41

- Kembali ke Spoon, kembali kita coba jalankan *transformation dim_produk.ktr* dan tunggu sampai proses tersebut selesai.
- Sekarang, hitung jumlah rows yang ada pada table **dim_produk** saat ini.



- Terlihat jumlah row saat ini adalah 82, yang merupakan hasil yang salah atau tidak kita harapkan karena akan terjadi duplikasi record seperti terlihat di bawah ini.

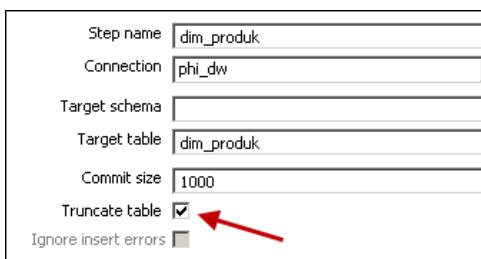
The screenshot shows the Spoon interface with a query editor at the top containing the following SQL code:

```
1 SELECT sk_produk, nama_kategori, nama_produk
2 FROM dim_produk ORDER BY sk_produk;
```

Below the query editor is a result grid titled '1 Result'. The grid has three columns: 'sk_produk', 'nama_kategori', and 'nama_produk'. It displays four rows of data, with the first two rows being identical:

sk_produk	nama_kategori	nama_produk
1	Sayur-Sayuran	bawang merah 1kg
1	Sayur-Sayuran	bawang merah 1kg
2	Sayur-Sayuran	bawang putih 1 kg
2	Sayur-Sayuran	bawang putih 1 kg

- Untuk memperbaiki hal ini, kita harus menghapus isi table terlebih dahulu sebelum dilakukan populasi. Kembali ke Spoon dan klik dua kali pada step **dim_produk**.
- Opsi **Truncate table** kita aktifkan.



- Simpan dan jalankan kembali *transformation* tersebut.
- Periksa kembali jumlah row yang ada pada table **dim_produk**, apakah kali ini jumlahnya tetap bertambah atau kembali ke nilai 41? Apa yang dapat Anda simpulkan?
- Struktur terakhir **dim_produk** yang kita dapatkan tampak seperti pada gambar berikut.

Column Name	Data Type	Length
sk_produk	int	11
kode_kategori	varchar	7
kode_produk	varchar	12
kode_item	varchar	7
nama_kategori	varchar	17
nama_produk	varchar	100
unit	int	11
kode_satuan	varchar	4

- Selesai.

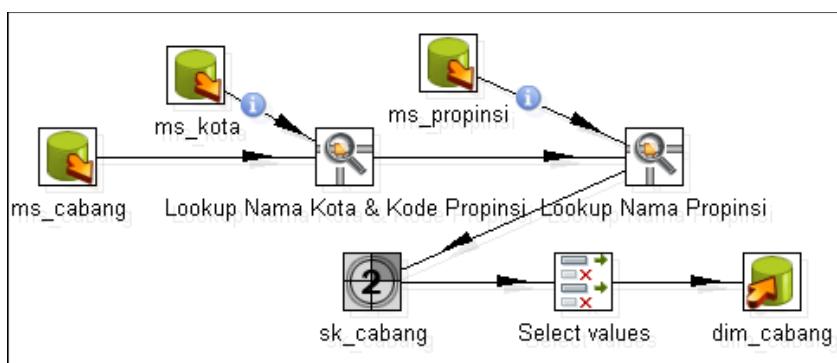
10.5 Praktek: Dimensi Cabang

Dengan pengetahuan yang Anda dapatkan pada praktek sebelumnya, cobalah buat suatu *transformation* dengan nama file **c:\kettletraining\dim_cabang.ktr** yang akan melakukan hal berikut:

- membaca table **ms_propinsi**, **ms_kota** dan **ms_cabang** dari database **phi_minimart**
- menghasilkan table **dim_cabang** dengan struktur akhir seperti pada gambar berikut.

Column Name	Data Type	Length
sk_cabang	int	11
kode_propinsi	varchar	3
nama_propinsi	varchar	25
kode_kota	varchar	8
nama_kota	varchar	16
kode_cabang	varchar	10
nama_cabang	varchar	100

Dan untuk membantu Anda, berikut adalah gambar screenshot salah satu rancangan flow *transformation* untuk menghasilkan table **dim_cabang**.



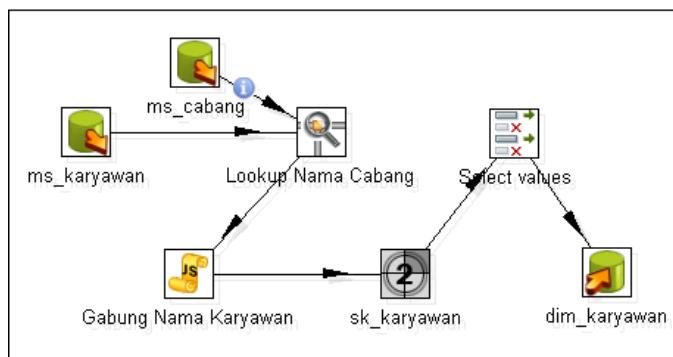
10.6 Praktek: Dimensi Karyawan

Dengan pengetahuan yang Anda dapatkan pada praktek sebelumnya, cobalah buat suatu *transformation* dengan nama file **c:\kettletraining\dim_karyawan.ktr** yang akan melakukan hal berikut:

- membaca table **ms_karyawan** dan **ms_cabang** dari database **phi_minimart**.
- menghasilkan table **dim_karyawan** dengan struktur akhir seperti pada gambar berikut.

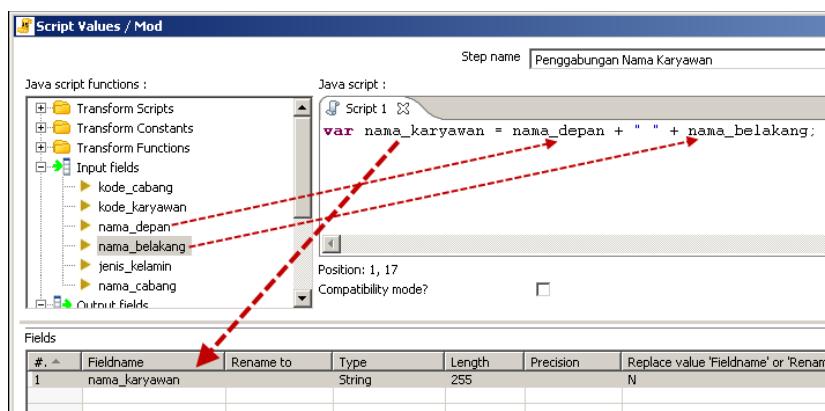
Column Name	Data Type	Length
sk_karyawan	int	11
kode_cabang	varchar	10
nama_cabang	varchar	100
kode_karyawan	varchar	10
nama_karyawan	varchar	255
jenis_kelamin	char	1

Dan untuk membantu Anda, berikut adalah gambar screenshot salah satu rancangan flow transformation untuk menghasilkan table **dim_karyawan**.



Perhatikan pada *transformation* ini terdapat satu *step* baru, yaitu **Gabung Nama Karyawan** yang merupakan tipe *step* **Modified Java Script Value**. Step ini digunakan untuk menggabungkan nama depan dan belakang dari dari *field-field* *table* **ms_karyawan**.

Cara penggabungan dari dengan *step* ini terlihat pada gambar di bawah ini. Pada editor Javascript terlihat kita mendefinisikan suatu variable **nama_karyawan** yang kemudian didefinisikan sebagai output dari *step* ini.

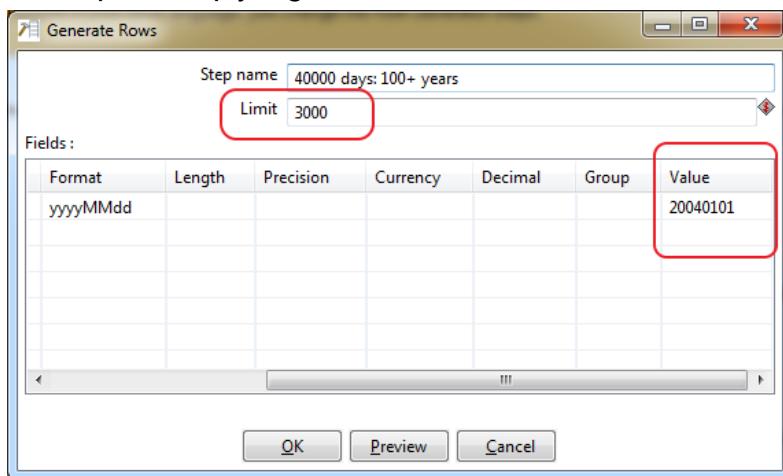


10.7 Praktek: Dimensi Waktu

- Pada paket Kettle telah disertakan satu contoh transformasi yang dapat digunakan untuk menghasilkan tabel dimensi waktu dengan struktur akhir seperti pada gambar berikut.

Column Name	Data Type	Length
sk_waktu	int	11
date_field	datetime	
year	int	11
month	int	11
day_of_year	int	11
day_of_month	int	11
day_of_week	int	11
week_of_year	int	11
day_of_week_desc	varchar	30
day_of_week_short	varchar	3
month_desc	varchar	30
month_short_desc	varchar	3
quarter	char	1
WeekOfMonth	bigint	20

- Cobalah buka file dengan nama **General - Populate date dimension.ktr** pada lokasi [Kettle]/samples/transformations.
- Klik dua kali pada step **40000 days: 100+ years**, pada kotak **Limit** ganti nilai 40000 menjadi 3000.
- Masih pada step yang sama masukkan nilai kolom **Value** menjadi 20040101.



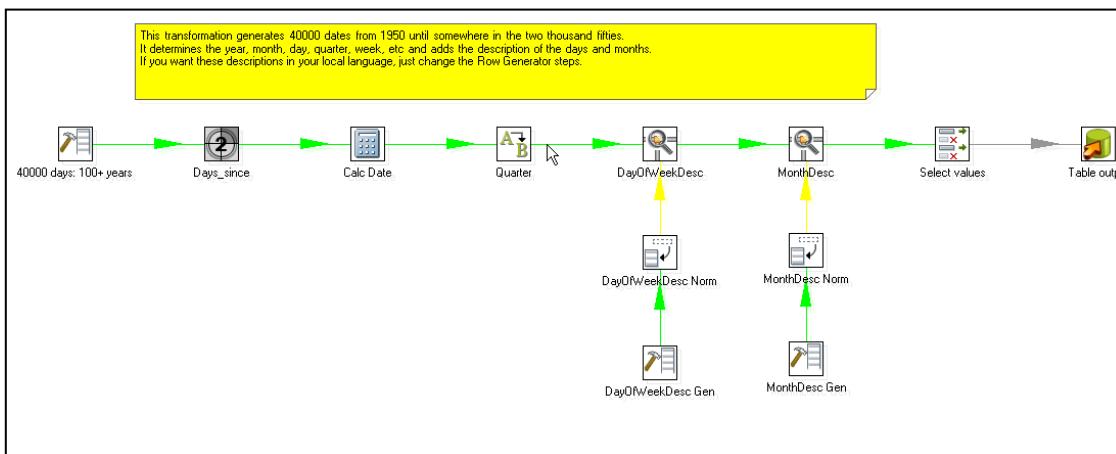
- Klik dua kali pada step **Select Values**, pada kolom **Fieldname** cari baris yang memiliki nilai **date_tk**. Setelah Anda temukan, masih pada baris yang sama masukkan **sk_waktu** pada kolom **Rename to**.

Fields :		
#	Fieldname	Rename to
1	date_tk	sk_waktu
2	Date	date_field

- Perhatikan bahwa *hop* antara step **Select Values** dan **Table Output** dalam keadaan *disabled* atau tidak aktif. Aktifkan *hop* tersebut dengan cara klik sekali, atau cara lainnya adalah dengan

klik kanan pada hop tersebut dan pilih **Enable hop**.

- Pada **Table Output** arahkan target table ke dim_waktu, aktifkan opsi truncate table. Bentuklah table tersebut dengan klik tombol **SQL** dan eksekusi perintah **Create Table...** yang direkomendasikan.
- Jalankan *transformation* ini dan lihatlah hasil output **dim_waktu** pada **SQLYog**.
- Mintalah penjelasan dari Instruktur akan logika transformasi ini serta semua *step* yang digunakan disertai contoh.
- Objektif utama dari *transformation* ini adalah menghasilkan table dengan nama **dim_waktu**.

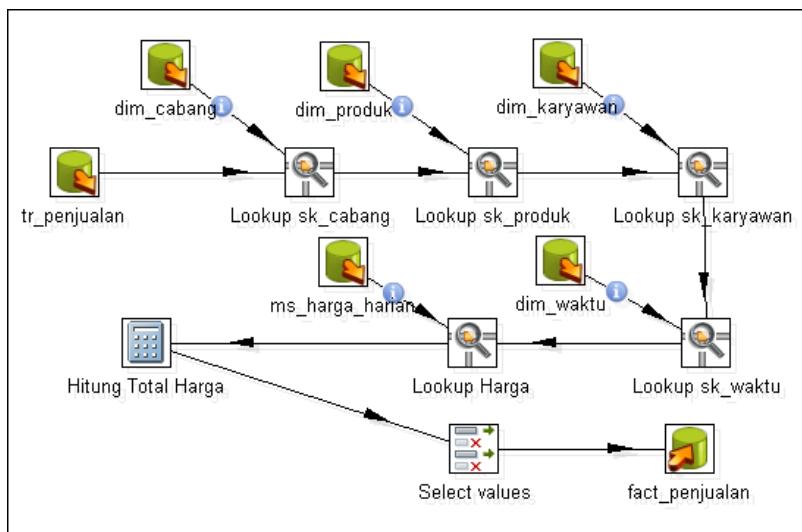


10.8 Praktek: Fact Penjualan PHI-Minimart

- Buat *transformation* baru dan simpan dengan nama **c:\kettletraining\fact_penjualan.ktr**.
- Selesaikan *transformation* ini dengan bantuan instruktur sehingga Anda mendapatkan table **fact_penjualan** pada **phi_dw** dengan struktur tampak seperti pada gambar berikut.

	Column Name	Data Type	Length
1	sk_produk	int	11
2	sk_cabang	int	11
3	sk_karyawan	int	11
4	sk_waktu	int	11
5	jumlah_pembelian	int	11
6	total_harga	int	11
7			

Untuk membantu Anda, berikut adalah gambar screenshot salah satu rancangan flow *transformation* untuk menghasilkan table **fact_penjualan**.



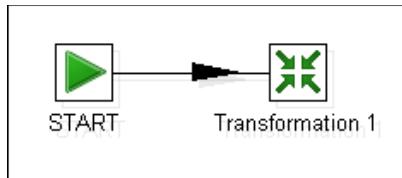
- Terlihat ada step baru, yaitu **Hitung Total Harga**. Step ini adalah **step Calculator**. Isi konfigurasi dari calculator tersebut seperti tampak pada gambar berikut ini. Mintalah penjelasan lebih lanjut dari instruktur mengenai penggunaan step ini.

Step name: Hitung Total Harga						
Fields:						
#	New field	Calculation	Field A	Field B	Fi...	Value type
1	total_harga	A * B	jumlah_pembelian	harga_berlaku_cabang		Integer

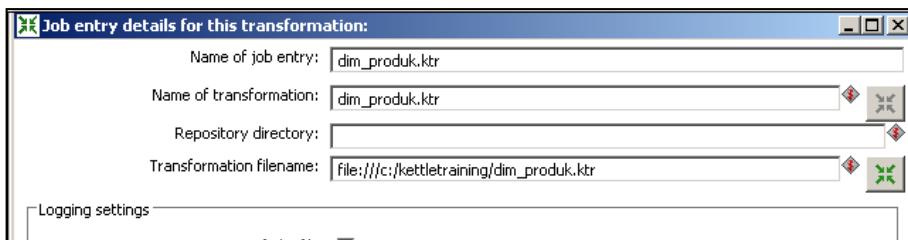
- Selesai.

10.9 Praktek: Menggabungkan Semua File Transformation dengan Job

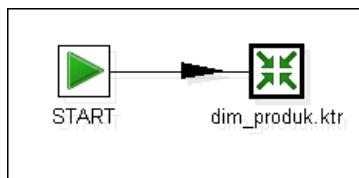
- Buat suatu job dan namakan dengan c:\kettletraining\dwh_job.kjb.
- Masukkan step **Start** pada job tersebut.
- Masukkan step **Transformation** dan hubungkan dengan step **Start**.



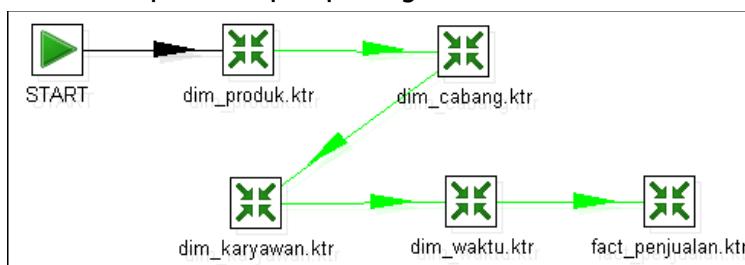
- Klik dua kali pada step **Transformation** dan pada bagian **Transformation filename** klik tombol untuk mengambil file *transformation*, masukkan file dim_produk.ktr yang telah kita buat sebelumnya dan namakan step ini dengan nama **dim_produk.ktr**.



- Klik tombol **OK** untuk menutup dialog.



- Tambahkan *transformation-transformation* lain dengan **fact_penjualan.ktr** sebagai step yang terakhir seperti tampak pada gambar berikut.



- Jalankan job ini dan perhatikan log yang ada pada panel bawah.

Job / Job Entry	Comment	Result
dwh_job		
Job: dwh_job	Start of job execution	
START	Start of job execution	
START	Job execution finished	Success
dim_produk.ktr	Start of job execution	
dim_produk.ktr	Job execution finished	Success
dim_cabang.ktr	Start of job execution	
dim_cabang.ktr	Job execution finished	Success
dim_karyawan.ktr	Start of job execution	
dim_karyawan.ktr	Job execution finished	Success
dim_waktu.ktr	Start of job execution	
dim_waktu.ktr	Job execution finished	Success
fact_penjualan.ktr	Start of job execution	

- Pastikan job dengan semua transformation tersebut telah selesai dan sukses dieksekusi, periksalah semua table yang terdapat pada database **phi_dw**.
- Selesai.

10.10 Praktek: Penggunaan Mondrian sebagai OLAP Engine PHI-Minimart

Mondrian adalah OLAP Server yang juga merupakan bagian dari Pentaho dan dapat mengenali Star Schema dengan baik.

- Ikuti instalasi Apache Tomcat dan Mondrian war (*web archive*) sesuai dengan petunjuk dari instruktur.
- Instruktur akan menunjukkan penggunaan Pentaho Workbench Editor untuk pemetaan antara schema Mondrian dengan struktur database kita.
- Instruktur akan melakukan demo pembuatan file JSP dengan suatu query MDX (Multi Dimensional Expression) sehingga kita dapat mudah melakukan navigasi OLAP ke data PHI-Minimart kita.
- Peserta meng-copy semua file schema dan JSP dari instruktur untuk dicoba pada instalasi lokal tiap-tiap peserta pelatihan.

PHI-Minimart - Cube Penjualan					
	Measures				
Karyawan	Total Penjualan	Total Biaya	Total Modal	Untung Bersih	
-Semua Karyawan	Rp 65,109,181,120	Rp 797,439,750	Rp 61,851,799,240	Rp 2,459,942,130.00	
-Pria	Rp 32,540,288,470	Rp 399,464,270	Rp 30,912,263,960	Rp 1,228,560,240.00	
Eko Rukun	Rp 2,210,246,280	Rp 26,776,750	Rp 2,099,578,810	Rp 83,890,720.00	
Ferdy Tenteram	Rp 2,211,853,150	Rp 27,481,460	Rp 2,101,210,360	Rp 83,161,330.00	
Sentosa Indrawan	Rp 2,186,009,070	Rp 26,498,030	Rp 2,076,709,820	Rp 82,801,220.00	
Mulia Setiawan	Rp 2,189,184,190	Rp 26,997,940	Rp 2,079,657,450	Rp 82,528,800.00	
Eriq Jagat	Rp 2,178,908,070	Rp 26,417,670	Rp 2,070,137,460	Rp 82,352,940.00	
Galang Setiawan	Rp 2,198,842,350	Rp 27,677,640	Rp 2,088,986,300	Rp 82,178,410.00	
Agus Dewwangga	Rp 2,182,763,800	Rp 27,060,950	Rp 2,073,601,250	Rp 82,101,600.00	
Mulyo Damai	Rp 2,176,178,610	Rp 26,971,420	Rp 2,067,279,170	Rp 81,928,020.00	
Galang Terang	Rp 2,166,765,130	Rp 26,661,510	Rp 2,058,347,970	Rp 81,755,650.00	
Eriq Menawen	Rp 2,151,267,410	Rp 26,235,580	Rp 2,043,312,040	Rp 81,719,790.00	
Aris Siberut	Rp 2,169,262,660	Rp 27,022,560	Rp 2,060,646,340	Rp 81,593,760.00	
Agung Alexander	Rp 2,134,846,460	Rp 25,768,050	Rp 2,027,823,050	Rp 81,255,360.00	
Budi Tenteram	Rp 2,154,287,100	Rp 26,640,160	Rp 2,046,489,450	Rp 81,157,490.00	
Erman Margo	Rp 2,121,434,110	Rp 25,795,590	Rp 2,015,521,670	Rp 80,116,850.00	
Kusuma Dominik	Rp 2,108,440,080	Rp 25,458,960	Rp 2,002,962,820	Rp 80,018,300.00	
Aron Indrawan					
+Wanita	Rp 28,138,596,010	Rp 343,108,510	Rp 26,731,257,340	Rp 1,064,230,160.00	

Bagian 3

Staging Database



10.11 Pendahuluan

Dalam proses ETL terdapat beberapa tugas yang memakan sumber CPU dan memori / RAM yang cukup besar seperti sorting, join data (terutama yang sumbernya tidak sama), data lookup, dan sebagainya.

Ini mengakibatkan ETL harus membebani dirinya dan menurunkan performa dengan tugas yang sebenarnya bisa dijalankan dengan baik oleh produk RDBMS. Untuk menghindarkan hal tersebut maka sering sekali kita membuat suatu database staging sebagai *buffer* untuk mengintegrasikan data terlebih dahulu sebelum diproses lebih lanjut.

10.12 Latar Belakang : Physical I/O versus In-Memory Processing

Hal lain yang menjadikan data staging sebagai solusi yang sangat baik adalah karena proses di memori (RAM) tentunya sangat terbatas dan ujung-ujungnya akan mencari *space* di hard drive kita untuk paging / caching.

Dengan batasan seperti itu, akhirnya lama kelamaan proses di memori akan penuh terus dan malah menjadi bottleneck di ETL kita. Oleh sebab itu apabila terjadi kondisi yang akan sangat membebani memori kita menggunakan strategi data staging.

10.13 Change Data Capture (CDC)

Salah satu permasalahan pada proyek data warehouse adalah kemampuan untuk hanya mengambil data sumber yang berubah.

Walaupun kelihatan sederhana tapi permasalahan ini sebenarnya sangat sulit dan klasik dikarenakan beberapa faktor berikut :

- Data sumber belum tentu sepenuhnya dalam kontrol kita sehingga untuk memasukkan logika seperti trigger tidak memungkinkan
- Walaupun trigger memungkinkan tapi masih terbatas pada policy dari database OLTP dan penambahan trigger akan sulit kalau terlalu banyak source tablenya
- Pembacaan transaction log file yang sangat kompleks dan dokumentasinya sendiri sangat susah dicari.

10.14 Demo: Change Data Capture pada SQL Server 2008

SQL Server 2008 adalah salah satu RDBMS yang telah mendukung penggunaan CDC secara native. Trainer akan mendemokan penggunaan CDC ini dimana operasi insert, update maupun delete dapat di-capture dengan baik.

10.15 Praktek: Baca file dari 3 sumber

- Baca file teks (csv) master produk, dokumen Excel tentang target, and database MySQL yang

diberikan oleh instruktur.

- Instruktur akan membimbing pelatihan untuk membaca dan menggabungkan 3 file tersebut dengan stream lookup.
- Perhatikan performa dari ETL tersebut.

10.16 Praktek: Menggunakan Staging Database

- Lakukan hal yang sama dengan point di atas, hanya kali ini dilakukan melalui staging database.

Bab 11. Advance Controls

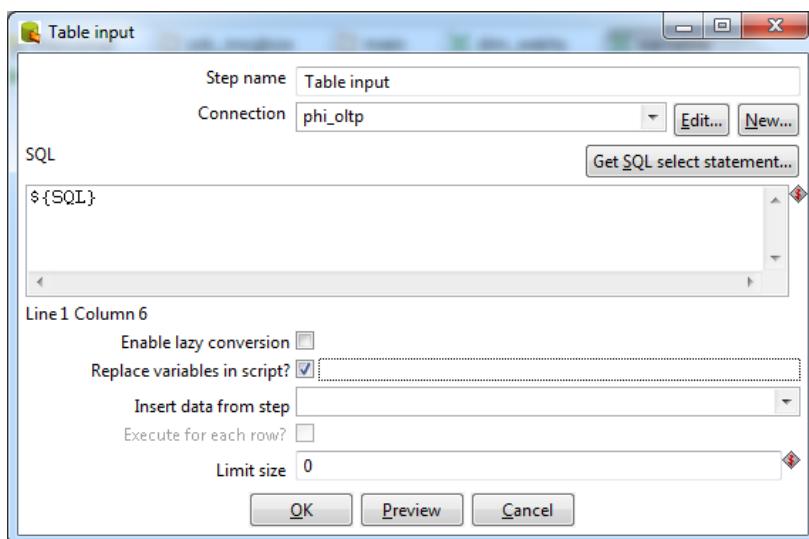
11.1 Environment Variable

Seperti telah dijelaskan pada bagian sebelumnya, variabel environment adalah jenis variable khusus yang dikenali di keseluruhan environment Kettle. Akses dapat dilakukan melalui step Javascript, Field yang memungkinkan penggunaan variable (tanda \$), dan lainnya.

Dapat didefinisikan di kettle.properties dan beberapa step script dan step yang berkaitan dengan job.

11.1.1 Praktek: Menggunakan Variable pada Table Input

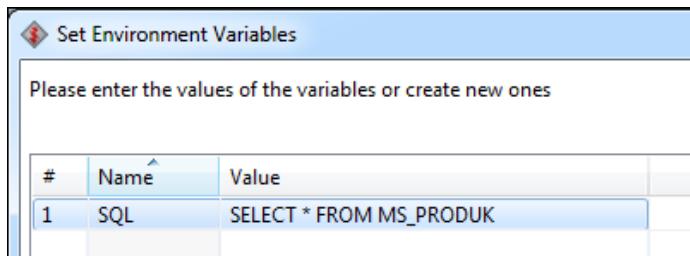
- Buatlah satu transformation baru dengan nama **c:\kettletraining\variable.ktr**.
- Masukkan step **Table Input** ke dalam ruang kerja kita.
- Klik dua kali pada step tersebut dan masukkan konfigurasi seperti pada gambar berikut.



- Klik tombol **OK** untuk menutup dialog ini.
- Jika Anda melakukan *preview* atau menjalankan *transformation* ini maka akan menemukan *error* karena variabel **SQL** belum kita definisikan dan berikan nilainya.



- Pada *pulldown menu*, pilih **Edit | Set Environment Variables....**
- Pada dialog Set Environment Variables masukkan konfigurasi seperti pada gambar berikut.



- Sekarang cobalah lakukan preview data lagi pada step Table Input, kali ini Anda akan mendapatkan hasil seperti pada gambar berikut.

#	kode_produk	kode_kategori	nama_produk
1	PROD-0000001	KTG-003	red onion 1kg
2	PROD-0000002	KTG-003	bawang putih
3	PROD-0000003	KTG-003	buncis 1 kg
4	PROD-0000004	KTG-003	kacang panjang
5	PROD-0000005	KTG-003	kacang hijau
6	PROD-0000006	KTG-003	kentang 1 kg
7	PROD-0000007	KTG-003	ketimun 1 kg
8	PROD-0000008	KTG-003	kobis 1 kg

- Selesai.

11.2 Shared Object

Di dalam folder konfigurasi Kettle (.kettle) terdapat satu file shared.xml yang berisi semua resource shared dari Kettle terutama koneksi database.

Dengan adanya file ini maka kita akan sangat terbantu apabila kita ingin berpindah *environment*, misalnya antara lingkungan pengembangan / *development* dengan *production*.

Shared Connections

```
<?xml version="1.0" encoding="UTF-8"?>
<sharedobjects>
<connection>
<name>PHI_DW</name>
<server>localhost</server>
<type>MYSQL</type>
<access>Native</access>
<database>dipa_dw</database>
<port>3306</port>
<username>root</username>
<password>Encrypted </password>
<servername/>
<data_tablespace/>
<index_tablespace/>
```

```
<attributes>
<attribute><code>EXTRA_OPTION_MYSQL.defaultFetchSize</code><attribute>500</attribute>
<attribute><code>..... </code><attribute>true</attribute></attribute>
.....
<code>USE_POOLING</code><attribute>N</attribute></attribute>
</attributes>
</connection>
.....
</connection>

</sharedobjects>
```

11.3 Error Handling

Error handling adalah suatu mekanisme penanganan kesalahan / error yang terjadi ketika menjalankan transformasi data seperti kesalahan konversi tipe data, kesalahan logika pemrograman, penanganan nilai null, dan lain sebagainya.

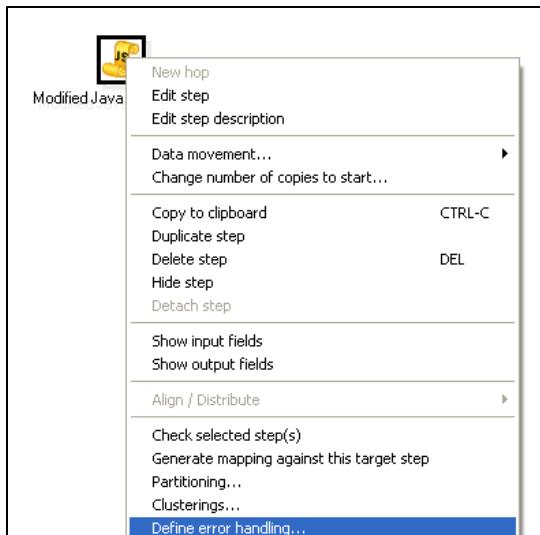
Dengan penanganan error ini, kita tidak membiarkan transformasi berhenti tiba-tiba secara tidak terkontrol, sebaliknya kita akan dapat mengarahkan dan mencatat aliran data yang tidak terproses dengan baik ini untuk perbaikan transformasi itu sendiri maupun data yang kita olah.

Beberapa kontrol yang dapat kita lakukan antara lain seperti mengulang eksekusi *transformation*, mengirimkan notifikasi kepada admin database, dan lain sebagainya.

Tidak semua *step transformation* dari Kettle memiliki fitur error handling, beberapa *step* yang sudah mendukung penanganan error atau error handling antara lain adalah **Email Step, Modified Javascript Value, Database Lookup, Table Exists, HTTP Post**, dan lain-lain.

11.4 Praktek: Error Handling pada Modified Javascript Value

- Contoh *step* yang memiliki penanganan error ini adalah “**Modified Javascript Value**”.



- Beberapa konfigurasi yang perlu dimasukkan pada dialog error handling ini adalah sebagai berikut :
 - Error handling *stepname* : Nama *step* dimana penanganan error ini berada.
 - Target Step : Step yang dituju jika terjadi error.
 - Enable the error handling : Mengaktifkan / menonaktifkan penanganan error.
 - Nr of errors fieldname : Nama *field* yang akan berisi nomor error.
 - Error descriptions fieldname : Nama *field* yang akan berisi deskripsi error.
 - Error fields fieldname : Nama *field* yang akan berisi *field* yang mengakibatkan error.

- Error codes fieldname : Nama *field* yang akan berisi kode error.
- Max nr errors allowed : Jumlah error yang diperbolehkan terjadi.
- Max % errors allowed : Persentase jumlah error yang diperbolehkan terjadi.
- Min nr rows to read before doing %: Jumlah error versus persentase, jika jumlah error sudah melebihi nilai yang dimasukkan walaupun belum mencapai nilai persentase maka flow ETL akan dihentikan.
- Instruktur akan membuat kasus dan menunjukkan penggunaan dari kontrol penanganan error ini.

11.5 Praktek: Setting Variable pada file kettle.properties

- Buat database MySQL baru bernama **phi_prod**.
- Copy semua folder **.kettle** yang berlokasi di folder C:\Documents and Settings\User ke c:\kettletraining
- Buat user baru dari control panel Anda.
- Login dengan user baru tersebut.
- Copy folder **.kettle** dari c:\kettletraining ke folder document dari user tersebut. Rubah setting semua output database dari **phi_minimart** ke **phi_prod** tanpa merubah job/transformasi kita.
- Jalankan job kita dengan login user baru tersebut.

11.6 Praktek: Notifikasi Error via Email

- Buat suatu file job / transformasi yang membaca file **c:\kettletraining\contoh-data\pola_angka.txt**
- Tambahkan step **Modified Java Script** dengan kode yang diberikan instruktur.
- Tambahkan step **Mail** dan masukkan konfigurasi SMTP yang valid di step tersebut.
- Kirimkan email apabila terjadi error pada step **Modified Java Script** tersebut dengan bimbingan instruktur.

11.7 Diskusi Kasus & Praktek: Changed Data Capture (CDC)

- Instruktur dan peserta akan berdiskusi mengenai contoh CDC dari PHI-Minimart dan beberapa strategi dan kompromi yang dapat digunakan dan merancang kumpulan job/ transformation yang mampu menangani CDC tersebut dengan didukung berbagai penggunaan *step*, kontrol dari Kettle dan *staging database*.

Bab 12. Slowly Changing Dimension

12.1 Apa itu Slowly Changing Dimension ?

Dapat dipastikan bahwa pada suatu saat sumber data pembentuk dimensi akan berubah. Perubahan tersebut dinamakan **Slowly Changing Dimension (SCD)** atau dimensi yang berubah secara perlahan.

Dari perubahan yang terjadi tersebut kita dapat menentukan apa respon kita. Ada 3 macam tipe penanganan SCD yaitu Tipe 1, 2, dan 3. Bagian berikut menerangkan apa perbedaan dari tiap tipe SCD dan contoh gambar yang diambil dari situs Wikipedia.

12.2 Type 1 SCD

- Menggunakan pendekatan timpa data yang berubah (*overwrite*).
- Ini dilakukan jika memang tidak ada kepentingan menyimpan data historis atau pergerakan isi data dari dimensi tersebut.

Supplier_key	Supplier_Name	Supplier_State
001	Phlogistical Supply Company	CA

Supplier_key	Supplier_Name	Supplier_State
001	Phlogistical Supply Company	IL

12.3 Type 2 SCD

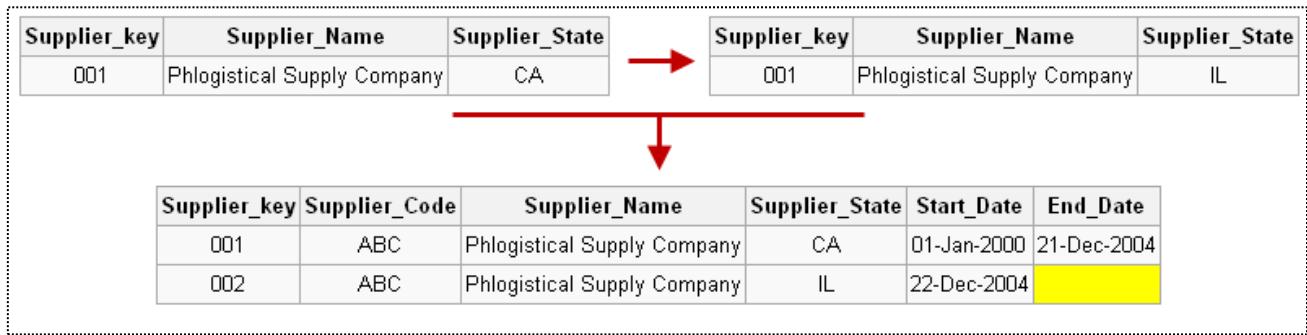
- Menyimpan semua historis data dengan surrogate key berbeda (*partitioning history*).
- Dapat menambahkan masa berlaku dari dimensi bersangkutan.
- Ini dilakukan jika memang pergerakan data historis tetap berpengaruh kepada analisis data seperti perpindahan salesman dari satu regional ke regional lain.

Supplier_key	Supplier_Name	Supplier_State
001	Phlogistical Supply Company	CA

Supplier_key	Supplier_Name	Supplier_State
001	Phlogistical Supply Company	IL

Supplier_key	Supplier_Code	Supplier_Name	Supplier_State	version
001	ABC	Phlogistical Supply Company	CA	0
002	ABC	Phlogistical Supply Company	IL	1

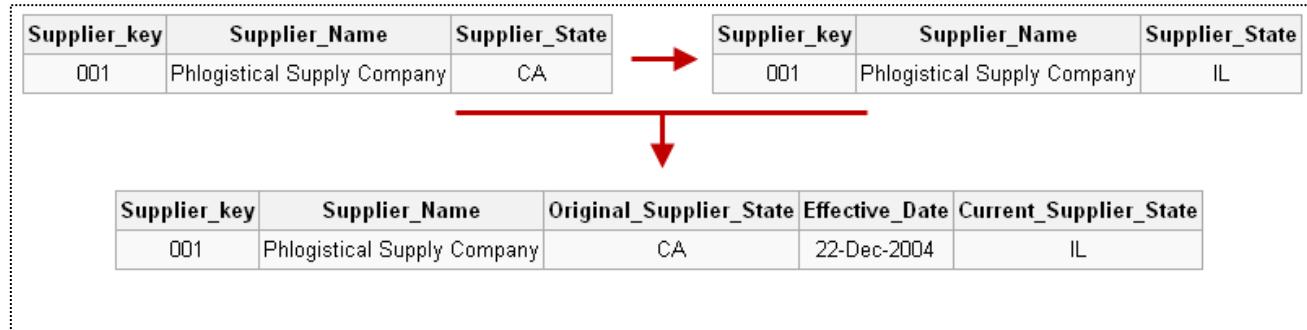
atau



12.4 Type 3 SCD

- Menyimpan satu historis data pada kolom lain (*alternate realities*).
- Sama seperti Type 2 hanya saja lebih terbatas kepada jumlah perubahan data (*update*) yang diijinkan.

Contoh pada gambar berikut hanya dapat melakukan pelacakan (*tracking*) satu perubahan saja.



12.5 SCD Step : Dimension Lookup / Update

Untuk mengakomodasi SCD yang jika kita kerjakan secara manual akan sangat memakan banyak waktu maka Kettle sudah mempersiapkan khusus untuk SCD yaitu Dimension Lookup / Update.

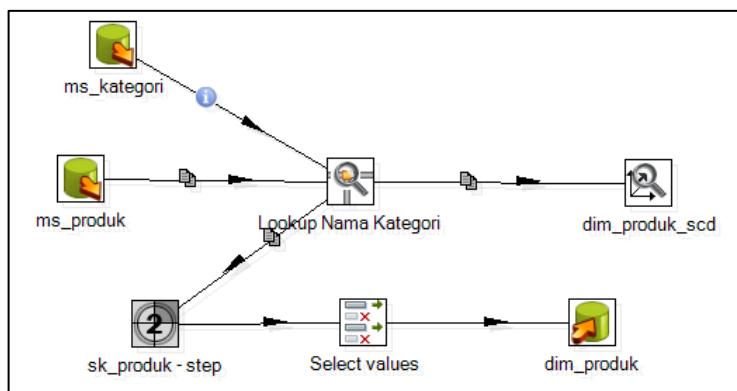
Step ini bisa mengerjakan SCD type 1 dan 3 tergantung konfigurasi yang kita berikan kepada *step* tersebut. Untuk SCD type 3 juga hybrid, yaitu bisa menggunakan versioning dan masa berlaku (*start date* dan *end date*) dari dimensi tersebut sekaligus.

Untuk tanggal berlaku juga bisa mengambil sumber dari :

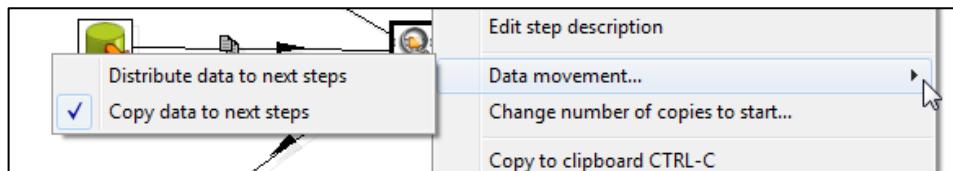
- tanggal dan jam eksekusi dari sistem operasi kita.
- tanggal dan jam dari *field* tertentu dari data sumber kita.

12.6 Praktek: SCD untuk Dimensi Produk

- Buka kembali file transformasi yang digunakan untuk membentuk dimensi produk pada praktek 10.4, yaitu file bernama **c:\kettletraining\dim_produk.ktr**.
- Tambahkan **step Data Warehouse | Dimension lookup/update**. Rubah nama **step** tersebut menjadi **dim_produk_scd**.
- Aturlah semua **step** seperti tampak pada gambar berikut.



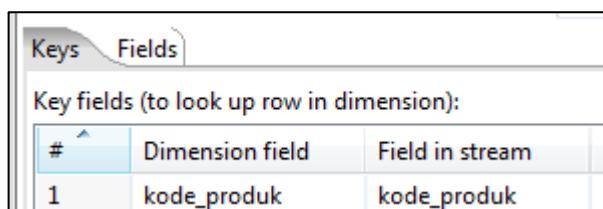
- Pastikan data movement pada **step Lookup Nama Kategori** adalah **Copy Data to next steps**.



- Klik dua kali pada **step dim_produk_scd** dan isikan konfigurasi berikut:

 - Connection : phi_dw
 - Target table : dim_produk_scd
 - Technical key field : sk_produk
 - Version field : versi
 - Date range start field: tgl_awal
 - Date range end : tgl_akhir

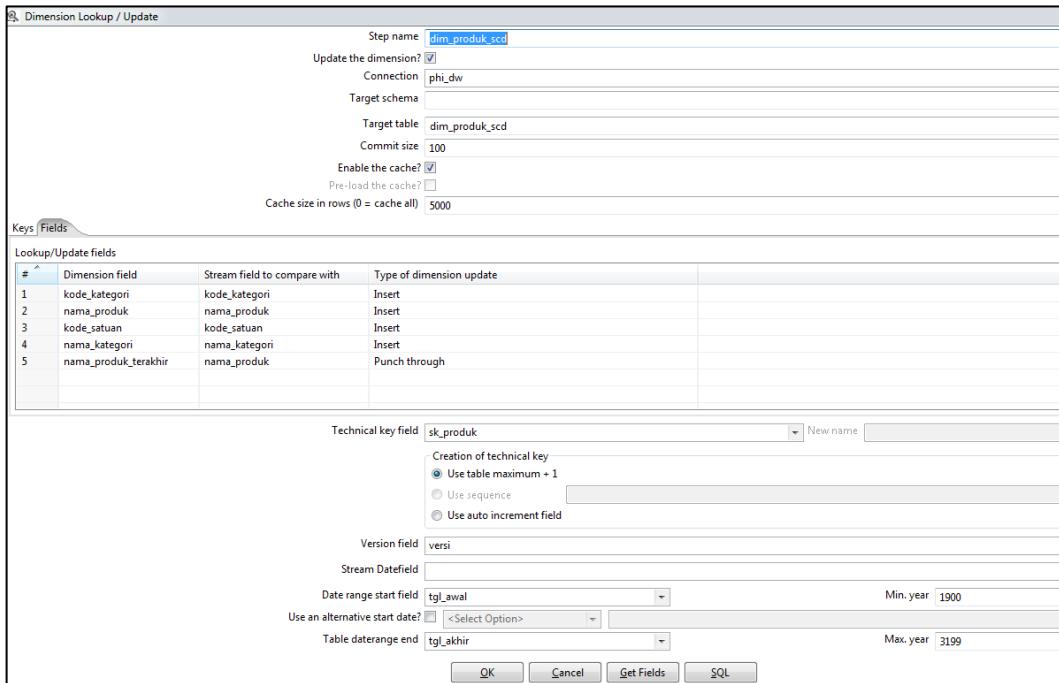
- Klik tab **Keys** pada bagian daftar **field** tombol, klik tombol **Get Fields**. Hapus semua **field** sehingga hanya **field kode_produk** yang tersisa.



- Klik tab **Fields** pada bagian daftar kolom, klik tombol **Get Fields**. Sesuaikan isinya dengan gambar berikut ini. Untuk entri **nama_produk_terakhir** Anda harus memasukkannya secara manual.

#	Dimension field	Stream field to compare with	Type of dimension update
1	kode_kategori	kode_kategori	Insert
2	nama_produk	nama_produk	Insert
3	kode_satuan	kode_satuan	Insert
4	nama_kategori	nama_kategori	Insert
5	nama_produk_terakhir	nama_produk	Punch through

- Hasil akhir dari *step* tersebut terlihat seperti gambar berikut.



- Klik tombol **SQL** dan pilih **Execute**.
- Jalankan transformasi tersebut dan lihat hasilnya pada SQLYog.

sk_produk	versi	tgl_awal	tgl_akhir	kode_produk	kode_kategori	nama_produk	kode_satuan
0	1	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000001	KTG-003	bawang merah 1kg	kg
2	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000002	KTG-003	bawang putih 1 kg	kg
3	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000003	KTG-003	buncis 1 kg	kg
4	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000004	KTG-003	kacang panjang 1 kg	kg
5	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000005	KTG-003	kacang hijau 1 kg	kg
6	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000006	KTG-003	kentang 1 kg	kg
7	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000007	KTG-003	ketimun 1 kg	kg
8	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000008	KTG-003	kobis 1 kg	kg
9	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000009	KTG-003	tomat 1 kg	kg
10	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000010	KTG-003	ubi kayu basah 1 kg	kg
11	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000011	KTG-003	ubi jalar 1 kg	kg
12	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000012	KTG-003	wortel 1 kg	kg
13	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000013	KTG-002	susu kental (800 ml)	tk
14	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000014	KTG-002	telur ayam ras 1 kg	kg
15	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000015	KTG-002	telur ayam kampung 1 kg	kg
16	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000016	KTG-002	mie instant 1 kotak (40 bungkus)	bks
17	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000017	KTG-002	sari minuman 500 ml	btl

- Lakukan perubahan nilai *field* **nama_produk** tabel **ms_produk** dengan membuka dan

mengeksekusi file **rubah_scd.sql** – terdapat pada DVD training Anda – dengan SQLYog.

- Jalankan kembali transformasi dan lihat kembali hasilnya pada table **dim_produk_scd**.

sk_produk	versi	tgl_awal	tgl_akhir	kode_produk	kode_kategori	nama_produk
0	1	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
42	2	2014-02-03 00:02:25	3199-12-31 23:59:59	PROD-0000001	KTG-003	red onion 1kg
1	1	1900-01-01 00:00:00	2014-02-03 00:02:25	PROD-0000001	KTG-003	bawang merah 1kg
2	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000002	KTG-003	bawang putih 1 kg
3	1	1900-01-01 00:00:00	3199-12-31 23:59:59	PROD-0000003	KTG-003	buncis 1 kg

- Apa yang dapat Anda simpulkan?

Bab 13. Otomatisasi

Setelah semua proses pengembangan script ETL untuk *data warehouse* selesai, saatnya melakukan otomatisasi dengan menjadwalkan proses tersebut untuk dijalankan secara *batch* per periode secara teratur. Otomatisasi ini biasanya dilakukan secara harian.

Selain itu kita perlu memperhatikan strategi waktu yang diambil, yaitu sebaiknya pada saat database OLTP tidak dalam keadaan sibuk. Sebagai contoh kita dapat mengambil waktu jam 12 malam malam sebagai periode waktu terbaik.

13.1 Otomatisasi Kettle

Untuk melakukan otomatisasi Kettle kita maka perlu dibuat batch / shell script yang akan mengatur eksekusi dari job / *transformation* kita.

- Eksekusi *Transformation* dengan **Pan**

Syntax:

```
pan.bat /option:value –file=transformation_file.ktr
```

- Eksekusi Job dengan **Kitchen**

Syntax:

```
kitchen.bat /option:value –file=transformation_file.ktr
```

- Beberapa hal yang perlu diperhatikan:

- Eksekusi **kitchen** dan **pan** harus dilakukan dari folder instalasi Kettle. Hal ini disebabkan pada kedua script tersebut mengandung pemanggilan library yang relatif terhadap folder instalasi.
- Harus memanggil *path* lengkap dari file job / *transformation*.

Opsi-opsi selengkapnya untuk kedua perintah ini dapat Anda lihat pada dokumentasi online Kettle pada alamat berikut:

- Kitchen: <http://wiki.pentaho.com/display/EAI/Kitchen+User+Documentation>
- Pan: <http://wiki.pentaho.com/display/EAI/Pan+User+Documentation>

13.2 Praktek: Script dan Penjadwalan Eksekusi

- Buat file **DataWarehouse.bat** yang di dalamnya mengandung perintah eksekusi job **main.kjb** dengan isi sebagai berikut.

```
c:  
cd c:\kettle-4.2  
kitchen.bat -file=c:\kettletraining\main.kjb -level=Detailed  
cd c:\kettletraining
```

- Jalankan file DataWarehouse.bat dan lihat proses eksekusinya pada *console windows* yang muncul.
- Gunakan Windows Task Scheduler untuk mengeksekusi file **DataWarehouse.bat** untuk setiap 10 menit.



Let us guide you to be a
MODERN DATA SCIENTIST

Register at: dqlab.id