Pengantar Pentaho Data Integration (Kettle)

Modul Tutorial Praktikum

Yudi Wibisono yudi@upi.edu / t: @yudiwbs
Ilmu Komputer UPI (cs.upi.edu)
Versi 0.5 (BETA) Oktober 2014

Pembaca modul ini diasumsikan telah menguasai konsep dasar basisdata, termasuk SQL

Pengantar

Pentaho Data Integration (PDI) atau Kettle adalah software dari Pentaho yang dapat digunakan untuk proses ETL (Extraction, Transformation dan Loading). PDI dapat digunakan untuk migrasi data, membersihkan data, loading dari file ke database atau sebaliknya dalam volume besar. PDI menyediakan graphical user interface dan drag-drop komponen yang memudahkan user.

Elemen utama dari PDI adalah Transformation dan Job. Transformation adalah sekumpulan instruksi untuk merubah input menjadi output yang diinginkan (input-proses-output). Sedangkan Job adalah kumpulan instruksi untuk menjalankan transformasi.

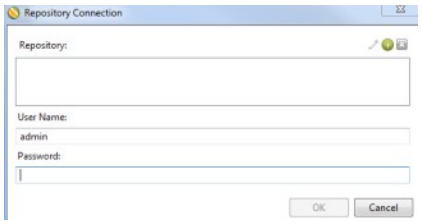
Ada tiga komponen dalam PDI: Spoon, Pan dan Kitchen. Spoon adalah user interface untuk membuat Job dan Transformation. Pan adalah tools yang berfungsi membaca, merubah dan menulis data. Sedangkan Kitchen adalah program yang mengeksekusi job.

Instalasi

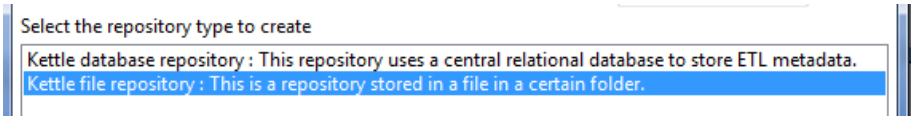
Sebelum menginstall, pastikan sistem telah memiliki JRE (Java Runtime Environtment) minimal versi 1.5. Kemudian download software versi community yang gratis di http://community.pentaho.com/projects/data-integration/ Nama file yang diperoleh akan berbentuk seperti pdi-ce-x.y.z-stable.zip. Ekstrak file zip ke direktori yang diinginkan. Jangan ekstrak ke direktori yang mengandung karakter spasi, '&' dsb.

Kemudian jalankan spoon.bat (atau spoon.sh untuk Linux). Akan muncul dialog untuk repository (gambar bawah). Repository adalah tempat penyimpanan Job dan Transformation. Klik tombol plus hijau di kiri atas untuk menambahkan repository baru. Pada beberapa versi PDI dialog ini tidak muncul, hal ini tidak menjadi masalah dan lanjutkan ke langkah berikutnya.

Catatan: Pada Windows, dialog repository seringkali tertutup aplikasi lain, sedangkan di TaskBar tidak muncul. Solusinya, minimize semua window satu persatu sampai dialog muncul



Repository dapat berbentuk database atau file. Untuk sekarang, buatlah dalam bentuk file.



Isi base direktori dan nama repository

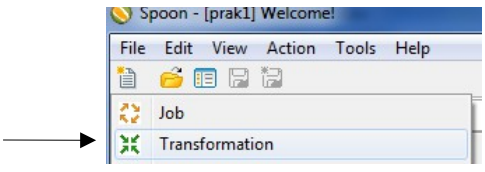
Task Pertama: Ekspor CSV ke XML

Misalnya kita memiliki file berisi informasi nama, alamat, kabupaten/kota dan propinsi dengan format CSV. Kita ingin merubah file tersebut dalam format lain, misalkan XML. Berikut adalah langkah-langkahnya.

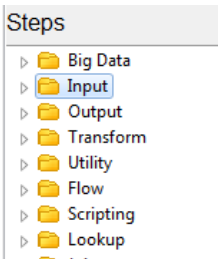
Pertama buat file csv dengan text editor seperti berikut . Simpan dengan nama alamat.csv pastikan tidak ada spasi setelah koma.

```
budi,jalan buah batu,bandung,jawa barat
ahmad,jl rumah sakit,purwakarta,jabar
badu,jl a yani,tegal,jawa tengah
```

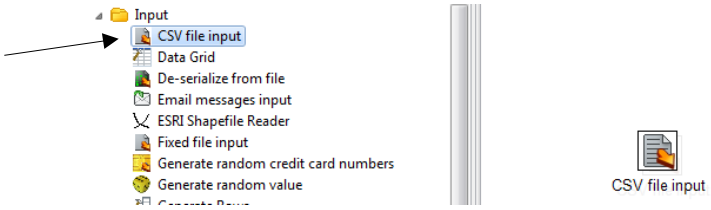
Selanjutnya kembali ke Spoon. Langkah pertama adalah membuat transformation dengan File → New Transformation (Ctrl-N) atau dengan button New di kiri atas → Transformation (gambar bawah)



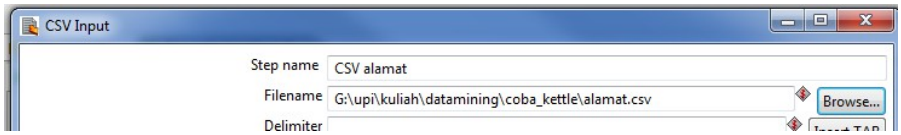
Dapat dilihat pada bagian kiri ada panel yang menampilkan jenis Steps yang disediakan. Step adalah elemen penyusun transformasi, yang dapat berupa input, proses atau output. Silahkan melihat-lihat step apa saja yang disediakan oleh PDI. Dari besar dan beragamnya pilihan steps, terlihat bahwa PDI dapat digunakan untuk transformasi yang kompleks dengan sumber data yang sangat beragam.



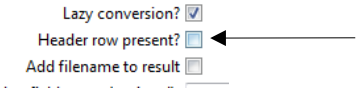
Kembali ke contoh kita, karena input berformat csv dan outputnya XML, maka pilih direktori Input dan pilih step CSV file lalu **drag** (gambar bawah)



Sekarang kita akan mengkonfigurasi step ini, double klik step “CSV file input”. Akan muncul dialog seperti gambar dibawah. Isi step name dan klik tombol “Browse” untuk memilih file csv alamat yang sebelumnya telah dibuat.



Karena file csv ini tidak memiliki header, jangan lupa uncheck “Header row present”



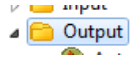
Selanjutnya klik tombol “Get Fields”

Spoon akan menanyakan jumlah sample yang akan digunakan untuk mendapatkan field. Setelah field dibangkitkan, edit sesuai dengan nama field yang cocok dengan lengthnya (gambar bawah)

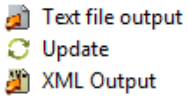
Name	Type	Format	Length
nama	String		50
alamat	String		100
kabu_kota	String		50
propinsi	String		50

Kemudian klik tombol “Preview” untuk melihat keluaran dari proses loading. Setelah selesai, tekan OK.

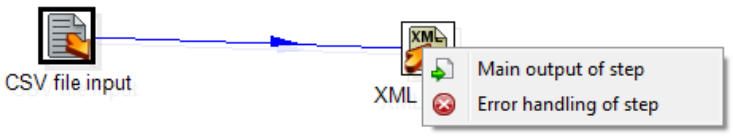
Selanjutnya kita akan menambahkan output XML. Pilih di panel Steps direktori output



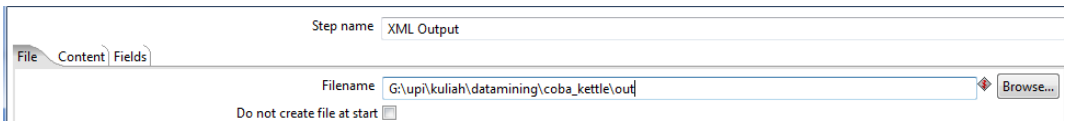
lalu pilih XML Output dan drag ke tab transformasi.



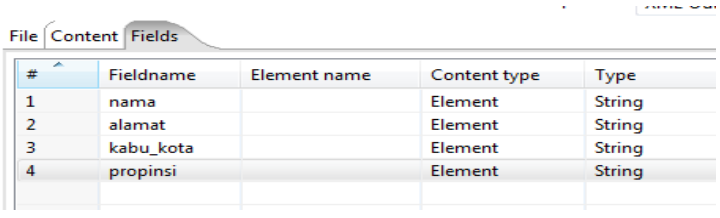
Selanjutnya kita akan menambahkan penghubung antara step input csv dan xml output. Dalam PDI, ini disebut **Hop**. Untuk menambahkan hop, **klik csv input, tekan SHIFT** dan drag ke xml output. Kemudian pilih "Main output of step" (gambar bawah).



Double klik XML output, masukan nama file output yang diinginkan

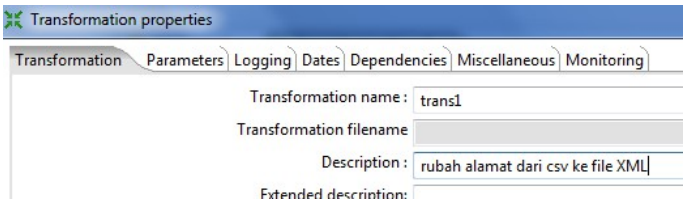


Pilih tab "Fields" (gambar bawah) lalu tekan button "Get Fields". Fields akan terisi. isi jenis **content type**, lalu tekan "OK".

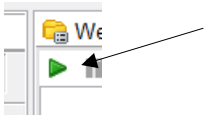


#	Fieldname	Element name	Content type	Type
1	nama		Element	String
2	alamat		Element	String
3	kabu_kota		Element	String
4	propinsi		Element	String

Sebelum kita jalankan, save dulu transformasi ini (Ctrl-S atau icon disk). Isi nama transformasi dan deskripsi.



Sekarang transformasi sudah siap dijalankan. Tekan tombol play dan klik “Launch”



Transformasi akan menghasilkan file XML out.xml, jika dilihat maka outputnya adalah sebagai berikut:

```
<?xml version="1.0" encoding="UTF-8"?>
<Rows>
 <Row><nama>budi</nama> <alamat>jl buah batu no 5</alamat> <kabu_kota>bandung</kabu_kota> <
 <Row><nama>ahmad</nama> <alamat>jl rumah sakit no 10</alamat> <kabu_kota>purwakarta</kabu_
 <Row><nama>badu</nama> <alamat>jl a yani</alamat> <kabu_kota> tegal</kabu_kota> <propinsi>
</Rows>
```

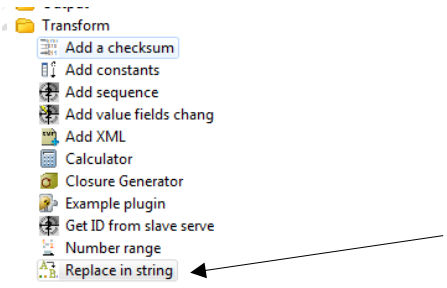
Task Kedua: Filter (Ganti Kata)

Data nama lokasi seringkali tidak konsisten, misalnya untuk propinsi Jawa Barat, sering dituliskan “jabar”. Bagaimana melakukan filter sehingga variasi ini ditulis menjadi satu nama Jawa Barat?

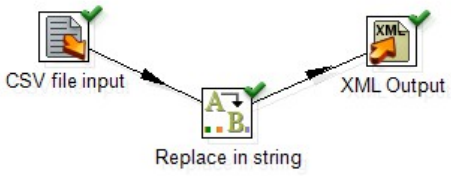
Untuk kasus ini, modifikasi alamat.csv, buat variasi “jabar” yang nantinya akan diganti menjadi Jawa Barat.

```
o 5,bandung,jabar
t no 10,purwakarta,jabar
al, jawa tengah
 bandung,jabar
```

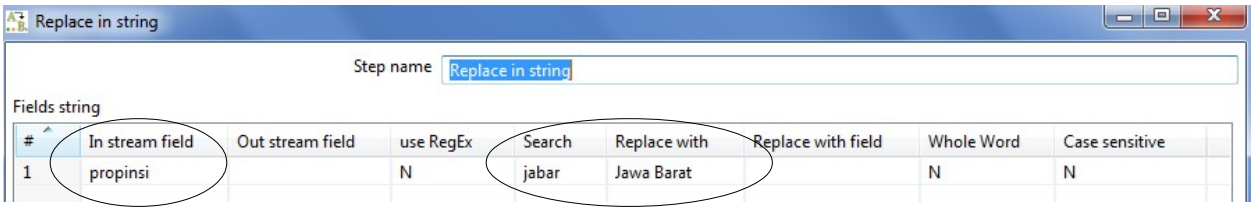
Selanjutnya **hapus hop** (hop saja) yang menghubungkan csv dan xml dan tambahkan step “Replace in String” yang ada di direktori Transform



Kemudian hubungkan antara csv, replace dan xml dengan hop (gambar bawah). Gunakan shift drag untuk membuat hop.

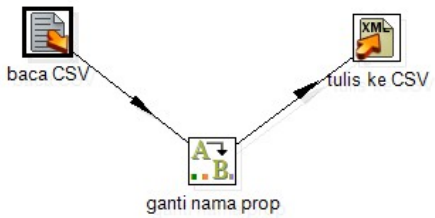


Double click “replace in string” step dan isi “In stream field”, “Search” dan “Replace with” seperti gambar bawah, dan tekan OK.



Jalan transformasi dan dapat dilihat hasilnya semua kemunculan “jabar” menjadi “Jawa Barat”

Untuk lebih memperjelas transformasi, beri nama setiap step dengan nama yang tepat.



Latihan: Buatlah agar transformasi dapat menangani kombinasi “Jabar”, “jabar”, “jawa barat”? Petunjuk: gunakan regex pada step ganti string

Task Ketiga: Output ke database

Pada task pertama dan kedua, output adalah file XML. Bagaimana jika kita menginginkan output dalam tabel database MySQL?

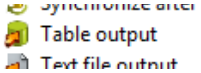
Secara default, PDI tidak mensupport MySQL karena masalah lisensi open source. Ini berbeda dengan Postgre, DB2, SQLite dan database open source lain yang langsung dapat digunakan.

Untuk menambahkan MySQL , download JDBC connector di www.mysql.com/downloads/connector/j/ ekstrak zip dan pindahkan file mysql-connector-java-x.y.z-bin.jar ke direktori [pdi]\data-integration\lib (x,y,z adalah versi dari connector). Restart Spoon agar JDBC ini dapat digunakan.

Pastikan server MySQL anda telah berjalan. Menggunakan phpMyAdmin atau tools yang lain, buat database kemudian tabel seperti berikut:

```
CREATE TABLE alamat (
  id int not null auto_increment primary key,
  nama varchar(50) not null,
  alamat varchar(50) not null,
  str_kabu_kota varchar(50),
  str_propinsi varchar(50)
)
```

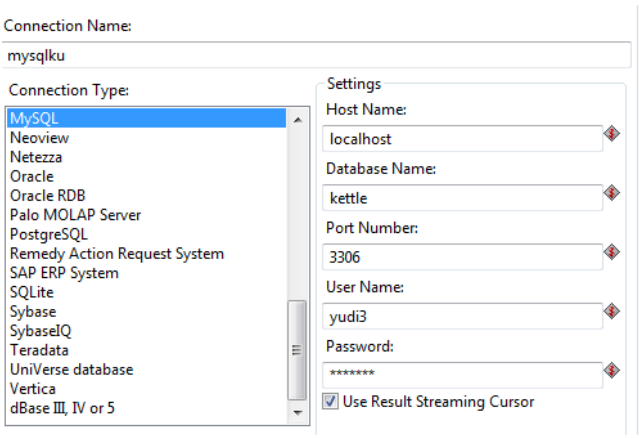
Kembali ke Spoon, buatlah transformasi baru, lalu dengan cara yang sama seperti task 1 dan task 2, tambahkan step input csv. Sedangkan untuk output, pilih Table output.



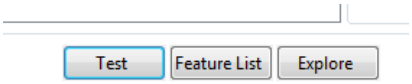
Hubungkan kedua step ini denga hop (shift-drag)

Double clik step “Table Output”, akan muncul dialog seperti di bawah, klik “new” untuk membuat koneksi ke database.

Pilih MySQL sebagai connection type. Perhatikan juga tipe koneksi yang lain yang dapat digunakan. Isi nama host, nama database, port (biasanya tidak perlu diubah), username dan password. Jika menggunakan XAMPP, default username adalah 'root' dengan password dikosongkan.



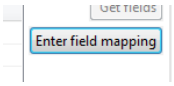
Untuk memastikan koneksi sudah berhasil, tekan tombol test.



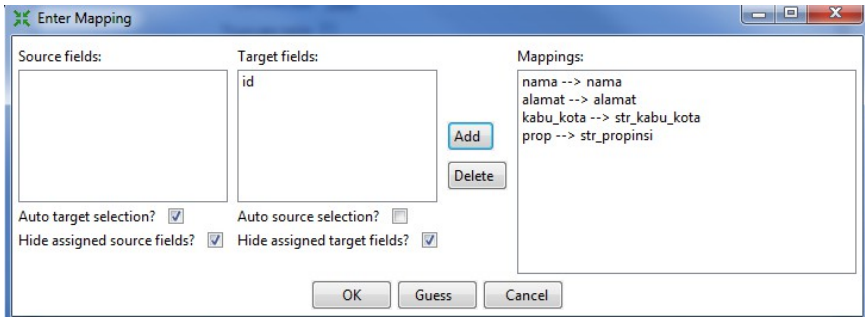
Isi target table dengan “alamat” sesuai dengan nama tabel yang dibuat.

Karena nama field input tidak sama dengan nama field pada tabel, jangan lupa check “specify database field”

Masuk ke “Database fields” lalu klik “Enter field mapping” untuk mendefinisikan hubungan antara field input dan field output.



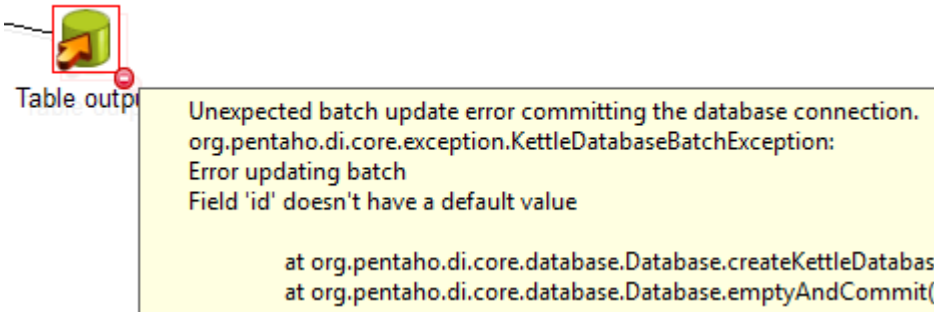
Petakan antara source dengan target.



Verifikasi transformasi dengan  lalu jalankan transformasi 

Jika semua berjalan lancar, maka akan keluar tanda check hijau pada tabel output. Periksa tabel untuk hasil.

Jika terjadi error, hover kursor ke tanda merah (gambar bawah), sebagai contoh, jika id tidak diset auto_increment akan muncul pesan seperti dibawah.



Task Keempat: Lookup tabel

Jika melihat tabel yang dihasilkan pada task tiga. Terlihat bahwa propinsi yang dihasilkan masih berupa string, ini dapat berbahaya karena bisa terjadi variasi (misalnya “Jogjakarta” denga “Yogyakarta”) sehingga propinsi yang sama dapat dianggap sebagai dua propinsi yang berbeda. Solusinya adalah membuat tabel lookup yang mengkonversi nama propinsi menjadi sebuah kode yang konsisten, seperti 1 untuk Jawa Barat dan 2 untuk Jawa Tengah.

Pertama kita akan menambahkan tabel lookup ke dalam database. Eksekusi query berikut untuk menambahkan tabel lookup dan datanya:

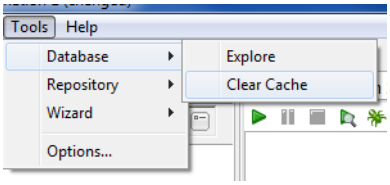
```
CREATE TABLE `lookup_propinsi` (
  `kode_propinsi` int(10) NOT NULL,
  `nama_propinsi` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `lookup_propinsi` (`kode_propinsi`, `nama_propinsi`) VALUES
       (1, 'Jawa Barat'),
       (2, 'Jawa Tengah');
```

Tabel alamat juga perlu ditambahkan kode propinsi ini, lakukan query berikut untuk menambahkan field kode_propinsi

```
alter TABLE alamat
  add kode_propinsi int;
```

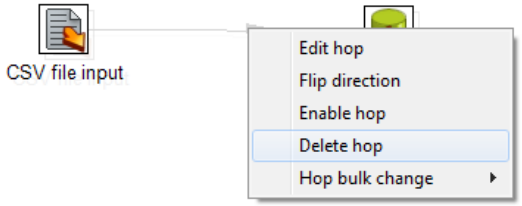
Apa yang ingin kita capai adalah membaca file CSV, lalu mengubah “jabar” menjadi kode 1, “jawa tengah” menjadi kode 2 dan seterusnya dan menyimpan hasilnya ke dalam field kode_propinsi di tabel alamat.

Kembali ke Spoon, agar perubahan database ini muncul di Spoon, bersihkan dulu cache dengan cara Tools→Database→Clear Cache:

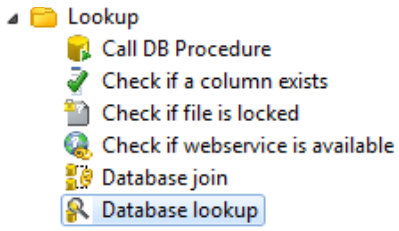


Jangan lupa lakukan save-as pada transformasi task 3.

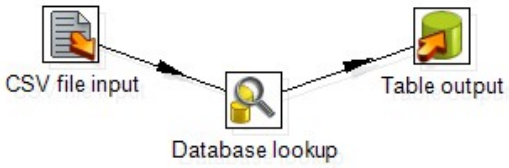
Selanjutnya hapus hop yang menghubungkan file input dengan tabel output.



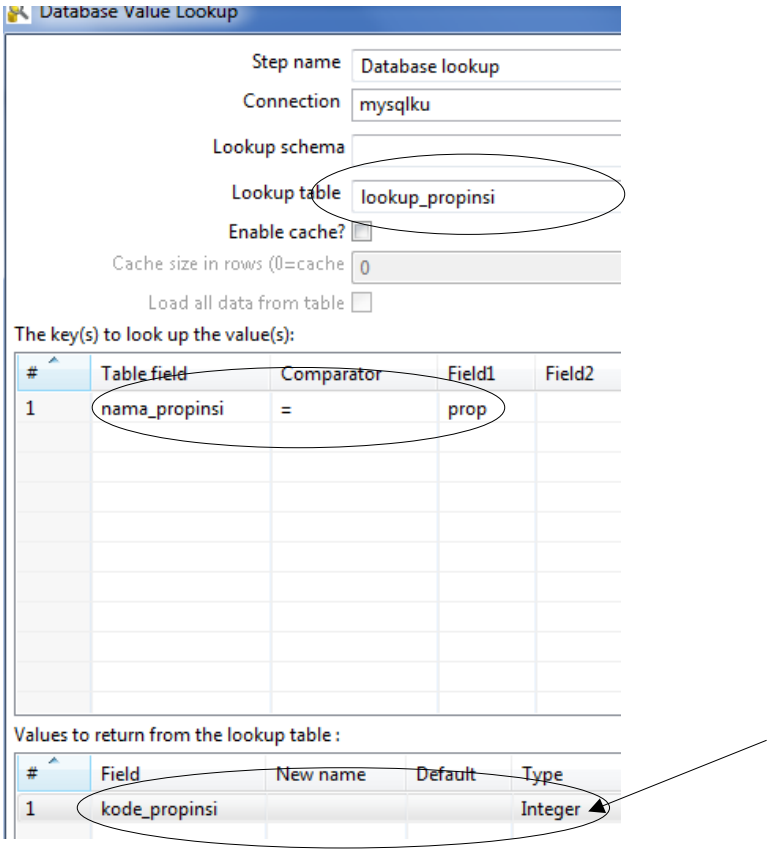
Lalu tambahkan step database lookup



dan buat hop yang menghubungkannya dengan input dan output.



Double klik “database lookup” untuk mengedit property. Isi lookup table, key dan lookup **dan jangan lupa tipe-nya diisi** (gambar bawah)



Double klik “table output” untuk menambahkan field kode_propinsi, jika belum muncul di dropbox, ketikkan secara manual.

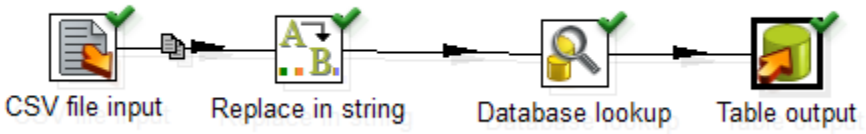
	Table field	Stream field
	nama	nama
	alamat	alamat
	str_kabu_kota	kabu_kota
	str_propinsi	prop
	kode_propinsi	kode_propinsi

Seperti biasa, lakukan verifikasi terlebih dulu kemudian jalankan. Jangan lupa lakukan tools → database → clear-cache setiap ada perubahan di database.

Setelah dijalankan maka kode_propinsi akan berisi sesuai dengan tabel lookup, lihat gambar bawah:

4	budi	jalan buah batu	bandung	jawa barat	1
5	ahmad	jl rumah sakit	purwakarta	jabar	(NULL)
6	badu	jl a yani	tegal	jawa tengah	2

Kenapa
jabar kosong? Karena di tabel lookup hanya berisi kode untuk "Jawa Barat". Solusinya adalah dengan menambah komponen replace string seperti pada task 2. Sehingga konfiigurasi steps-nya jadi seperti ini:



Jika kode_propinsi tidak terisi padahal nilai sudah benar, dapat disebabkan penggunaan spasi pada data csv setelah koma. " jawa barat" (ada spasi di depan) dengan "jawa barat" akan dianggap dua string yang berbeda. Untuk mengatasi hal ini adalah dengan menggunakan step Transformation → "String operations" dan meng-'trim' semua field sebelum melalui proses lookup.

Latihan: Buatlah lookup untuk field kabupaten dan kota

Task Kelima: Mengisi Nilai Lookup

Sebagai contoh kita memiliki file MS Excel dengan isi sebagai berikut

TAHUN	SMT	NIM	NAMA	KODEPST	NAMAPST	JENJANG	KODEMK	NAMAMK	SKS	KODEDSN	NAMADSN
2011/2012	1	040203	SUSILAWATI	D0151	PENDIDIKAN MATEMATIKA	S1	MA301	FISIKA UMUM	3	1810	ANDI SUHA
2011/2012	1	040203	SUSILAWATI	D0151	PENDIDIKAN MATEMATIKA	S1	MT310	KAPITA SELEKTA MA	3	0519	DRS.H. KAI
2011/2012	1	040203	SUSILAWATI	D0151	PENDIDIKAN MATEMATIKA	S1	MT419	PENGANTAR TOPOI	3	0518	DARHIM, P
2011/2012	1	040203	SUSILAWATI	D0151	PENDIDIKAN MATEMATIKA	S1	MT317	PROGRAM LINEAR	3	1911	LUKMAN, S
2011/2012	1	040203	SUSILAWATI	D0151	PENDIDIKAN MATEMATIKA	S1	MT401	SEMINAR PENDIDIK	2	1656	DR. ELAH N
2011/2012	1	040203	SUSILAWATI	D0151	PENDIDIKAN MATEMATIKA	S1	MT598	SKRIPSI	6		
2011/2012	1	040203	SUSILAWATI	D0151	PENDIDIKAN MATEMATIKA	S1	MT599	UJIAN SIDANG	0		
2011/2012	1	040236	FIPIT PEBRIANI W	D0451	PENDIDIKAN KIMIA	S1	KI598	SKRIPSI	6		
2011/2012	1	040236	FIPIT PEBRIANI W	D0451	PENDIDIKAN KIMIA	S1	KI599	UJIAN SIDANG	0		

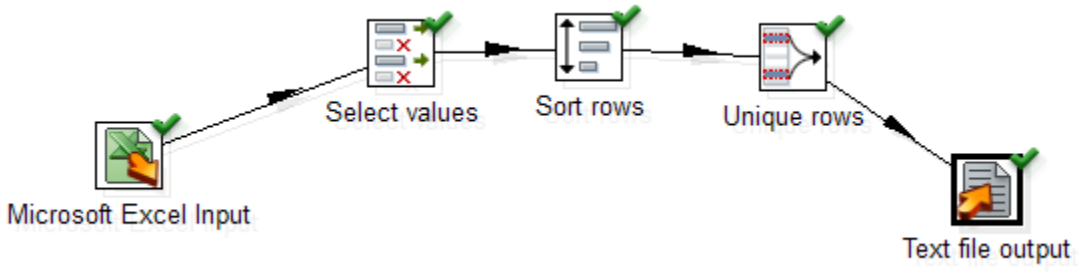
Dapat dilihat data diatas kondisinya tidak ternormalisasi (data nama program studi terduplikasi). Sebelum dipindahkan ke dalam database, perlu dibuat dulu tabel lookup untuk program studi, jenjang, matakuliah dan dosen. Kita dapat menggunakan cara pada task 4, dengan mengisi sendiri tabel lookup propinsi, tapi masalahnya nama propinsi tetap dan jumlahnya sedikit, sedangkan nama dosen dan nama matakuliah jumlahnya bisa banyak. Pengisian secara manual membutuhkan waktu lama.

Solusinya adalah mengisi tabel lookup ini secara otomatis.

Kita mulai dari lookup program studi (KODEPST, NAMAPST). Langkah-langkahnya adalah sebagai berikut:

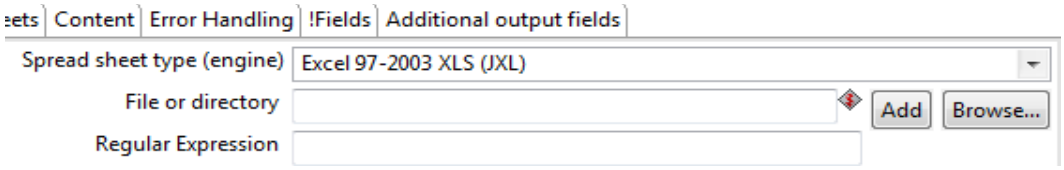
1. Input file excel
2. Transformasi dengan menghilangkan field selain KODEPST dan NAMAPST
3. Sort, karena sebelum diambil nilai row unik, harus disort terlebih dulu.
4. Ambil row yang unik
5. Tulis ke tabel lookup

Untuk itu perlu digunakan step Microsoft Excel Input (input), Select Value (transform), Sort Rows (transform), Unique Rows (transforms) dan tabel output (output). Skemanya sebagai berikut:

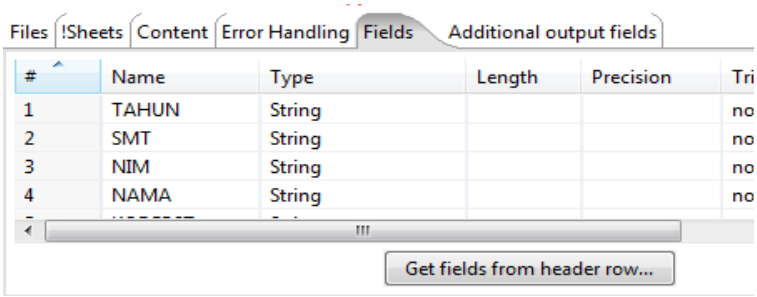


Pertama, buat file excel seperti pada contoh diatas atau minta pada asisten praktikum.

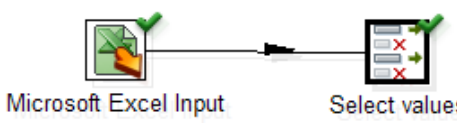
Drop step input: Microsoft Excel Input, edit propertynya. Klik “browse” (gambar bawah), pilih file Excel kemudian klik “add”.



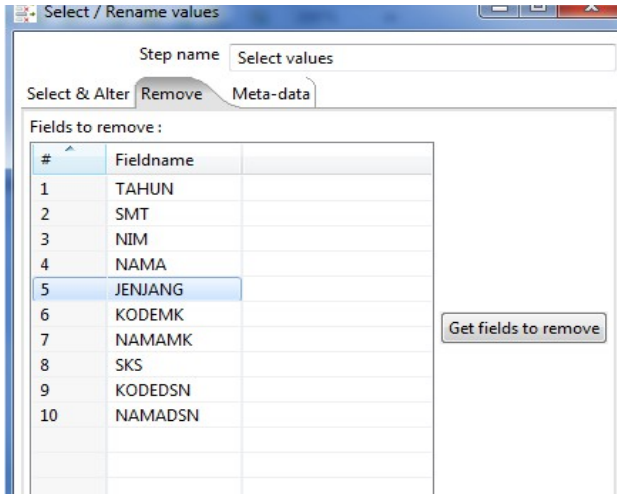
Kemudian klik tab “Fields” dan klik “Get fields from header row” (gambar bawah).



Selanjutnya tambahkan steps transform → “Select Values” , dan tambahkan hop. Sehingga seperti gambar dibawah:

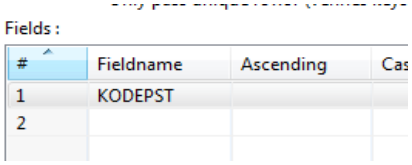


Edit property “Select Value”, **masuk ke tab “REMOVE”,** tambahkan semua field **kecuali** kode_pst dan nama_pst (gambar bawah). Efek step ini adalah membuang semua field kecuali yang berkaitan dengan program studi.

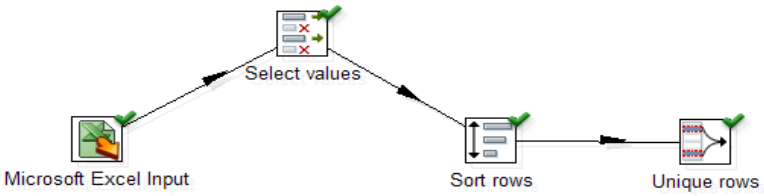


Selanjutnya tambahkan step untuk mensort, transform → “Sort rows” (gambar bawah). Step ini diperlukan sebelum row yang unik diambil.

Edit property sort rows, tambahkan bahwa pengurutan berdasarkan field KODEPST



Kemudian tambahkan Transform → “Unique row” dan hop seperti gambar bawah. Propertynya tidak perlu di-edit.

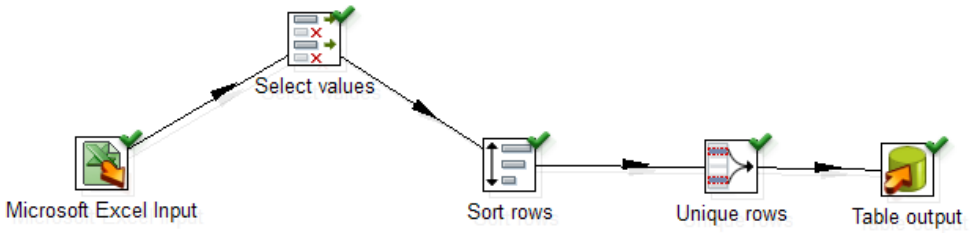


Terakhir, tambahkan Output → “Table Output” seperti pada task2. Isi koneksi dan nama tabelnya.

Struktur tabel adalah sebagai berikut:

```
create table prodi (kodepst varchar(20), namapst varchar(50));
```

Sedangkan skema stepsnya adalah sebagai berikut:



Jalankan transformasi. Maka tabel prodi akan berisi sebagai berikut, jika masih ada nama yang ganda, itu disebabkan kode-nya yang memang berbeda:

kodepst	namapst
D0151	PENDIDIKAN MATEMATIKA
D0155	PENDIDIKAN MATEMATIKA
D0251	PENDIDIKAN FISIKA
D0252	PENDIDIKAN FISIKA
D0255	PENDIDIKAN FISIKA
D0351	PENDIDIKAN BIOLOGI
D0355	PENDIDIKAN BIOLOGI
D0451	PENDIDIKAN KIMIA
D0455	PENDIDIKAN KIMIA

Job

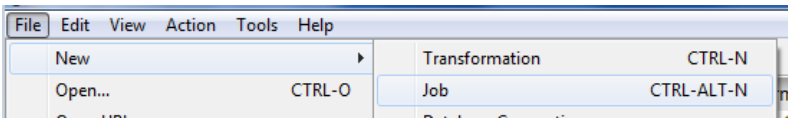
Pada task pertama sampai dengan kelima, kita telah membuat transformasi data menjadi berbagai bentuk. Semua transformasi tersebut masih dijalankan secara manual. Padahal salah satu karakter utama dari ETL adalah proses harus dibuat seotomatis mungkin. Belum lagi jika ada berbagai transformasi yang harus dikombinasikan.

Dalam PDI, Job digunakan untuk mengkoordinasikan proses ETL. Fungsi Job adalah:

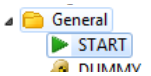
- Mengatur urutan transformasi.
- Penjadwalan transformasi.
- Pengecekan kondisi sebelum dilakukan transformasi. Misalnya apakah file atau tabel input sudah tersedia.
- Pengelolaan file (FTP, copy, delete file)
- Mengirimkan notifikasi melalui email.

Sekarang kita akan membuat sebuah job sederhana. Job ini menjalankan transformasi pada task 4, tapi dengan pengecekan apakah file csv input ada.

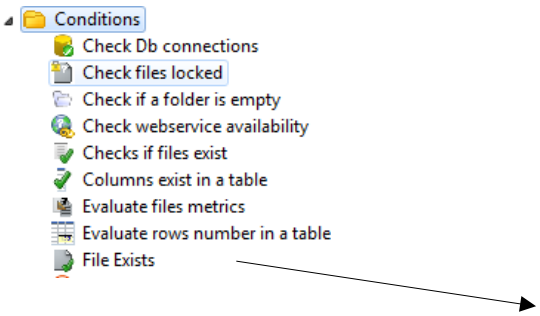
Pilih New → Job atau CTRL-ALT-N.



Ambil step General → Start.

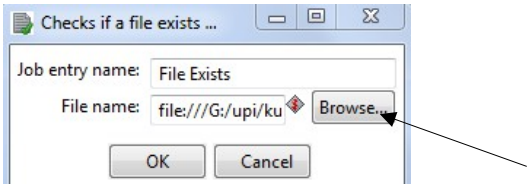


Ambil step Conditions → File Exits. Step ini untuk mengecek apakah file input sudah ada.

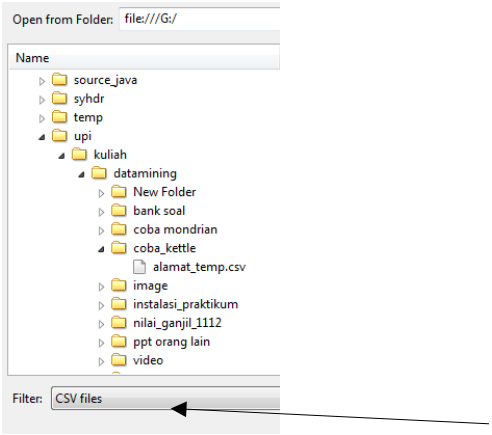


Buat hop antara start dan file exist (gambar bawah)

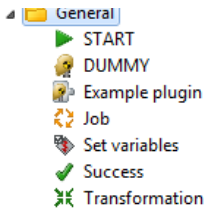
Double klik step "File Exits", lalu klik browse.



Pilih filter "CSV Files" dan pilih file csv alamat.



Selanjutnya tambahkan step General → Transformation untuk menambahkan transformasi yang akan kita eksekusi.



Tambahkan hop yang menghubungkan transformasi dengan file exist.

Double klik transformation. Pilih specify by reference, dan pilih transformasi pada task 4. Atau anda dapat memilih tranformation by file name dan pilihlah file transformation yang sesuai.

Coba jalankan Job ini  Silahkan coba hilangkan file csv dan perhatikan efek yang terjadi.

Dari pembahasan Job sederhana ini terlihat Job dapat digunakan untuk mengatur aliran transformasi. Anda dapat menambahkan banyak transformasi dengan berbagai aliran bergantung kondisi yang ada.