

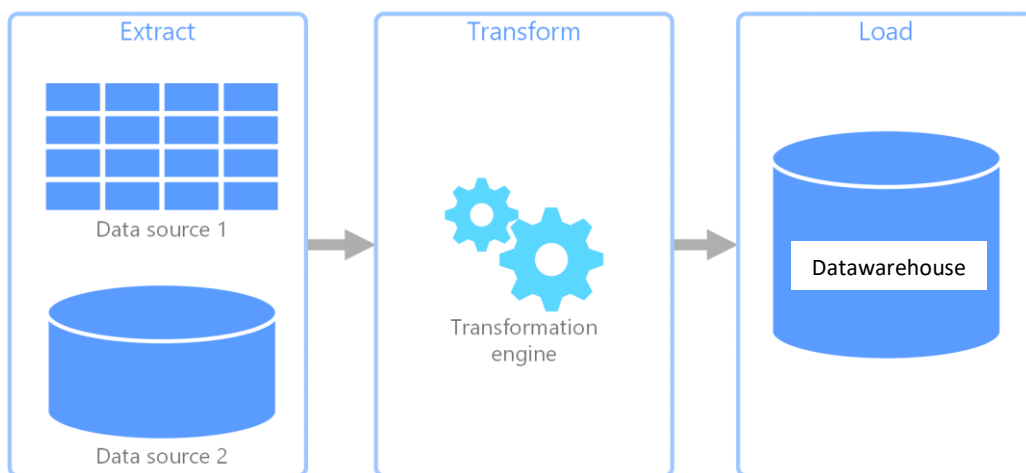
MODUL PRAKTIKUM 3

EKSTRAKSI TRANSFORMASI LOAD DATA

PRAKTIKUM DATA WAREHOUSE DAN BUSINESS INTELLIGENT

Dasar Teori

Extraction - Transformation – Loading (ETL) merupakan sebuah proses penting yang merupakan bagian dari Data Warehouse yang berfungsi untuk mentransfer data yang ada pada basisdata transaksi kedalam basisdata Data Warehouse yang sudah dimodelkan sedemikian rupa berdasarkan hypercubes yang dibentuk berdasarkan laporan - laporan yang sering dipakai manajemen tingkat atas.



Gambar Proses ETL

EKSTRAKSI

Ekstraksi (Extraction) adalah operasi ekstraksi data dari sebuah sistem sumber untuk digunakan lebih jauh dalam lingkungan Data Warehouse. Tahapan ini adalah yang paling pertama dalam proses ETL. Setelah Ekstraksi, data ini akan ditransformasikan dan di-load ke dalam Data Warehouse.

Metode Ekstraksi Secara Umum :

1. Statis, umumnya dilakukan pada saat pemuatan data awal dan dilakukan pada saat sistem OLTP tidak berjalan.
2. Terjadwal, ada 2 model pengambilan data terjadwal yaitu :
 - a. Berdasarkan waktu, setiap pengaksesan record basisdata OLTP ditandai dengan waktu berupa tanggal dan jam dan secara terjadwal sebuah aplikasi program akan dijalankan untuk mengakses Data Warehouse berdasarkan perubahan waktu pada basisdata OLTP.
 - b. Berdasarkan perbedaan berkas lama dan baru, adanya backup harian terhadap basis data OLTP, dan secara terjadwal sebuah aplikasi program akan dijalankan untuk mengakses Data Warehouse jika sebuah tabel hari ini dan duplikat tabel hari sebelumnya berbeda.

3. Seketika, ada 3 model pengambilan data seketika yaitu dengan mekanisme log transaksi, mekanisme basis data trigger dan sumber aplikasi dengan modifikasi program aplikasi OLTP.

Metode Ekstraksi Fisik (physical extraction):

1. Online Extraction
Data diekstrak langsung dari sistem sumber itu sendiri. Proses ekstraksi dapat berhubungan secara langsung dengan sistem sumber untuk mengakses tabel sumber atau ke sebuah sistem perantara yang menyimpan data dengan sebuah cara yang dikonfigurasi terlebih dahulu (sebagai contoh log atau tabel perubahan). Dengan catatan bahwa sistem perantara secara fisik tidak berbeda dari sistem sumber.
2. Offline Extraction
Data tidak diekstrak secara langsung dari sistem sumber namun dibatasi secara eksplisit diluar sistem sumber orisinil. Data telah memiliki struktur atau telah dibuat melalui prosedur ekstraksi.

TRANSFORMASI

Proses pengambilan data mentah yang belum bisa disimpan pada Data Warehouse, oleh karena itu data harus sesuai standar struktur Data Warehouse yang telah ditentukan sehingga bisa disimpan ke Data Warehouse. Transformation data terdiri dari beberapa tahap yaitu:

- a. Seleksi, men- select record dari tabel basisdata OLTP, tahap ini merupakan bagian dari proses pengambilan data.
- b. Pemisahan dan Penggabungan, manipulasi data yang dibutuhkan untuk men- select record OLTP, yaitu melakukan proses pemisahan dan penggabungan bila dibutuhkan.
- c. Konversi, dilakukan untuk dua alasan yaitu:
 - Standarisasi pengambilan data dari berbagai sumber
 - Membuat field dapat digunakan Data Warehouse dan dipahami oleh pengguna
- d. Ringkasan, data yang terlalu detail tidak selalu dibutuhkan pada Data Warehouse oleh karena itu perlu diringkas berdasarkan kebutuhan Data Warehouse.
- e. Pengayaan, menggunakan sebuah atau beberapa field untuk membuat hasil data yang terbaik pada Data Warehouse, prinsip ini merupakan pengembangan dari kumpulan sebuah atau beberapa field dari beragam record yang menghasilkan sebuah field untuk Data Warehouse.

Fungsi Transformasi yaitu, revisi format, penghilangan pengkodean field, menghitung dan menghasilkan nilai, memecah field, penggabungan informasi, konversi kumpulan karakter konversi unit pengukuran, konversi tanggal dan jam, ikhtisar, restrukturisasi kunci dan deduplikasi

LOADING

Merupakan tahapan yang berfungsi untuk memasukkan data ke dalam target akhir, yaitu ke dalam suatu data warehouse

Terdapat 4 Fase Load yaitu :

- Panggil, jika data sudah ada pada tabel Data Warehouse maka proses panggil ini akan menghapus data yang sudah ada dan menggantinya, jika data belum ada maka proses ini akan mengisi tabel Data Warehouse.
- Tambah, jika data sudah ada pada tabel Data Warehouse maka proses tambah ini akan menambah data dan ada kemungkinan terdapat duplikat record dan jika dikehendaki dimungkinkan duplikat record ditolak.
- Destructive Merge, jika kunci record yang datang cocok dengan kunci record yang ada maka akan merubah record yang ada dan jika record yang datang adalah record baru maka akan menambah record baru.
- Constructive Merge, jika kunci record yang datang cocok dengan kunci record yang ada maka akan menambah record baru dan menandai record baru tersebut sebagai penghubung ke record lama yang cocok dengan record yang akan datang.

PENGENALAN TRANSFORMATION PADA PENTAHO

Transformasi adalah jaringan tugas logis yang disebut **Steps**. Transformasi pada dasarnya adalah aliran data. Dalam contoh di bawah ini, pengembang database telah membuat transformasi yang membaca file data, memfilternya, mengurutkannya, dan memuatnya ke tabel database relasional. Misalkan pengembang database mendeteksi kondisi kesalahan dan alih-alih mengirim data ke langkah Dummy, (yang tidak melakukan apa-apa), data dicatat kembali ke tabel.



Dua komponen utama yang terkait dengan transformasi adalah steps dan hops:

- Steps adalah blok transformasi, misalnya input file teks atau output tabel. Ada lebih dari 140 steps yang tersedia di Pentaho Data Integration dan mereka dikelompokkan menurut fungsinya; misalnya input, output, scripting, dan sebagainya. Setiap steps dalam transformasi dirancang untuk melakukan tugas tertentu, seperti membaca data dari flat file, memfilter baris, dan masuk ke database seperti yang diperlihatkan dalam contoh di atas. Langkah-langkah dapat dikonfigurasi untuk melakukan tugas yang Anda perlukan.
- Hop adalah jalur data yang menghubungkan steps dan memungkinkan metadata skema untuk berpindah dari satu step ke step lainnya. Hop menentukan aliran data melalui steps tetapi tidak harus jalannya berurutan. Saat Anda menjalankan transformasi, setiap langkah dimulai di threadnya sendiri dan mendorong serta meneruskan data.

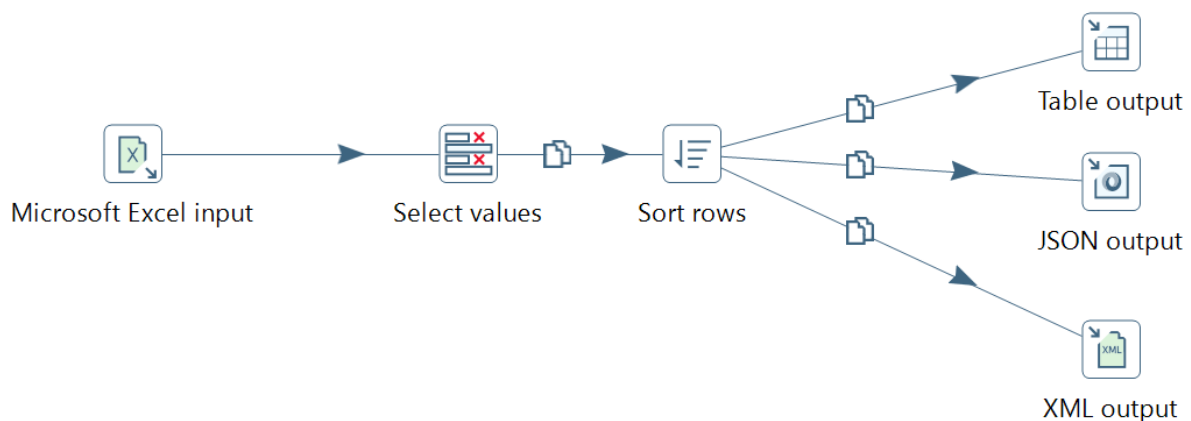
Cara Membuat Transformasi Baru : File ➔ New ➔ Transformation

1. SKEMA PROSES ETL 1

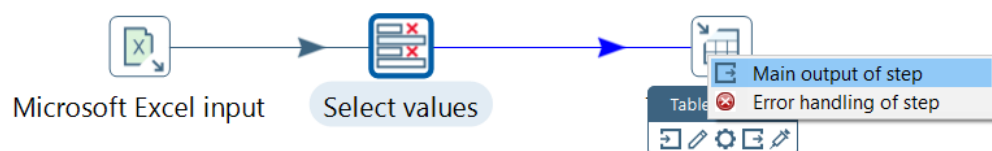
1. Buatlah tabel pada aplikasi Ms. Excel seperti gambar dibawah ini.

	A	B	C
1	Provinsi	Ibukota	Populasi
2	sulut	Manado	417586
3	Sulawesi Tengah	Palu	268644
4	sulsel	Makasar	1168258
5	Sulawesi Tenggara	Kendari	226056
6	sulbar	Mamuju	114101
7	Gorontalo	Gorontalo	138354

2. Buatlah **New Transformation** dengan membuat koneksi antara step dan hop seperti desain dibawah ini:



- Step **Microsoft Excel Input** berada di perluasan node **Input**
- Step **Select values** berada diperluasan node **Transform**
- Step **Sort rows** berada di perluasan node **Transform**
- Step **table output** berada diperluasan node **Output**
- Step **JSON Output** berada diperluasan node **Output**
- Step **XML Output** berada diperluasan node **Output**
- Saat menghubungkan tiap steps dengan hops, antara Step yang memiliki aliran data seperti **Select Values** atau **Sort Rows** ke **Table Output** pilih **Main output of step**.



- Dikarenakan saat menghubungkan **Step yang memiliki percabangan lebih dari dua hops**, memiliki dua pilihan pembagian data, yaitu:
 - **Copy**: membuat salinan dari seluruh kumpulan data dan mengirimkannya masing-masing ke setiap aliran keluaran.
 - **Distribute**: baris set data didistribusikan atau dibagikan pada step tujuan. Step tersebut berjalan di thread terpisah, jadi distribusi adalah cara untuk mengimplementasikan pemrosesan paralel.
- Praktikum ini menggunakan Copy sebagai penghubungnya.

3. Pada step **Microsoft Excel Input**, penjelasan Step ini bisa dilihat di: [https://help.pentaho.com/Documentation/8.2/Products/Data Integration/Transformation Step Reference/Microsoft Excel Input](https://help.pentaho.com/Documentation/8.2/Products/Data%20Integration/Transformation%20Step%20Reference/Microsoft%20Excel%20Input)

- Pada Tab **File**,
 - Spread sheet type (engine): Excel 2007 XLSX (Apache POI)
 - File or directory: Klik **Browse** untuk menentukan lokasi file excel yang dibuat tadi, setelah itu Klik **Add**.

The screenshot shows the 'Microsoft Excel input' configuration window with the 'File' tab selected. The 'Step name' is 'Microsoft Excel input'. The 'Spread sheet type (engine)' is set to 'Excel 2007 XLSX (Apache POI)'. The 'File or directory' field is empty, with 'Add' and 'Browse...' buttons. The 'Regular Expression' and 'Exclude Regular Expression' fields are empty. The 'Password' field is empty. The 'Selected files' table has one entry: '1 D:\PNP\DWBI\DataExcel.xlsx'. The 'Accept filenames from previous steps' section has a checkbox 'Accept filenames from previous step' which is unchecked. The 'Step to read filenames from' and 'Field in the input to use as filename' dropdowns are empty. The 'Show filename(s)...' button is visible. At the bottom are 'Help', 'OK', 'Preview rows', and 'Cancel' buttons.

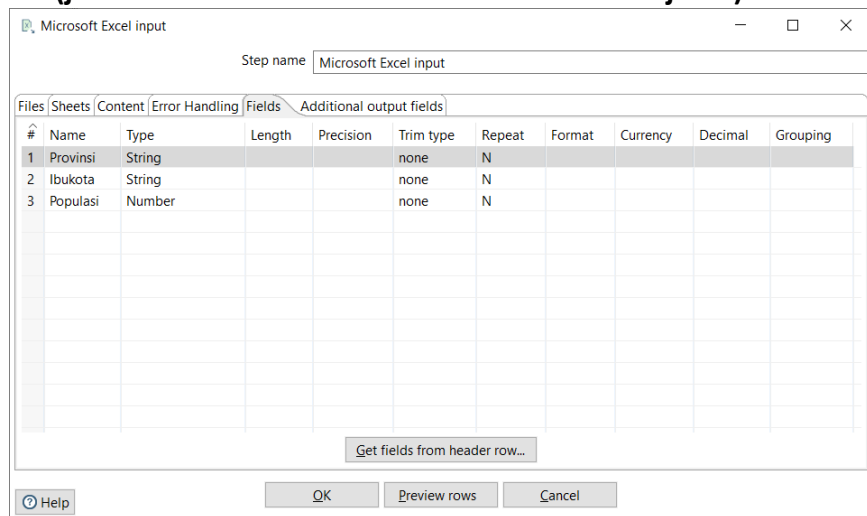
#	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1	D:\PNP\DWBI\DataExcel.xlsx			N	N

- Pada Tab **!Sheets**,
 - Klik **Get sheetname(s)**, untuk memilih sheet yang akan kita ambil datanya. Pilih nama Sheet(s) nya setelah itu tekan tombol ">" untuk memindahkan memilih Sheet(s) tersebut.

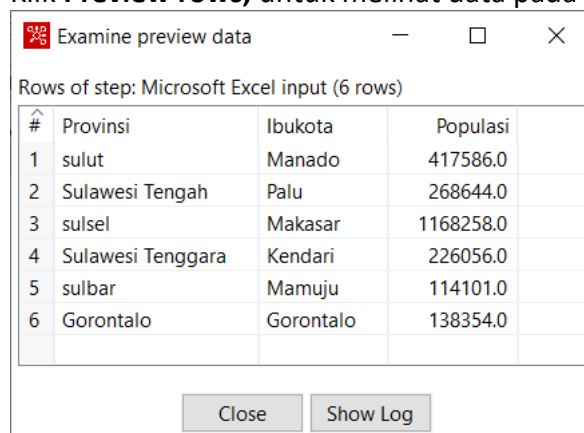
The screenshot shows the 'Microsoft Excel input' configuration window with the '!Sheets' tab selected. The 'Step name' is 'Microsoft Excel input'. The 'List of sheets to read' table has one entry: '1 Sheet1'. The 'Get sheetname(s)...' button is highlighted. At the bottom are 'Help', 'OK', 'Preview rows', and 'Cancel' buttons.

#	Sheet name	Start row	Start column
1	Sheet1		

- Pada Tab **Fields**,
 - Pilih fields yang akan digunakan dengan tombol **Get fields from header row (jika data kita memiliki header row atau baris judul)**.

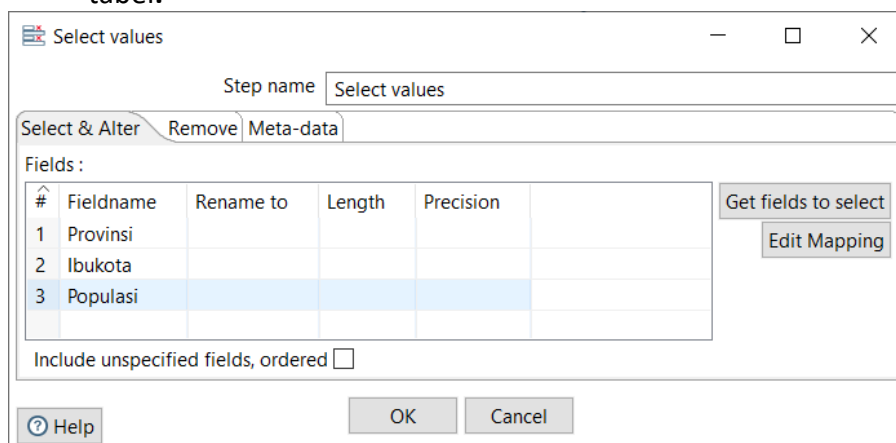


- Klik **Preview rows**, untuk melihat data pada Excel yang kita buat tadi.



4. Pada step **Select values**, penjelasan tentang step ini bisa dilihat di: [https://help.pentaho.com/Documentation/8.2/Products/Data Integration/Transformation Step Reference/Select Values](https://help.pentaho.com/Documentation/8.2/Products/Data%20Integration/Transformations/Step%20Reference/Select%20Values)

- Pada Tab **Select & Alter**,
 - Tombol **Get fields to select** untuk mengambil semua field yang ada pada tabel.



- Pilih Fieldname kolom yang mana datanya akan dipakai untuk step -step berikutnya.

5. Pada step **Sort Rows**, penjelasan tentang step ini bisa dilihat di:
https://help.pentaho.com/Documentation/9.1/Products/Sort_rows

Sort rows

Step name

Sort rows

Sort directory

%%java.io.tmpdir%%

Browse...

TMP-file prefix

out

Sort size (rows in memory)

1000000

Free memory threshold (in %)

Compress TMP Files?

☐

Only pass unique rows? (verifies keys only)

☒

Fields:

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	Populasi	Y	N	N	0	N

Help

OK

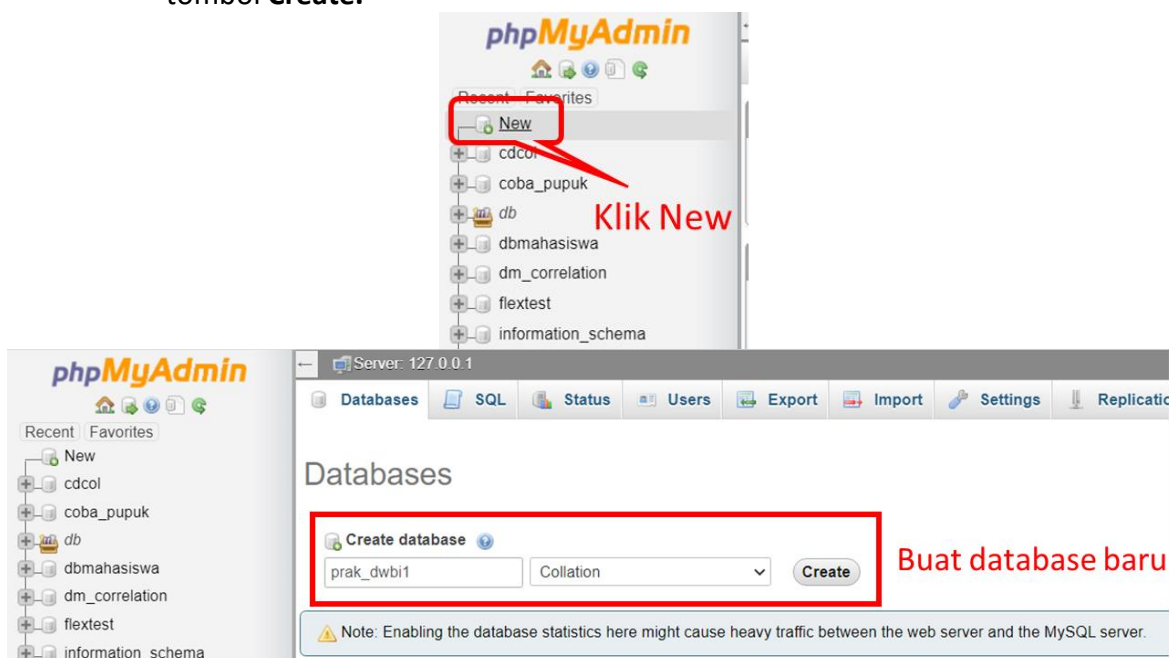
Cancel

Get Fields

Kita hanya ingin mengurutkan data secara Ascending berdasarkan kolom Populasi.

6. Pada step **Table output**, penjelasan tentang step ini bisa dilihat di: [https://help.pentaho.com/Documentation/8.2/Products/Data Integration/Transformation Step Reference/Table Output](https://help.pentaho.com/Documentation/8.2/Products/Data%20Integration/Transformation%20Step%20Reference/Table%20Output)

- Aktifkan aplikasi **XAMPP** pada komputer anda, Start modul **Apache** dan **MySQL**
- Klik tombol **Admin** pada modul **MySQL**, maka pada browser anda akan diarahkan ke <http://localhost/phpmyadmin/>
- Buat Database baru, Langkahnya: Klik New pada bagian kiri, inputkan nama database yang ingin dibuat (disini saya isi dengan nama **prak_dwbi1**) dan klik tombol **Create**.



- Pada Field **Connection**, klik tombol **New**, selanjutnya Buat Koneksi Database seperti yang dilakukan praktikum sebelumnya.

Table output

Step name: Table output

Connection: prak_transform1_conn [Edit...] **New...** Wizard...

Target schema: [Browse...]

Target table: [Browse...]

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☐

Main options Database fields

Partition data over tables: ☐

Partitioning field: [v]

Partition data per month: ☒

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

Field that contains name of table: [v]

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key field: [v]

[Help] [OK] [Cancel] [SQL]

- Buat Koneksi Database baru seperti contoh dibawah ini:

Database Connection

General Advanced Options Pooling Clustering

Connection name: prak_transform1_conn

Connection type:

- MariaDB
- MaxDB (SAP DB)
- MonetDB
- MySQL**
- Native Mondrian
- Neoview
- Netezza
- Oracle
- Oracle RDB
- Palo MOLAP Server
- Pentaho Data Services
- PostgreSQL
- Redshift
- Remedy Action Request System

Access:

- Native (JDBC)**
- ODBC
- JNDI

Settings

Host Name: localhost

Database Name: prak_dwbi1

Port Number: 3306

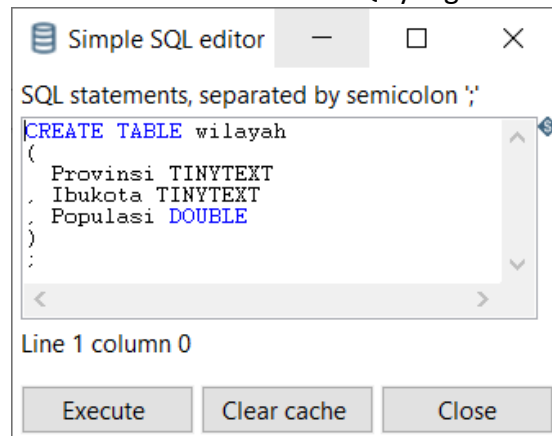
Username: root

Password:

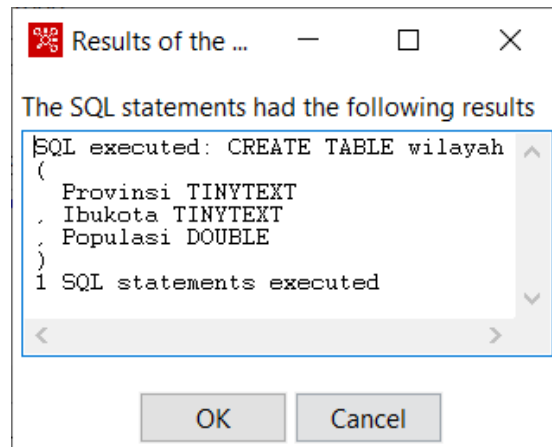
☒ Use Result Streaming Cursor

[Test] [Feature List] [Explore] [OK] [Cancel]

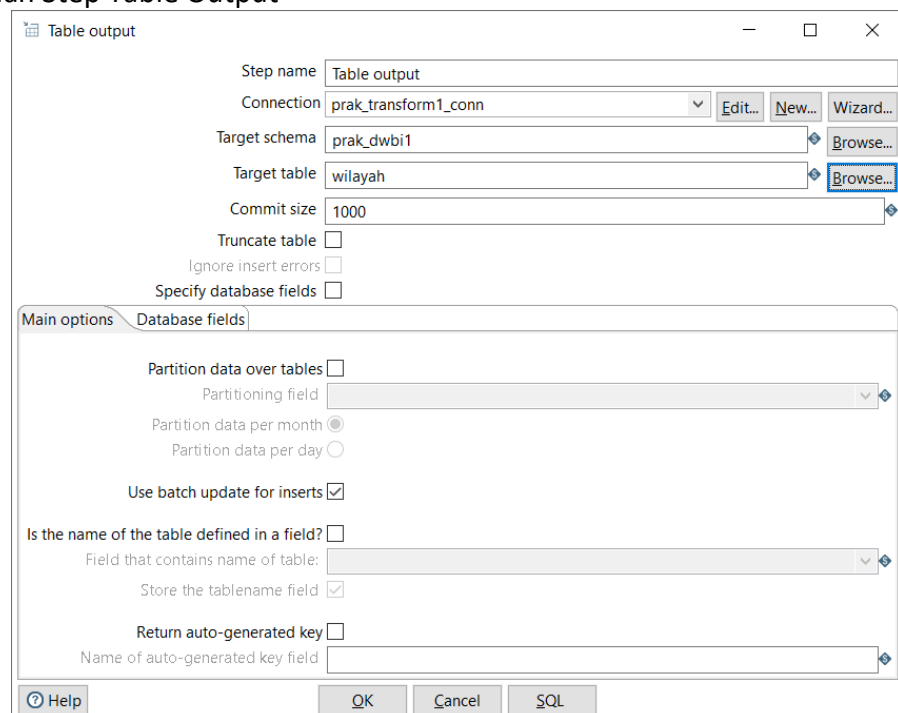
- Buat tabel baru pada database kita melalui Pentaho, Untuk melakukan operasi SQL, tekan tombol **SQL** pada bagian bawah dialog box.
 - Pada field **Target Table** isi dengan nama **wilayah**.
 - Setelah itu tekan tombol **SQL** yang berada dibagian bawah dialog box.



- Klik Tombol **Execute**

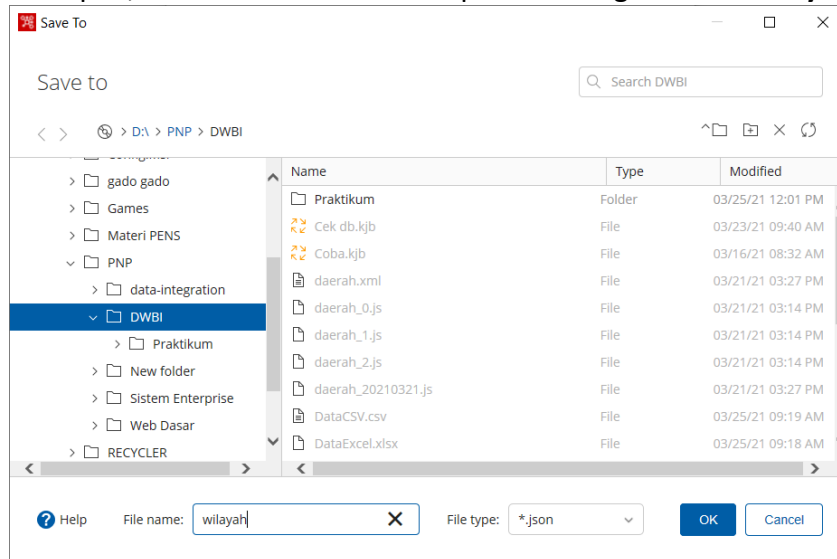


- Tampilan Step Table Output

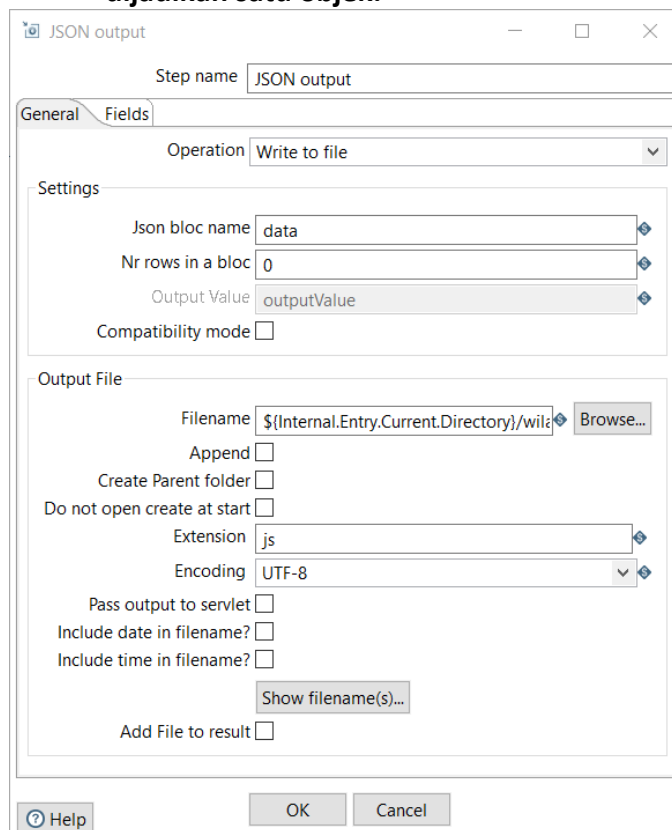


7. Pada Step JSON Output, penjelasan tentang step ini bisa dilihat di: <https://wiki.pentaho.com/display/EAI/JSON+output>

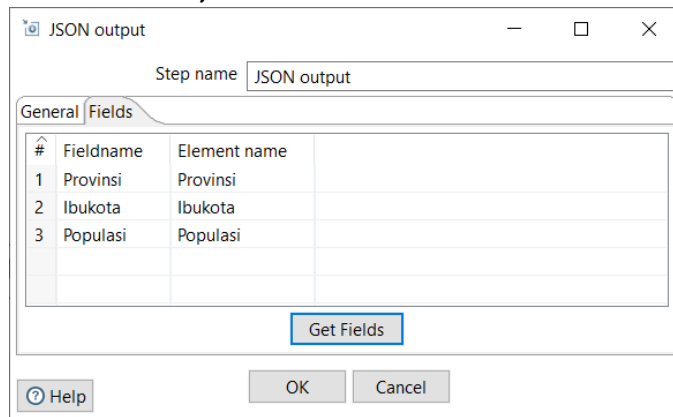
- Pada Tab **General**, pada field **Filename** tentukan nama dan lokasi file akan disimpan, Klik tombol **Browse**. Simpan file dengan nama **wilayah**.



- Tampilan dialog box JSON output.
 - Json bloc name : Nilai ini akan digunakan sebagai nama untuk blok json. Dapat berupa string kosong yang akan mempengaruhi struktur keluaran json.
 - Nr rows in a bloc: Jumlah kunci blok json - pasangan nilai. CATATAN, 1 adalah nilai khusus, dalam kasus nilai 1 setiap keluaran akan dihasilkan sebagai satu objek. **Pada praktikum ini diinputkan nilai 0, agar dianggap semua keluaran dijadikan satu objek.**

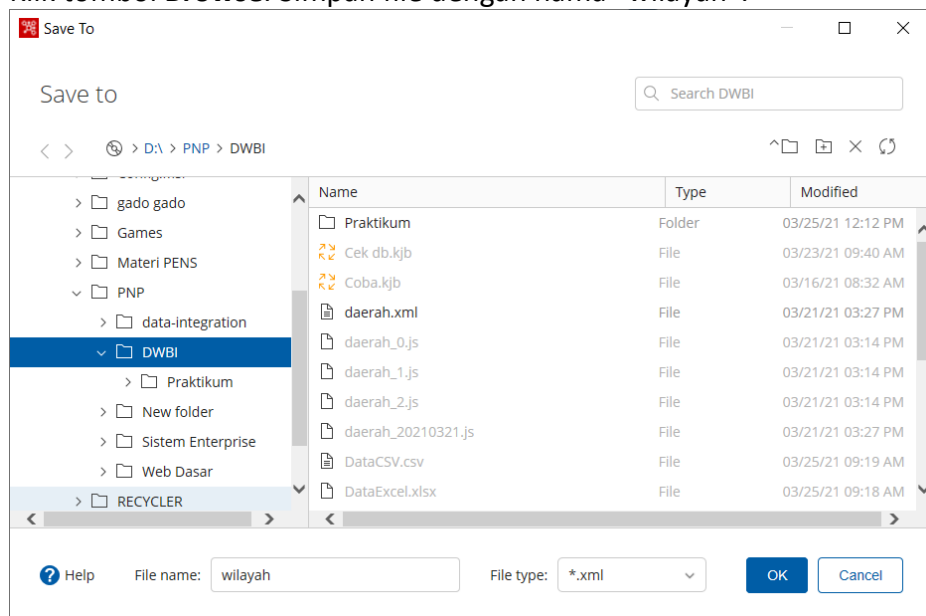


- Pada Tab **Field**, Klik tombol **Get Fields**

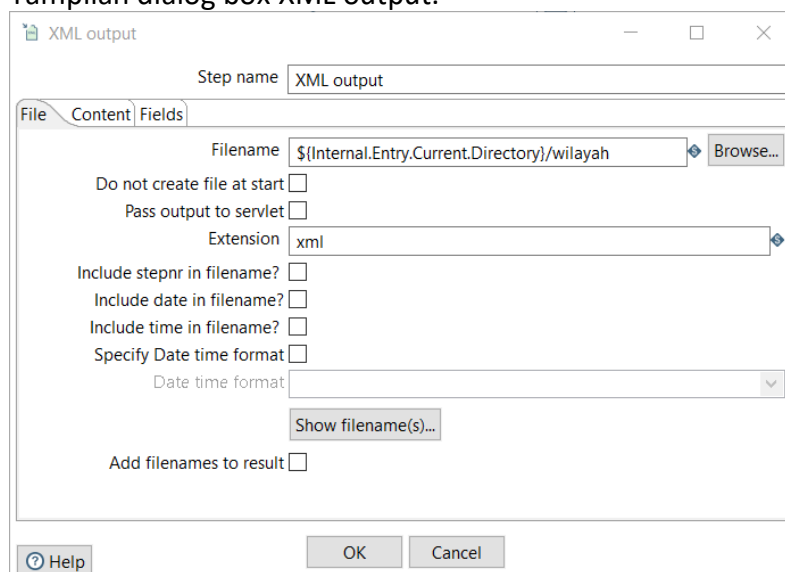


6. Pada Step XML Output, penjelasan tentang step ini bisa dilihat di: <https://wiki.pentaho.com/display/EAI/XML+Output>

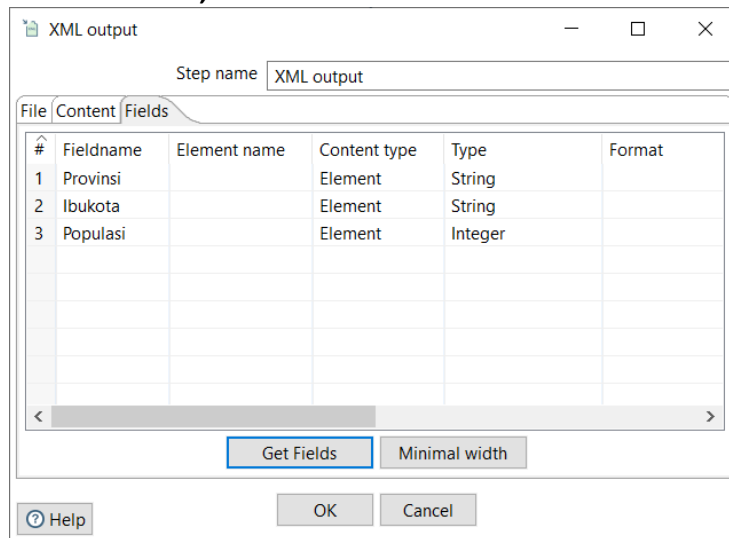
- Pada Tab **File**, pada field **Filename** tentukan nama dan lokasi file akan disimpan, Klik tombol **Browse**. Simpan file dengan nama “wilayah”.



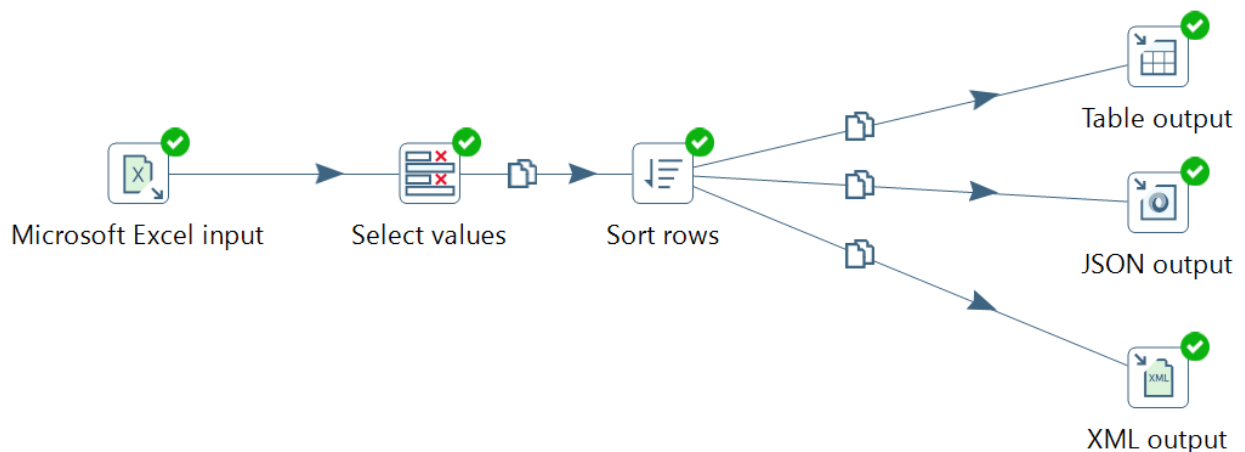
- Tampilan dialog box XML output.



- Pada Tab **Field**, Klik tombol **Get Fields**



7. Run Transformation.



- Pada Database MySQL kita, bisa di cek bahwa data telah masuk ke tabel **wilayah**.

Provinsi	Ibukota	Populasi
sulbar	Mamuju	114101
Gorontalo	Gorontalo	138354
Sulawesi Tenggara	Kendari	226056
Sulawesi Tengah	Palu	268644
sulut	Manado	417586
sulsel	Makasar	1168258

- Pada file JSON,

```

{"data": [
  {"Ibukota": "Mamuju", "Provinsi": "sulbar", "Populasi": 114101.0},
  {"Ibukota": "Gorontalo", "Provinsi": "Gorontalo", "Populasi": 138354.0},
  {"Ibukota": "Kendari", "Provinsi": "Sulawesi Tenggara", "Populasi": 226056.0},
  {"Ibukota": "Palu", "Provinsi": "Sulawesi Tengah", "Populasi": 268644.0},
  {"Ibukota": "Manado", "Provinsi": "sulut", "Populasi": 417586.0},
  {"Ibukota": "Makasar", "Provinsi": "sulsel", "Populasi": 1168258.0}
]}

```

- Pada file XML,

```
<?xml version="1.0" encoding="UTF-8"?>
- <Rows>
  - <Row>
    <Provinsi>sulbar</Provinsi>
    <Ibukota>Mamuju</Ibukota>
    <Populasi>114101.0</Populasi>
  </Row>
  - <Row>
    <Provinsi>Gorontalo</Provinsi>
    <Ibukota>Gorontalo</Ibukota>
    <Populasi>138354.0</Populasi>
  </Row>
  - <Row>
    <Provinsi>Sulawesi Tenggara</Provinsi>
    <Ibukota>Kendari</Ibukota>
    <Populasi>226056.0</Populasi>
  </Row>
  - <Row>
    <Provinsi>Sulawesi Tengah</Provinsi>
    <Ibukota>Palu</Ibukota>
    <Populasi>268644.0</Populasi>
  </Row>
  - <Row>
    <Provinsi>sulut</Provinsi>
    <Ibukota>Manado</Ibukota>
    <Populasi>417586.0</Populasi>
  </Row>
  - <Row>
    <Provinsi>sulsel</Provinsi>
    <Ibukota>Makasar</Ibukota>
    <Populasi>1168258.0</Populasi>
  </Row>
</Rows>
```

TUGAS 1

BAGAIMANA HASIL KELUARAN DATA SETELAH MELEWATI PROSES PADA STEP SORT ROWS?

- **JAWAB:**

TUGAS 2

PADA JSON OUTPUT UBAH NR ROWS IN BLOC MENJADI 1, BAGAIMANA HASIL YANG DIDAPTAKAN? DAN APA FUNGSI FIELD NR ROWS IN BLOC TERSEBUT?

- **HASIL:**
- **ANALISA:**

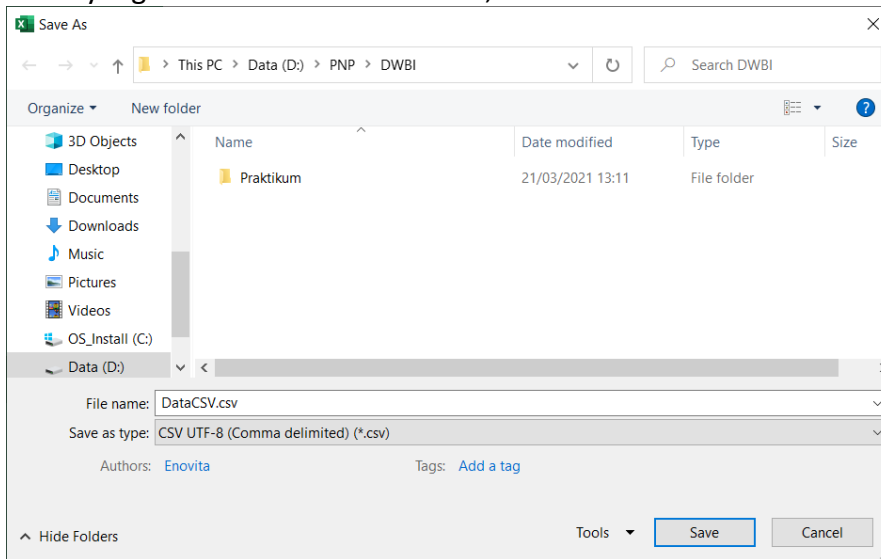
TUGAS 3

PADA PROSES MENGHUBUNGKAN STEP SORT ROWS DENGAN TABLE OUTPUT, JSON OUTPUT DAN XML OUTPUT, PILIH TOMBOL DISTRIBUTE, BAGAIMANA HASIL AKHIRNYA? HASIL TAMPILKAN ISI DARI SETIAP FILE!

- **HASIL:**
- **ANALISA:**

2. SKEMA PROSES ETL 2

1. Tabel yang tadi kita buat di Ms. Excel, di **Save As** ke dalam bentuk csv



• Hasil File CSV:

1	Provinsi, Ibukota, Populasi
2	sulut, Manado, 417586
3	Sulawesi Tengah, Palu, 268644
4	sulsel, Makassar, 1168258
5	Sulawesi Tenggara, Kendari, 226056
6	sulbar, Mamuju, 114101
7	Gorontalo, Gorontalo, 138354

(hasil bisa jadi menggunakan “,” koma sebagai pembatas antar data atau semicolon “;”)

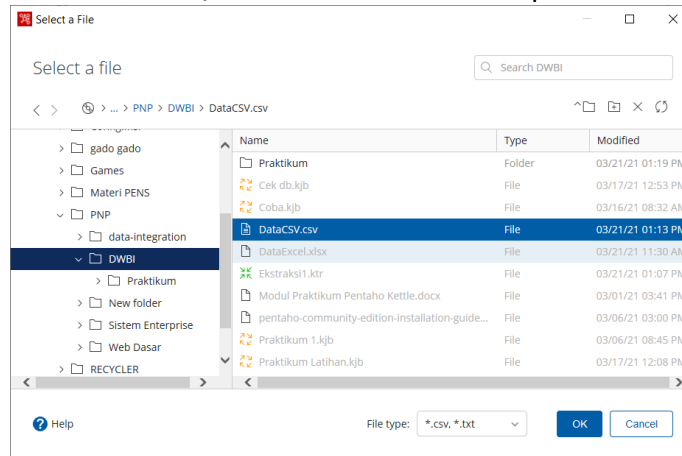
2. Buat desain transformasi baru seperti gambar dibawah ini:



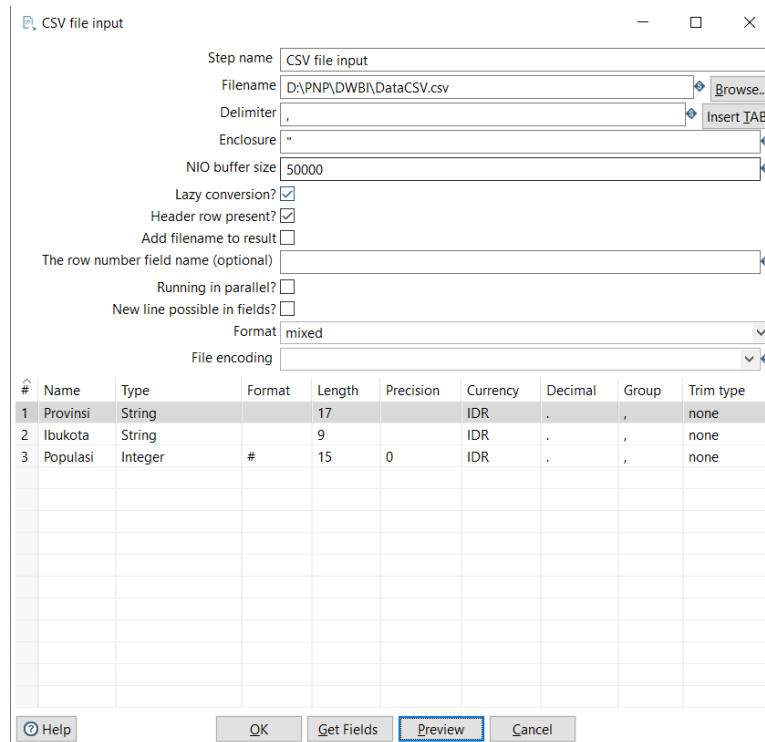
- Step CSV file Input berada di perluasan node **Input**
- Step Select values berada diperluasan node **Transform**
- Step Replace in String berada diperluasan node **Transform**
- Step Table output berada diperluasan node **Output**
- Saat menghubungkan tiap steps dengan hops pilih **Main output of step**.

3. Pada step **CSV file input**, penjelasan tentang step ini bisa dilihat di: [https://help.pentaho.com/Documentation/8.2/Products/Data Integration/Transformation Step Reference/CSV File Input](https://help.pentaho.com/Documentation/8.2/Products/Data%20Integration/Transformation%20Step%20Reference/CSV%20File%20Input)

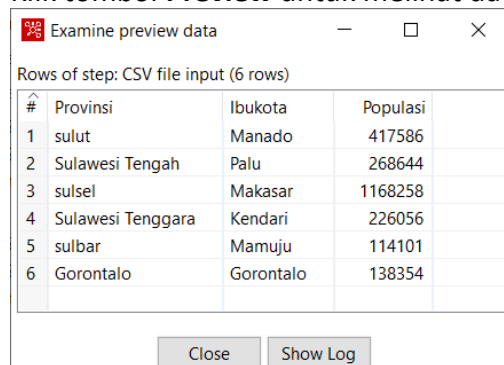
- Pada **Filename**, klik tombol Browse dan pilih file CSV yang ingin kita inputkan



- Pada **Delimiter**, isi dengan pemisah dari data kita bisa koma “,” atau semikolon “;”.
- Klik tombol **Get Fields** dan akan muncul header table kita tadi.



- Klik tombol **Preview** untuk melihat data dari csv yang akan kita ambil.



4. Pada step **Select values**,

- Pada Tab **Select & Alter**,
- Klik tombol **Get fields to select**.

Step name: Select values

Tab: Select & Alter | Remove | Meta-data

Fields:

#	Fieldname	Rename to	Length	Precision
1	Provinsi			
2	Ibukota			
3	Populasi			

Buttons: Get fields to select, Edit Mapping

Include unspecified fields: ☐

Buttons: Help, OK, Cancel

- Pilih Fieldname kolom yang mana datanya akan dipakai untuk step berikutnya.

5. Pada step **Replace in String**, penjelasan tentang step ini bisa dilihat di:

[https://help.pentaho.com/Documentation/8.2/Products/Data Integration/Transformation on Step Reference/Replace In String](https://help.pentaho.com/Documentation/8.2/Products/Data%20Integration/Transformation%20Step%20Reference/Replace%20In%20String)

Step name: Replace in string

Fields string:

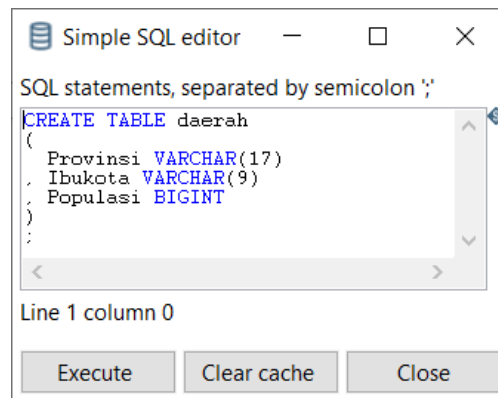
#	In stream field	Out stream field	use RegEx	Search	Replace with	Set empty string?	Replace with field	Whole Word	Case sensitive	Is Unicode
1	Provinsi		N	sulut	Sulawesi Utara	N		N	N	N
2	Provinsi		N	sulsel	Sulawesi Selatan	N		N	N	N
3	Provinsi		N	sulbar	Sulawesi Barat	N		N	N	N

Buttons: Help, OK, Get fields, Cancel

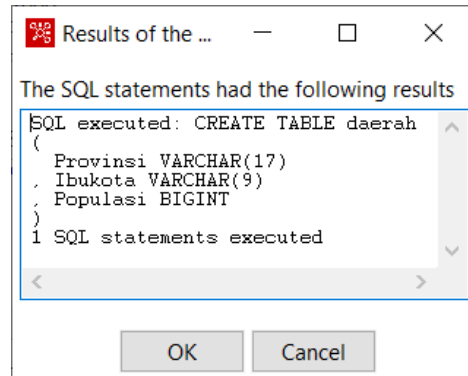
Kita mengubah data pada kolom provinsi dimana hanya singkatan diubah menjadi nama panjang.

6. Pada step **Table Output**,

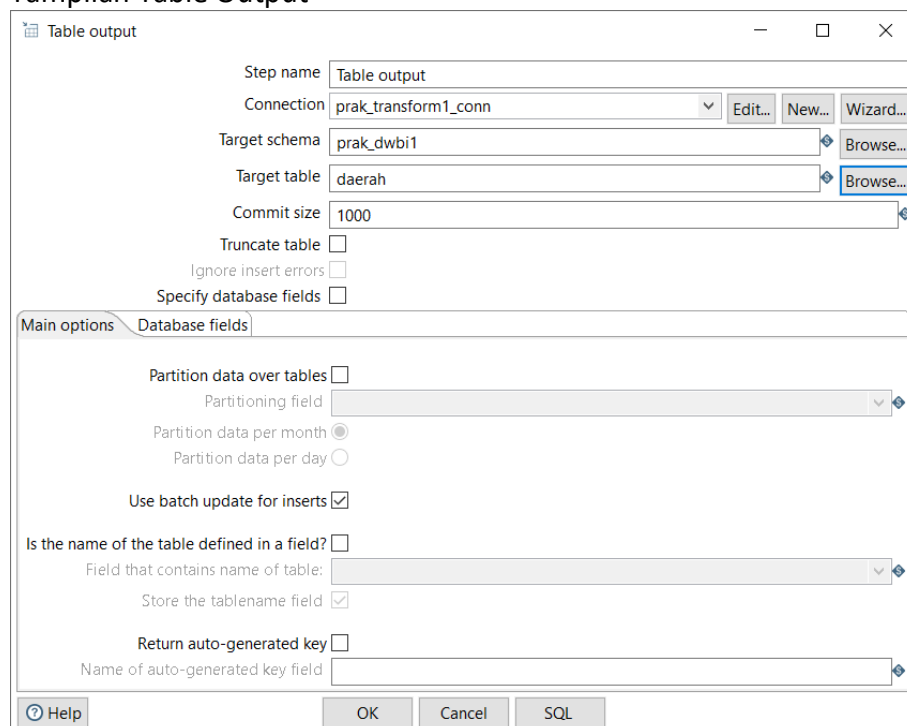
- Pada Field Connection, pilih Koneksi database yang sudah kita buat.
- Buat tabel baru pada database kita melalui Pentaho, Untuk melakukan operasi SQL, tekan tombol **SQL** pada bagian bawah dialog box.
 - Pada field **Target Table** isi dengan nama **daerah**.
 - Setelah itu tekan tombol **SQL** yang berada dibagian bawah dialog box.



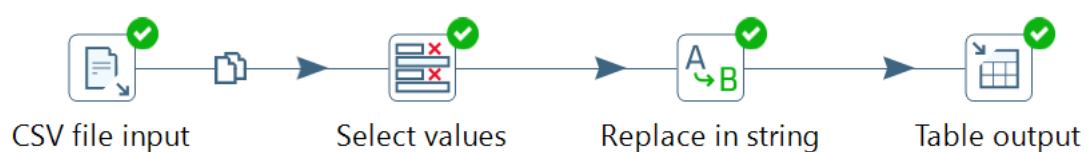
➤ Klik Tombol **Execute**



• Tampilan Table Output



7. Run Transformation

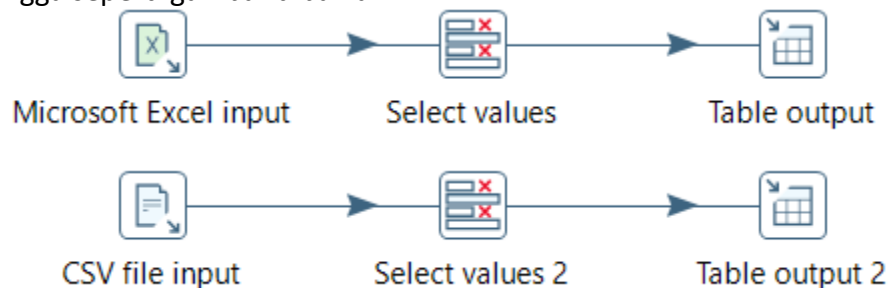


Pada Database MySQL kita, bisa di cek bahwa data telah masuk ke tabel dengan nama daerah.

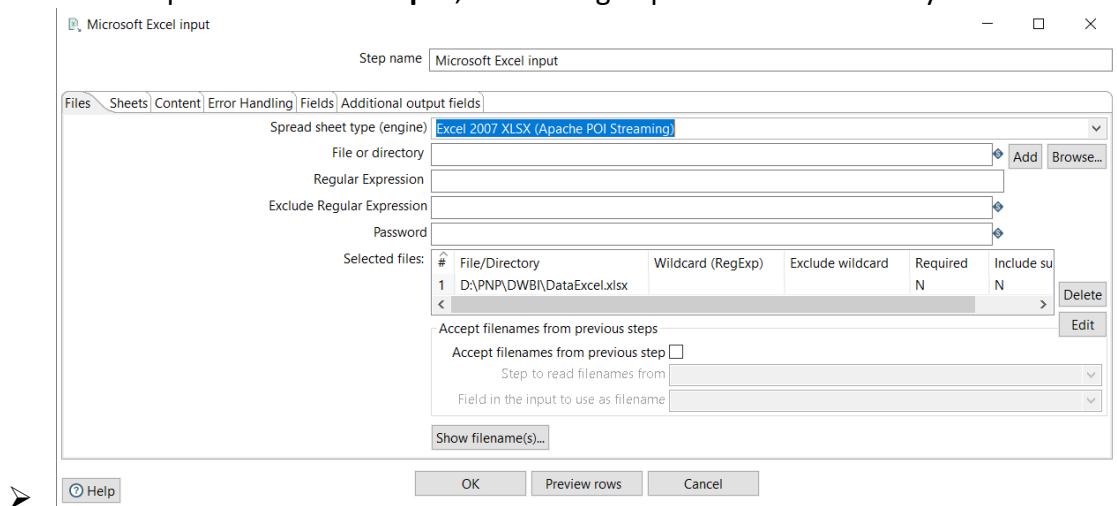
Provinsi	Ibukota	Populasi
Sulawesi Utara	Manado	417586
Sulawesi Tengah	Palu	268644
Sulawesi Selatan	Makasar	1168258
Sulawesi Tenggara	Kendari	226056
Sulawesi Barat	Mamuju	114101
Gorontalo	Gorontalo	138354

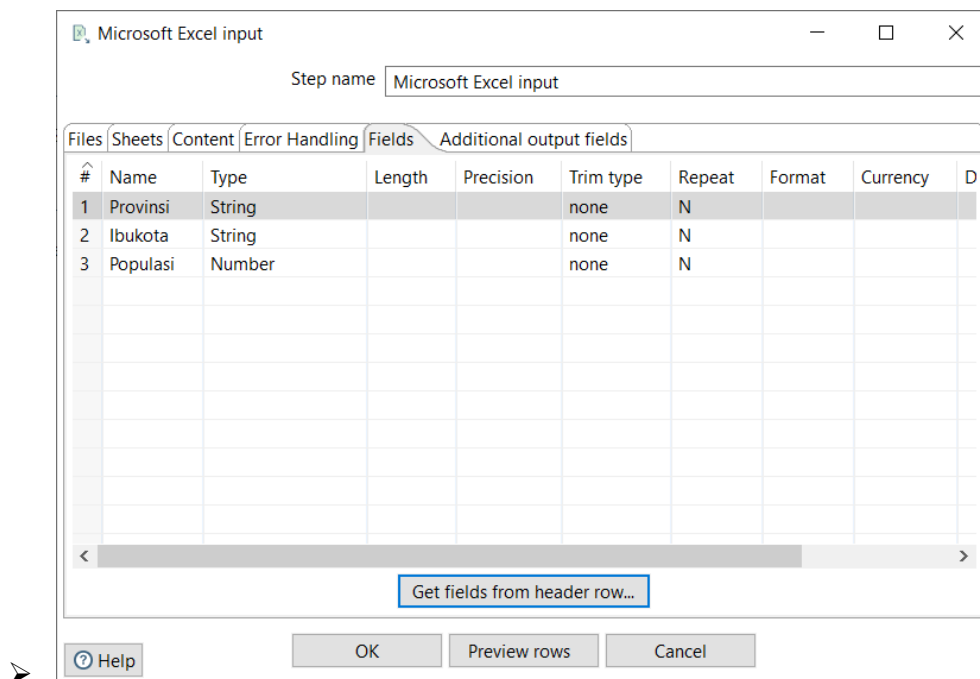
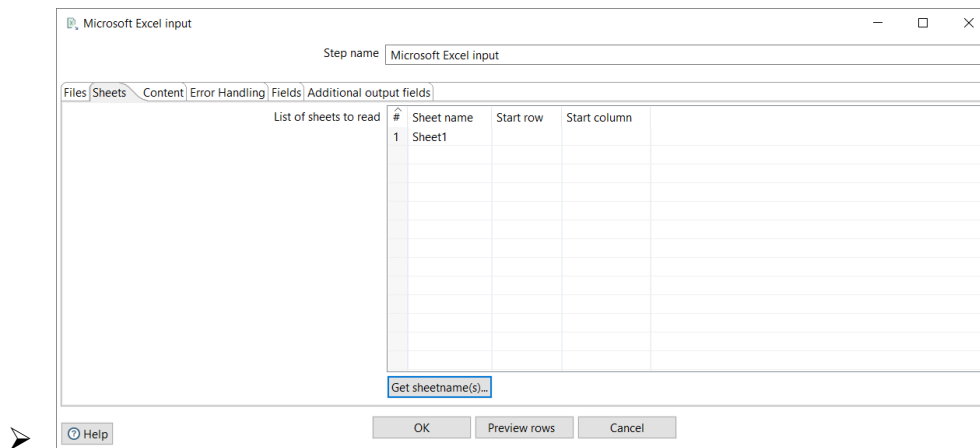
3. SKEMA PROSES ETL 3

1. Buat Desain Transformasi dengan menggabungkan kedua percobaan yang kita lakukan tadi sehingga seperti gambar dibawah ini:

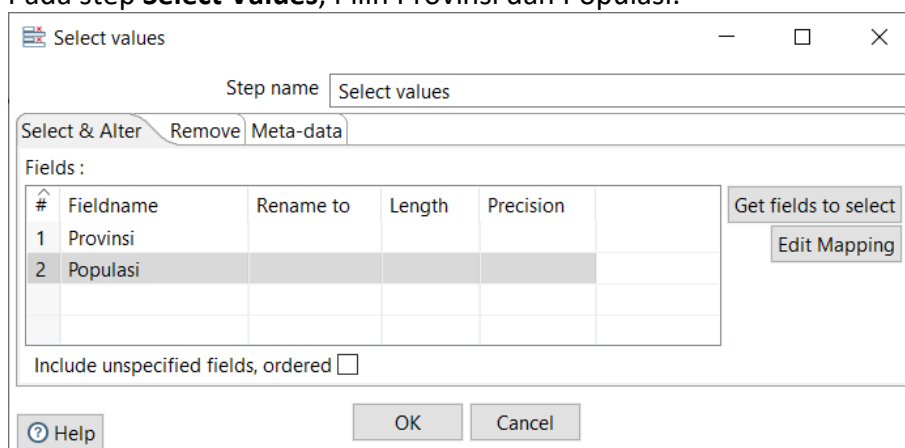


- Pada step **Microsoft Excel input**, sama dengan percobaan sebelumnya.





- Pada step **Select Values**, Pilih Provinsi dan Populasi.



- Pada step **Table Output**, Target Table adalah **buat tabel baru dengan nama “gabung” pada koneksi yang sudah kita buat.**

Table output

Step name: Table output

Connection: prak_transform1_conn [Edit... New... Wizard...]

Target schema: prak_dwbi1 [Browse...]

Target table: gabungan [Browse...]

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☐

Main options Database fields

Partition data over tables: ☐

Partitioning field: []

Partition data per month: ☒

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

Field that contains name of table: []

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key field: []

[?] Help [OK] [Cancel] [SQL]

Simple SQL editor

SQL statements, separated by semicolon ';' :

```
CREATE TABLE gabungan
(
  Provinsi TINYTEXT
,
  Populasi DOUBLE
)
;
```

Line 1 column 0

[Execute] [Clear cache] [Close]

Results of the ...

The SQL statements had the following results

```
SQL executed: CREATE TABLE gabungan
(
  Provinsi TINYTEXT
,
  Populasi DOUBLE
)
1 SQL statements executed
```

[OK] [Cancel]

- Pada step **CSV file input**, sama dengan percobaan sebelumnya.

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Provinsi	String		17		IDR	.	,	none
2	Ibukota	String		9		IDR	.	,	none
3	Populasi	Integer	#	15	0	IDR	.	,	none

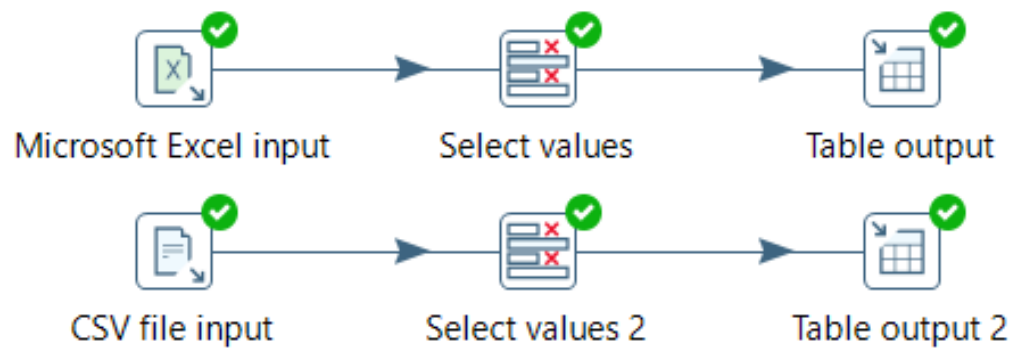
- Pada step **Select values 2**, pilih Pilih Provinsi dan Populasi.

#	Fieldname	Rename to	Length	Precision
1	Provinsi			
2	Populasi			

- Pada step **Table Output 2**,

➤ Pada field Target Table, Klik Tombol browse dan cari tabel **gabung**.

2. Run Transformasi



Pada Database MySQL kita, bisa di cek bahwa ada dua *batch* data telah masuk ke tabel **gabung**.

Provinsi	Populasi
sulut	417586
Sulawesi Tengah	268644
sulsel	1168258
sulut	417586
Sulawesi Tenggara	226056
Sulawesi Tengah	268644
sulsel	1168258
Sulawesi Tenggara	226056
sulbar	114101
Gorontalo	138354
sulbar	114101
Gorontalo	138354

4. DATABASE LOOKUP

1. Buat File CSV, dengan nama "Provinsi Di Pulau Sumatera.csv"

```

1 "Provinsi","Ibu Kota","Populasi"
2 "Aceh","Banda Aceh",223446
3 "Sumatera Utara","Medan",2097610
4 "Sumatera Barat","Padang",833562
5 "Riau","Pekanbaru",897767
6 "Jambi","Jambi",531857
7 "Sumatera Selatan","Palembang",1455284
8 "Bengkulu","Bengkulu",308544
9 "Lampung","Bandar Lampung",881801
0 "Kepulauan Bangka Belitung","Pangkal Pinang",174000
1 "Kepulauan Riau","Tanjung Pinang",192493

```

2. Buat tabel baru pada database anda

a. Tabel lookup_provinsi

Server: 127.0.0.1 » Database: prak_dwbt1

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer

Table name: lookup_provinsi Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_	Comments	MIME t
kode_provinsi	INT	10	None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>		
nama_provinsi	VARCHAR	256	None			<input type="checkbox"/>	---	<input type="checkbox"/>		
	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>		
	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>		

Table comments:

Storage Engine: InnoDB Collation:

PARTITION definition:

Save

kode_provinsi	nama_provinsi
1	Jawa Barat
2	Jawa Tengah
3	Sumatera Barat
4	Sumatera Selatan
5	Sumatera Utara
6	Riau
7	Jambi
8	Lampung

b. Table alamat

Server: 127.0.0.1 » Database: prak_dwbt1 » Table: alamat

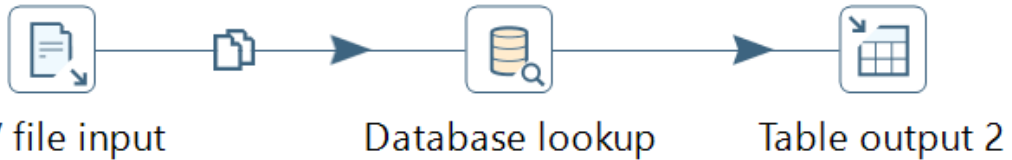
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Name	Type	Length/Values	Default	Collation	Attributes	Null	A_	Comments	Move column	MIME
provinsi	VARCHAR	25	None	utf8mb4_general_		<input type="checkbox"/>	<input type="checkbox"/>			
populasi	INT	25	NULL			<input checked="" type="checkbox"/>	<input type="checkbox"/>			
ibukota	VARCHAR	25	NULL	latin1_swedish_ci		<input checked="" type="checkbox"/>	<input type="checkbox"/>			
kode_provinsi	INT	11	NULL			<input checked="" type="checkbox"/>	<input type="checkbox"/>			

Save

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	provinsi	varchar(25)	utf8mb4_general_ci		No	None	
<input type="checkbox"/> 2	populasi	int(25)			Yes	NULL	
<input type="checkbox"/> 3	ibukota	varchar(25)	latin1_swedish_ci		Yes	NULL	
<input type="checkbox"/> 4	kode_provinsi	int(11)			Yes	NULL	

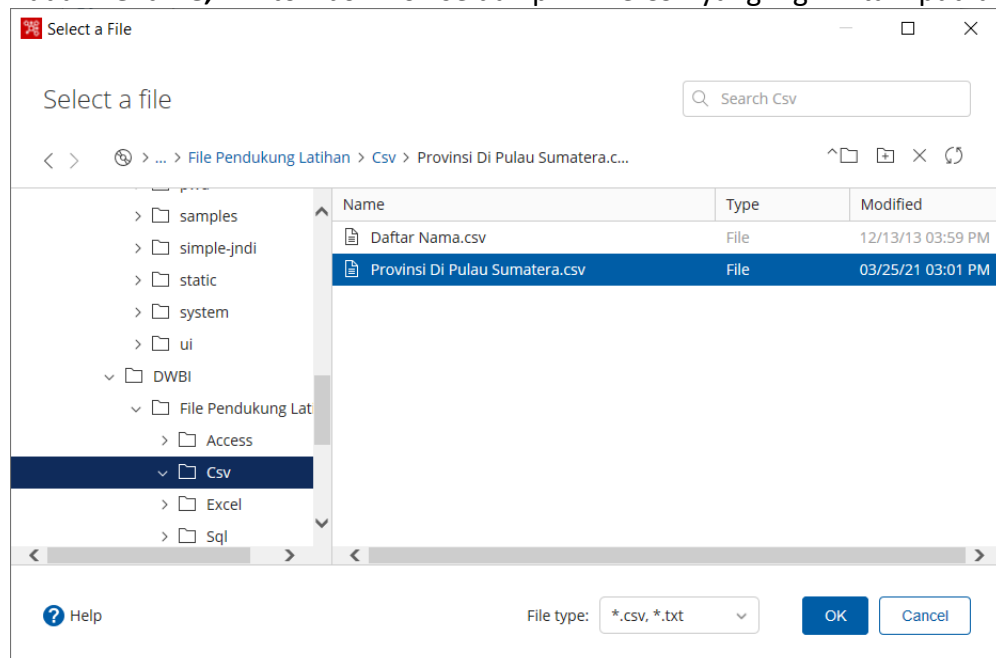
3. Buat desain transformation seperti gambar dibawah ini:



- Step **Database lookup** terletak di perluasan node Lookup.

4. CSV FILE INPUT

- Pada **Filename**, klik tombol Browse dan pilih file CSV yang ingin kita inputkan



- Pada **Delimiter**, isi dengan pemisah dari data kita bisa koma “,” atau semikolon “;”.
- Klik tombol **Get Fields** dan akan muncul header table kita tadi.

CSV file input

Step name: CSV file input

Filename: \${Internal.Entry.Current.Directory}/File Pendukung Latihan/Csv/Provi... Browse...

Delimiter: , Insert TAB

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Provinsi	String		25		IDR	.	,	none
2	Ibu Kota	String		14		IDR	.	,	none
3	Populasi	Integer	#	15	0	IDR	.	,	none

Help OK Get Fields Preview Cancel

- Klik tombol **Preview** untuk melihat data dari csv yang akan kita ambil.

Examine preview data

Rows of step: CSV file input (10 rows)

#	Provinsi	Ibu Kota	Populasi
1	Aceh	Banda Aceh	223446
2	Sumatera Utara	Medan	2097610
3	Sumatera Barat	Padang	833562
4	Riau	Pekanbaru	897767
5	Jambi	Jambi	531857
6	Sumatera Selatan	Palembang	1455284
7	Bengkulu	Bengkulu	308544
8	Lampung	Bandar Lampung	881801
9	Kepulauan Bangka Belitung	Pangkal Pinang	174000
10	Kepulauan Riau	Tanjung Pinang	192493

Close Show Log

5. DATABASE LOOKUP, penjelasan tentang step ini bisa dilihat di:

<https://wiki.pentaho.com/display/EAI/Database+lookup>

- a. Buat koneksi baru atau bisa gunakan koneksi database yang lama.
- b. Setelah itu pilih lookup tablenya output pada 'lookup table' yaitu **lookup_provinsi**.
- c. Klik "Get lookup field".
- d. Pilih field yang akan di lookup (Pilih secara dropdown tidak diketik).
 - Pada The key(s) to look up the value(s): pada kolom Table field pilih nama_provinsi pada tabel lookup_provinsi gunakan comparator '=' dan kolom Field 1 isi dengan Provinsi pada file csv.
 - Pada values to return from the lookup table: pada kolom field pilih kode_provinsi.

Database lookup

Step name: Database lookup

Connection: prak_transform1_conn

Lookup schema: prak_dwbi1

Lookup table: lookup_provinsi

Enable cache? ☐

Cache size in rows (0=cache): 0

Load all data from table ☐

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	nama_provinsi	=	Provinsi	

Values to return from the lookup table:

#	Field	New name	Default	Type
1	kode_provinsi			Integer

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

Buttons: Help, OK, Cancel, Get Fields, Get lookup fields

6. TABLE OUTPUT

- Pada Tab Main Option
 - a. Pilih koneksi yang sama yang telah dibuat sebelumnya.
 - b. Pilih target tabel lookupnya.
 - c. Centang pada specify database field.

Specify database fields ☒

Table output

Step name: Table output 2

Connection: prak_transform1_conn

Target schema: prak_dwbi1

Target table: alamat

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options

Database fields

Partition data over tables: ☐

Partitioning field:

Partition data per month: ☒

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

Field that contains name of table:

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key field:

Help OK Cancel SQL

- Pada Tab database field,
 - a. klik “enter field mapping”.

Enter field mapping

- b. Pilih kode_provinsi, Provinsi, Populasi, dan Ibu Kota.

Enter Mapping

Source fields:

Target fields:

Mappings:

kode_provinsi --> kode_provinsi

Provinsi --> provinsi

Populasi --> populasi

Ibu Kota --> ibukota

Add

Delete

Auto target selection? ☒

Auto source selection? ☐

Hide assigned source fields? ☒

Hide assigned target fields? ☒

OK Guess Cancel

7. Run Transformasi



CSV file input

Database lookup

Table output 2

Hasil tabel alamat

provinsi	populasi	ibukota	kode_provinsi
Aceh	223446	Banda Aceh	NULL
Sumatera Utara	2097610	Medan	5
Sumatera Barat	833562	Padang	3
Riau	897767	Pekanbaru	6
Jambi	531857	Jambi	7
Sumatera Selatan	1455284	Palembang	4
Bengkulu	308544	Bengkulu	NULL
Lampung	881801	Bandar Lampung	8
Kepulauan Bangka Belitung	174000	Pangkal Pinang	NULL
Kepulauan Riau	192493	Tanjung Pinang	NULL

TUGAS 4: Jelaskan apa guna dari database lookup pada percobaan ini!

5. *Tugas*

1. Lakukan semua percobaan yang telah dijelaskan pada modul diatas dan buat laporannya.
2. Dalam laporan sesuaikan tampilan screenshot dari hasil praktikum.
3. Kerjakan perintah yang di tugaskan **(tulisan tugas yang di cetak tebal)**.
4. Berikan analisa dan kesimpulan setiap percobaan yang telah dilakukan.
5. **Tambahkan hasil percobaan dari praktikum yang dikerjakan langsung atau yang ditambahkan saat tatap muka.**

FORMAT LAPORAN, terdiri dari:

- A. Cover
- B. Dasar Teori
- C. Langkah dan Hasil Percobaan
- D. Analisa dan Kesimpulan Percobaan
- E. Referensi