

BAB IV HASIL DAN PEMBAHASAN

4.1 Tools Kebutuhan Sistem

Adapun aplikasi yang digunakan dalam pembuatan *warehouse* ini adalah sebagai berikut:

- a. Pentaho 9.1
- b. JRE 8 (*Java Run Time Environment*)
- c. MySQL J-connector
- d. Xampp

4.2 Hasil

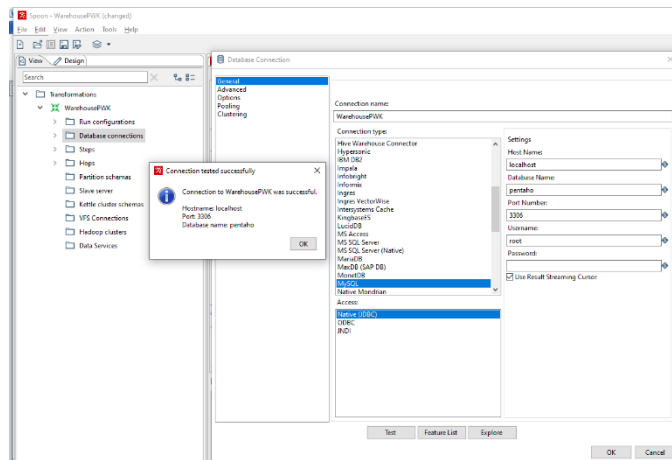
Penelitian yang dilakukan mencakup kasus kejahatan di Polres Way Kanan menghasilkan suatu sistem pengelolaan *warehouse* data kasus kejahatan Bagian Operasional Polres Way Kanan. Data kasus tersebut didapat dari beberapa sektor wilayah naungan Polres Way Kanan yaitu Polres, Umpu, Datu, Kasui, RBT, Banjit, W Tuba, Bahuga, G Labuhan, P Ratu, Neg Batin, dan Neg Besar. Informasi data yang disajikan dalam *warehouse* tersebut dapat dimanfaatkan Bag. Ops dalam mendukung pengambilan keputusan.

4.3 Pembahasan dan Simulasi

Adapun pembahasan dan simulasi tahapan pembuatan *warehouse* kasus kejahatan Bag. Ops Polres Way Kanan yang dimulai dari tahap pembuatan ETL sampai koneksi *database* dijelaskan pada sub pokok bahasan di bawah ini.

4.3.1 Membuat *Connection Database*

Sebelum membuat proses ETL, langkah pertama yang harus dilakukan adalah membuat koneksi ke *database* dengan menggunakan aplikasi xampp dengan cara membuka **localhost/phpmyadmin** kemudian buat *database* dengan nama pentaho (nama sesuai selera). Selanjutnya pada pentaho klik ***view-double klik pada database connection***, akan tampil jendela seperti pada Gambar 4.1.



Gambar 4.1 Tampilan Jendela *Database Connection*

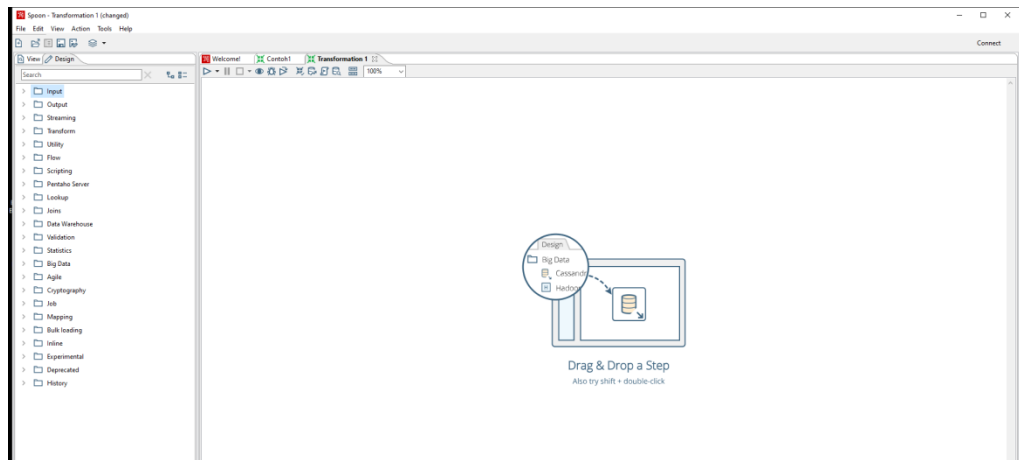
Kemudian isi **connection name** dengan nama warehousePWK (nama sesuai selera). **Host name** diisi dengan *localhost*, **database name** diisi dengan nama *database* yang telah dibuat di xampp. Pada **connection type** pilih MySQL, pada **port number** diisi dengan 3306, **username** diisi dengan *root* (tidak perlu diisi), **access** pada *native (JDBC)* kemudian tekan **test**. Jika muncul pemberitahuan sukses, maka klik **OK**.

4.3.2 Proses ETL

Pada perancangan data *warehouse* ini, data *input* yang digunakan berformat **.xlsx** (*input data excel*). Aplikasi yang digunakan dalam pembuatan ETL adalah Pentaho 9.1. Adapun tahapan dari proses ETL adalah sebagai berikut :

a. Membuka Aplikasi Pentaho

Hal pertama yang dilakukan adalah membuka aplikasi pentaho dengan cara buka *folder* tempat aplikasi pentaho disimpan, kemudian **double klik spoon.bat**. Sistem menampilkan jendela pentaho seperti pada Gambar 4.2.



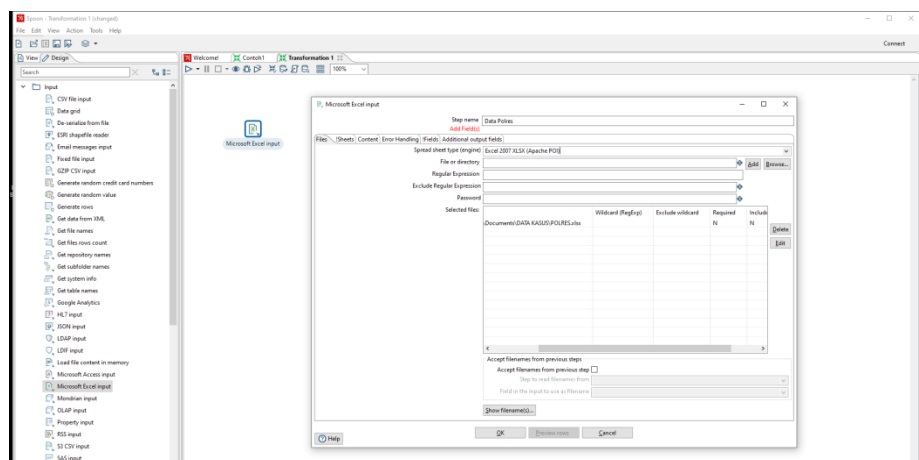
Gambar 4.2 Tampilan Jendela Aplikasi Pentaho

b. Membuat *File Input Data*

Setelah jendela aplikasi pentaho terbuka, maka langkah selanjutnya adalah membuat *input* data. Data yang digunakan dalam proses ini adalah data kasus tiap wilayah naungan Polres Way Kanan tahun 2020. Adapun tahapan dalam mengelola data *input* adalah sebagai berikut :

1. Membuka Jendela *Microsoft Excel Input*

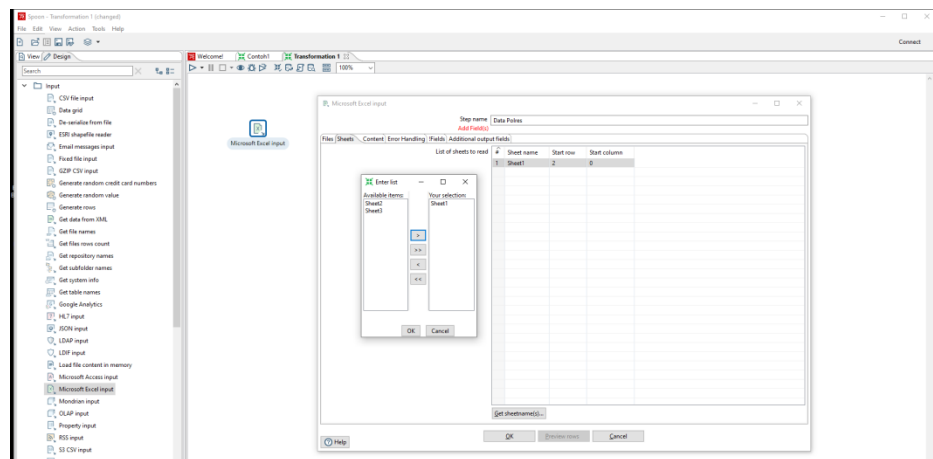
Setelah jendela aplikasi pentaho terbuka, klik ***transformation-input-Microsoft Excel Input*** kemudian tekan dan geret ke dalam jendela utama. Setelah itu, ***double*** klik pada ***Microsoft Excel Input*** di dalam jendela utama dan sistem menampilkan jendela ***Microsoft Excel Input*** seperti pada Gambar 4.3.



Gambar 4.3 Tampilan Jendela *Microsoft Excel Input*

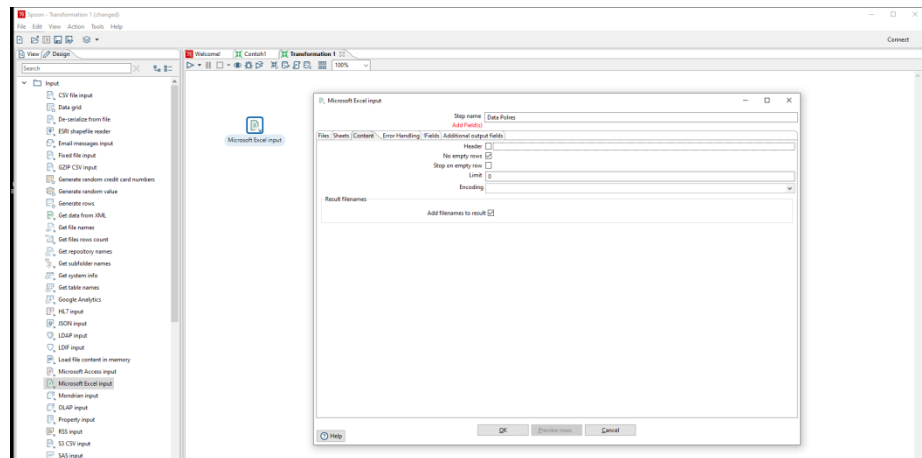
Step name pada kolom **files** diisi dengan nama data polres. Kemudian pada **spread sheet type (engine)** isi dengan **excel 2007xlsx apache POI**. **Browse file or directory** (isi dengan data kasus polres .xlsx / nama **file input**), kemudian **add** dan data **input** akan tercantum dalam kolom **selected files**.

Pada kolom **sheets**, klik **get_sheetname(s)**, kemudian pilih **sheet** yang akan diambil datanya (pada kasus ini sheet1) setelah itu tekan **OK**. **Sheet** yang dipilih akan tertera pada **sheet name**. Isi **start row** dengan angka 2 dan **start column** dengan angka 0. **Start row** disini menunjukkan baris ke berapa data tersebut diambil, sedangkan start column menunjukkan batasan kolom yang akan diambil. Pada kasus ini, data diambil (dari data **excel**) pada baris ke 2 dan kolom 0 seperti pada Gambar 4.4.



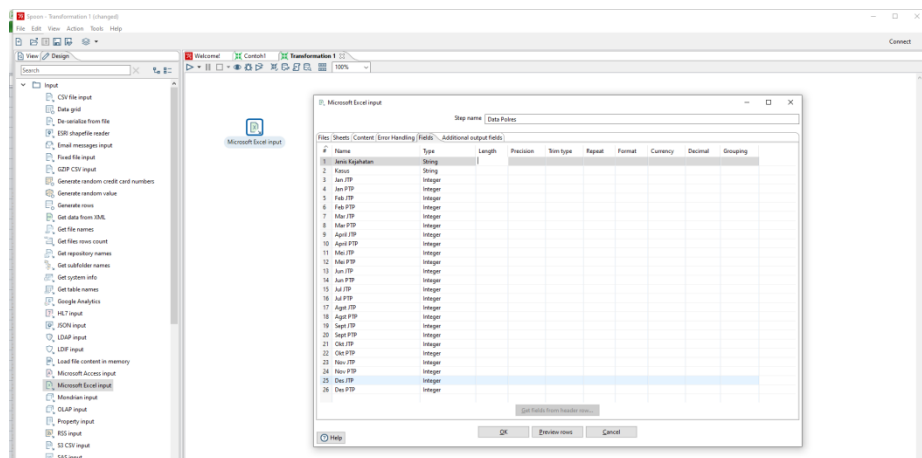
Gambar 4.4 Tampilan Jendela **Sheets** pada **Microsoft Excel Input**

Pada kolom **content**, centang pada **header** dihilangkan agar data pada **row** pertama tidak dijadikan **field** (ingin mendefinisikan **field** secara manual) seperti pada Gambar 4.5.



Gambar 4.5 Tampilan jendela *Content* pada *Microsoft Excel Input*

Setelah itu, pada kolom **fields**, **name** diisi dengan *field* berdasarkan data yang ada di dalam *file input*, dan **type** tiap *field* disesuaikan seperti pada Gambar 4.6.



Gambar 4.6 Tampilan jendela *Fields* pada *Microsoft Excel Input*

Setelah semua tahap tersebut terisi, langkah selanjutnya adalah melihat data yang akan dikelola apakah sesuai atau tidak dengan menekan tombol **preview rows** sehingga data akan tampil seperti pada Gambar 4.7.

Row	Name	Jan (JP)	Feb (TP)	Mar (JP)	Apr (JP)	Mei (JP)	Jun (JP)	Jul (JP)	Sept (JP)
1	KEHAKIMAN KONSERNSIONAL	4	2	1	1	1	2	2	2
2	KEHAKIMAN KONSERNSIONAL	2	0	1	1	1	1	1	1
3	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	2	2	1
4	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
5	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
6	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
7	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
8	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
9	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
10	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
11	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
12	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
13	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
14	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
15	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
16	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
17	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
18	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1
19	KEHAKIMAN KONSERNSIONAL	1	1	1	1	1	1	1	1

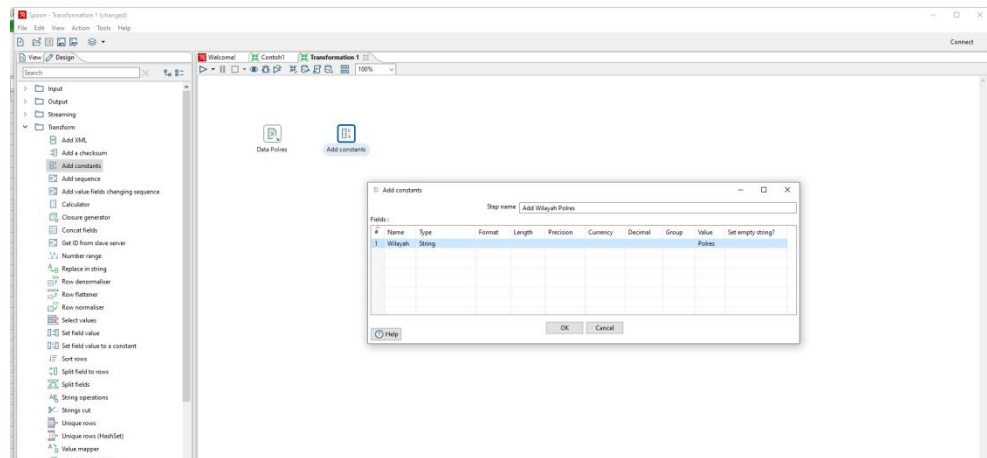
Gambar 4.7 Tampilan Jendela *Preview Rows* pada *Microsoft Excel Input*

Setelah data sesuai maka tekan **OK**. Untuk data kasus Umpu, Datu, Kasui, RBT, Banjit, W Tuba, Bahuga, G Labuhan, P Ratu, Neg Batin, dan Neg Besar lakukan tahap yang sama seperti tahapan *input* data pada kasus Polres.

2. Membuka Jendela *Add Constants* (Menambahkan Wilayah/Sektor)

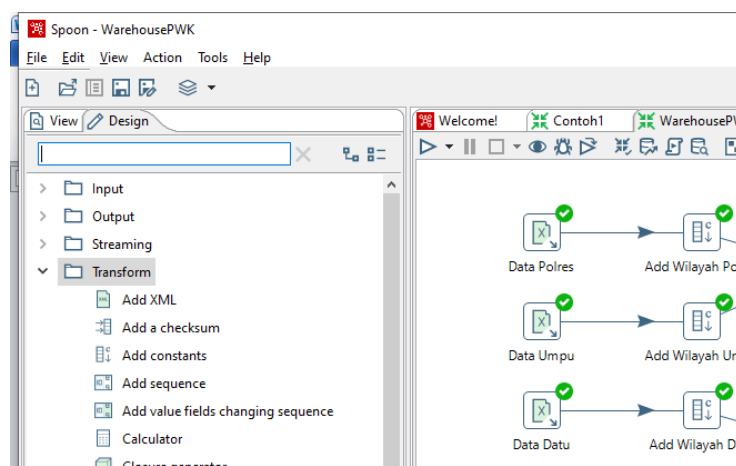
Setelah data kasus sudah terbaca pada tahap *Microsoft Excel Input*, langkah selanjutnya adalah menambahkan data wilayah/sektor pada kolom *input* data dikarenakan di dalam *file* data kasus tidak mencantumkan data wilayah/sektor kasus tersebut. Caranya adalah klik **transform-add constants- drag ke dalam jendela utama**.

Step name diisi dengan nama wilayah kasus. Penamaan disini dapat diisi dengan nama apapun. Kemudian pada kolom **name** diisi dengan wilayah karena menambahkan *field* wilayah ke dalam data kasus. **Type** data diisi dengan **string**. Value diisi dengan polres sesuai dengan data kasus pada *Microsoft Excel Input*, kemudian **OK** seperti pada Gambar 4.8. Untuk data *add* wilayah Umpu, Datu, Kasui, RBT, Banjit, W Tuba, Bahuga, G Labuhan, P Ratu, Neg Batin, dan Neg Besar lakukan tahap yang sama seperti tahapan pada kasus Polres.



Gambar 4.8 Tampilan Jendela *Add Constants* (Menambahkan Wilayah/Sektor)

Setelah data berhasil ditambahkan, maka pada data polres pada *Microsoft Excel Input* ditarik panah ke **add wilayah polres** yang berfungsi untuk menghubungkan data seperti pada Gambar 4.9.



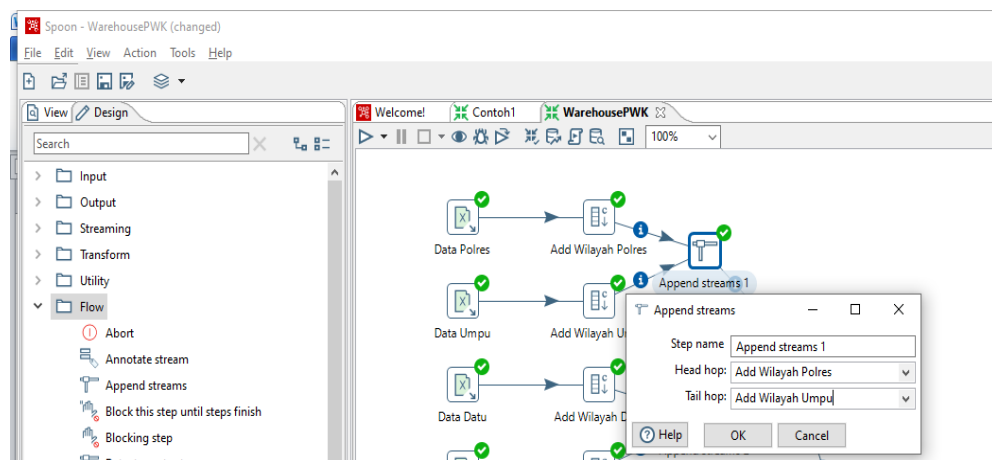
Gambar 4.9 Jendela Hubungan Data Polres dengan Add Wilayah Polres

3. Membuka Jendela *Append Stream*

Langkah selanjutnya adalah membuka jendela *append stream* dengan cara klik **flow-append stream-drag** ke layar utama. Fungsi *append stream* disini menggabungkan data satu dengan yang lainnya dengan syarat *append*

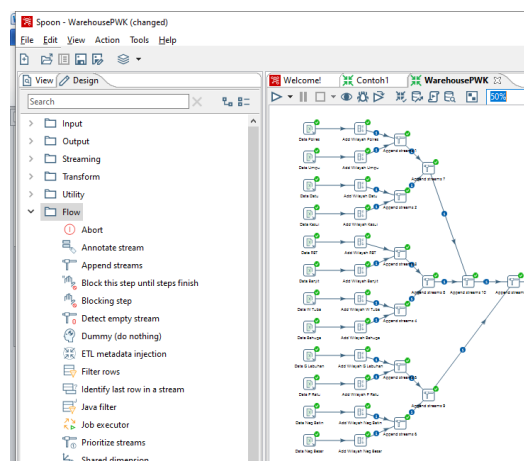
stream hanya dapat menggabungkan dua data menjadi satu. Oleh karena itu, dibutuhkan lebih dari satu *append stream* dalam menggabungkan data.

Double klik *append stream* yang ada pada jendela utama, kemudian isi *step name* dengan nama *append streams 1*. Pada *head hop* diisi dengan *add wilayah polres*, sedangkan pada *tail hop* diisi dengan *add wilayah umpu* kemudian **OK** seperti pada Gambar 4.10.



Gambar 4.10 Tampilan Jendela *Append Stream*

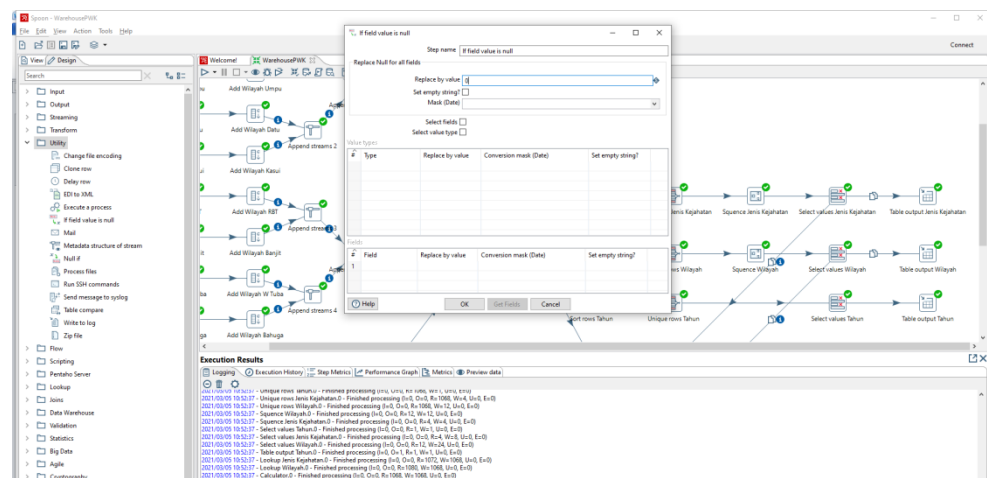
Lakukan tahap yang sama pada *add wilayah* yang belum digabung hingga penggabungan terkumpul pada satu *append stream* seperti pada Gambar 4.11.



Gambar 4.11 Tampilan Jendela Hasil Penggabungan *Append Stream*

4. Menambahkan Utility If Field Value is Null

Jendela ini digunakan untuk merubah data “*null*” menjadi nilai “0”. Tahap ini diperlukan agar jika terjadi penjumlahan data di dalam database, nilai tidak menjadi “*null*” atau ada nilainya. Pada *replace by value* diisi dengan 0, OK.



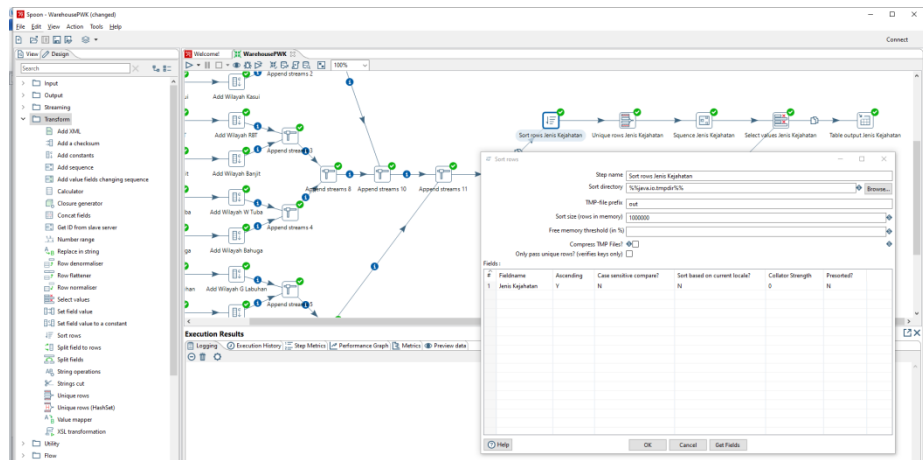
Gambar 4.12 Tampilan Jendela If Field Value is Null

5. Langkah Membuat Data Dimensi

Langkah awal dalam membuat data dimensi adalah dengan menambahkan *sort rows*, *unique rows*, *add squence*, *select values*, dan *table output* di tiap data dimensi yang diolah. Terdapat tiga data dimensi yang akan diolah, yaitu dimensi jenis kejahatan, dimensi wilayah, dan dimensi tahun.

a) Membuat *Sort Rows* Jenis Kejahatan

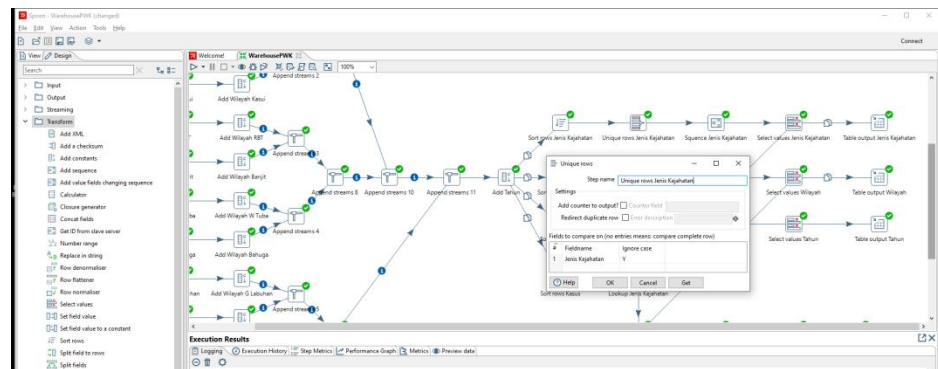
Setelah menambahkan data tahun pada *add constant* tahun sebelumnya, maka langkah selanjutnya adalah membuat *sort rows* jenis kejahatan dengan fungsi untuk mengurutkan data berdasarkan jenis kejahatan. Pada tahap ini juga dapat menghilangkan data jenis kejahatan yang ganda. Adapun tampilan jendela *sort rows* jenis kejahatan adalah seperti pada Gambar 4.13.



Gambar 4.13 Tampilan Jendela *Sort Rows* Jenis Kejahatan

b) Membuat *Unique Rows* Jenis Kejahatan

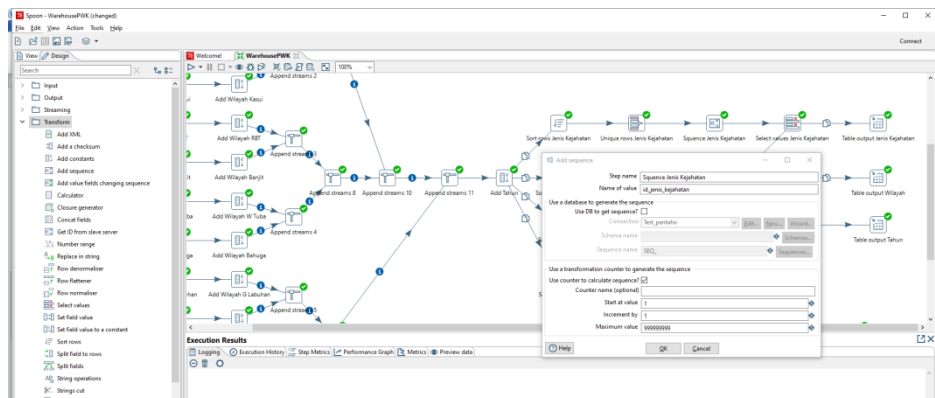
Setelah menambahkan *sort rows* jenis kejahatan, langkah selanjutnya adalah menambahkan *unique rows* jenis kejahatan seperti pada Gambar 4.14. Fungsinya adalah agar data jenis kejahatan bersifat unik atau tidak ganda.



Gambar 4.14 Tampilan Jendela *Unique Rows* Jenis Kejahatan

c) Membuat *Add Sequence* Jenis Kejahatan

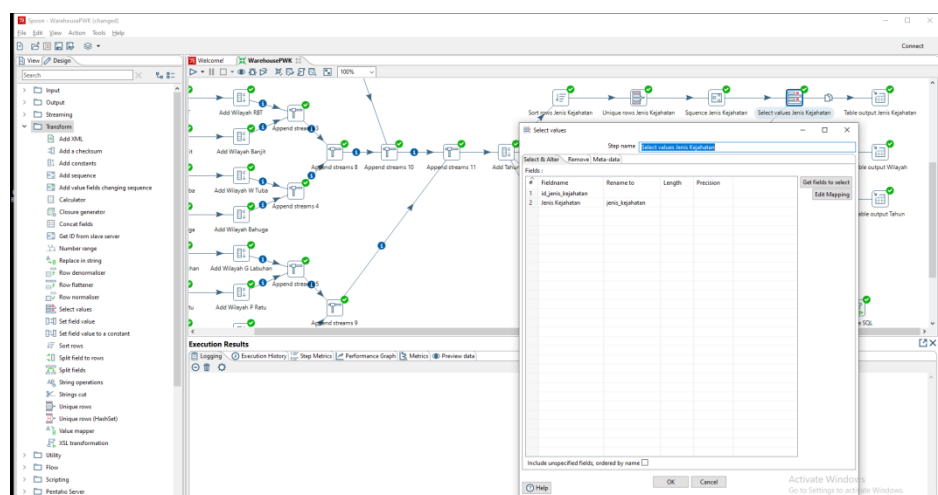
Setelah menambahkan *unique rows* jenis kejahatan, langkah selanjutnya adalah menambahkan *add sequence* jenis kejahatan seperti pada Gambar 4.15. Fungsinya adalah untuk menambahkan *field* atau indeks data yang berurutan.



Gambar 4.15 Tampilan Jendela *Add Sequence* Jenis Kejahatan

d) Membuat *Select Values* Jenis Kejahatan

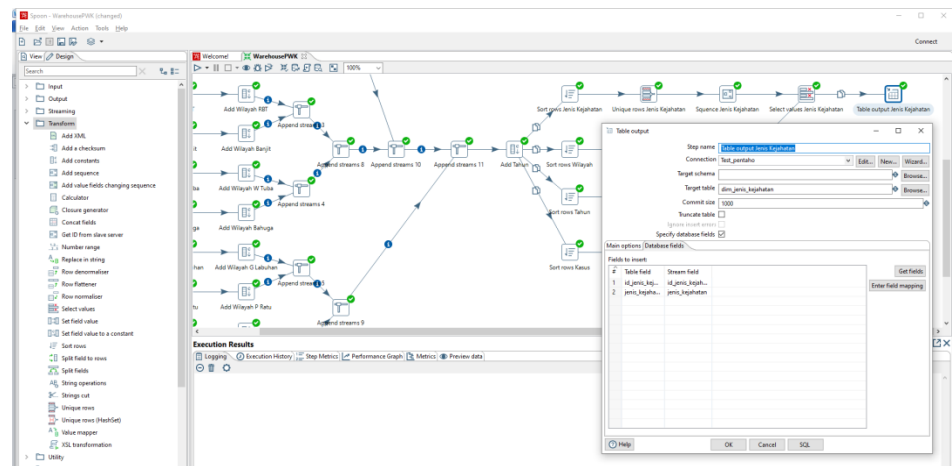
Setelah menambahkan *add sequence* jenis kejahatan, langkah selanjutnya adalah menambahkan *select values* jenis kejahatan seperti pada Gambar 4.16. Fungsinya adalah untuk memilih *field* yang akan digunakan. *Field* yang digunakan dalam proses ini adalah *field* yang berkaitan dengan jenis kejahatan. *Field* akan di *rename* sesuai dengan standar penulisan *field* di dalam *database* yaitu dengan menghapus *space* kosong dan menggunakan huruf kecil.



Gambar 4.16 Tampilan Jendela *Select Values* Jenis Kejahatan

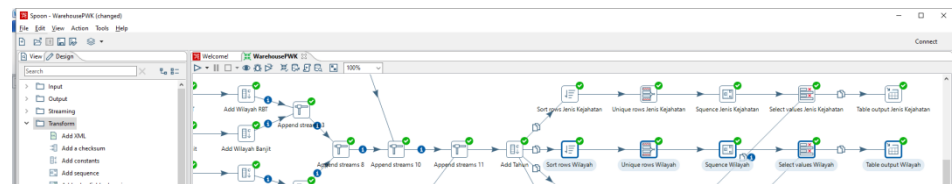
e) Membuat *Table Output* Jenis Kejahatan

Setelah menambahkan *select values* jenis kejahatan, langkah selanjutnya adalah menambahkan *table output* jenis kejahatan seperti pada Gambar 4.17. Fungsinya adalah untuk mengirimkan hasil data jenis kejahatan dari transformasi ke tabel di dalam *database*.



Gambar 4.17 Tampilan Jendela *Table Output* Jenis Kejahatan

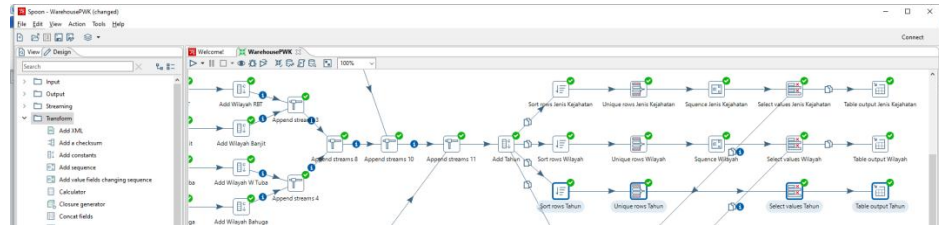
Lakukan tahap yang sama pada data dimensi wilayah dengan menambahkan *sort rows*, *unique rows*, *add sequence*, *select values*, dan *table output* menggunakan data wilayah seperti yang tertera pada Gambar 4.18.



Gambar 4.18 Tampilan Jendela Tahapan Dimensi Wilayah

Berbeda dengan tahapan data dimensi jenis kejahatan dan dimensi wilayah, dimensi tahun tidak menambahkan fitur *add sequence* karena

data tahun yang akan ditampilkan hanya satu. Adapun tahapan membuat data dimensi tahun adalah seperti pada Gambar 4.19.



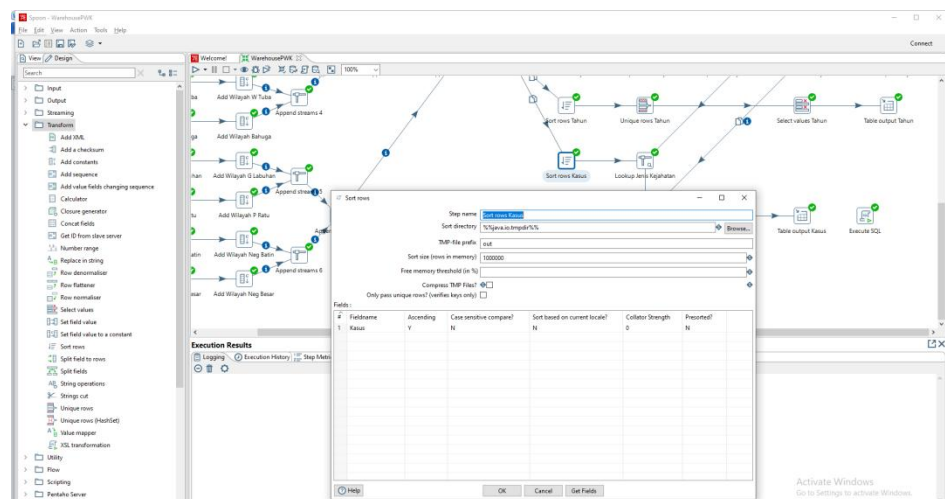
Gambar 4.19 Tampilan Jendela Tahapan Dimensi Tahun

6. Langkah Membuat Data Fakta

Setelah tahapan data dimensi dibuat, langkah selanjutnya adalah membuat data fakta dengan menambahkan *sort rows*, *lookup* jenis kejahatan, *lookup* wilayah, *select values*, dan *table output*. Data fakta yang digunakan adalah data kasus.

a) Membuat *Sort Rows* Kasus

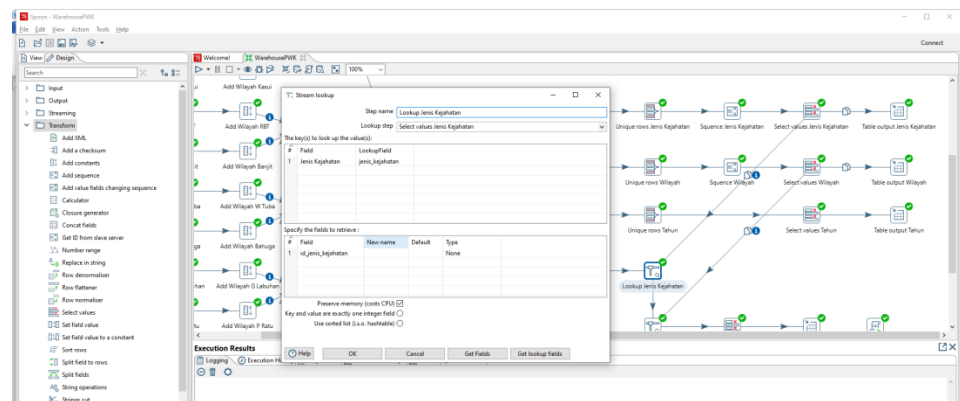
Setelah menambahkan data tahun pada *add constant* tahun sebelumnya, maka langkah selanjutnya adalah membuat *sort rows* kasus dengan fungsi untuk mengurutkan data berdasarkan kasus. Adapun tampilan jendela *sort rows* kasus adalah seperti pada Gambar 4.20.



Gambar 4.20 Tampilan Jendela *Sort Rows* Kasus

b) Membuat *Lookup* Jenis Kejahatan

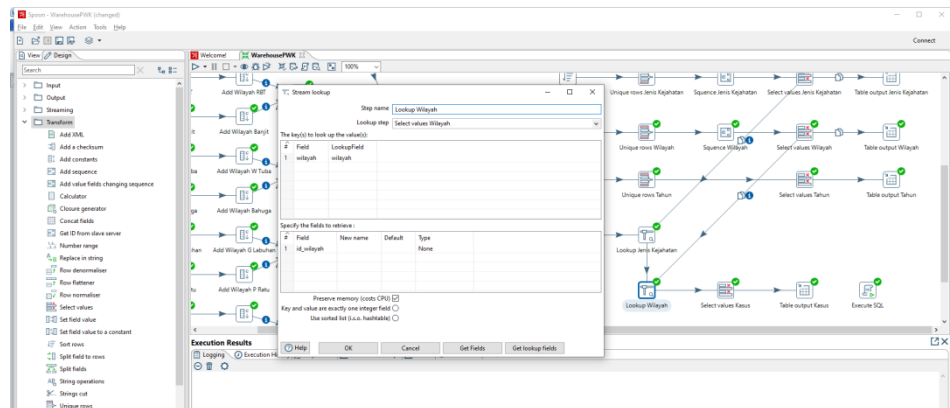
Setelah *sort rows* kasus berhasil dibuat, maka langkah selanjutnya adalah menambahkan *lookup stream*. Fungsi *lookup stream* disini untuk mengambil data berdasarkan parameter tertentu. Misalnya dengan mencari data *id_jenis_kejahatan* dengan *jenis_kejahatan* di dalam data dimensi jenis kejahatan, sedangkan pada fakta kasus tidak tertera data *id_jenis_kejahatan*. Adapun isi data pada jendela *lookup* jenis kejahatan adalah seperti pada Gambar 4.21.



Gambar 4.21 Tampilan Jendela *Lookup* Jenis Kejahatan

c) Membuat *Lookup* Wilayah

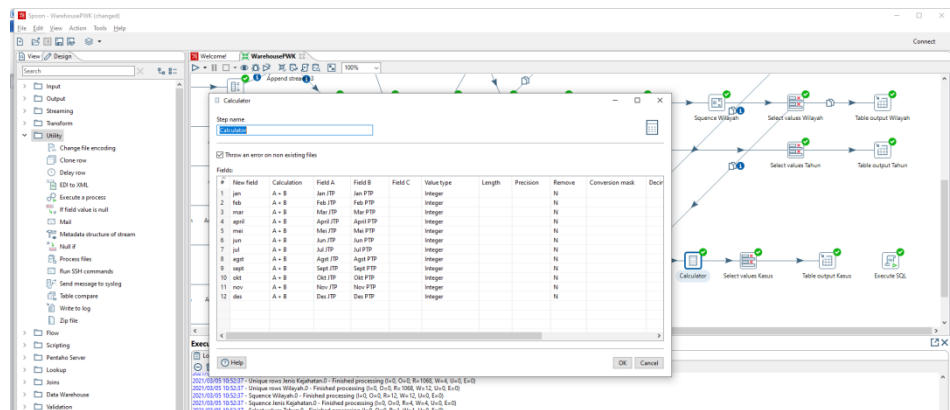
Setelah *lookup* jenis kejahatan berhasil dibuat, maka langkah selanjutnya adalah menambahkan *lookup stream*. Fungsi *lookup stream* disini untuk mengambil data berdasarkan parameter tertentu. Misalnya dengan mencari data *id_wilayah* dengan *wilayah* di dalam data dimensi wilayah, sedangkan pada fakta kasus tidak tertera data *id_wilayah*. Adapun isi data pada jendela *lookup* wilayah adalah seperti pada Gambar 4.22.



Gambar 4.22 Tampilan Jendela *Lookup Wilayah*

d) Membuat *Calculator*

Pada tahap ini diperlukan perhitungan untuk penjumlahan nilai kasus tiap bulan. Contoh pada **nilai field janptp** dan **janjtp** dijumlahkan ke dalam satu *field* yaitu **field jan**. Tahap ini diperlukan untuk mempermudah mengambil nilai jumlah kasus pada bulan januari dalam proses OLAP. Adapun isi data pada jendela *calculator* adalah seperti pada Gambar 4.23.

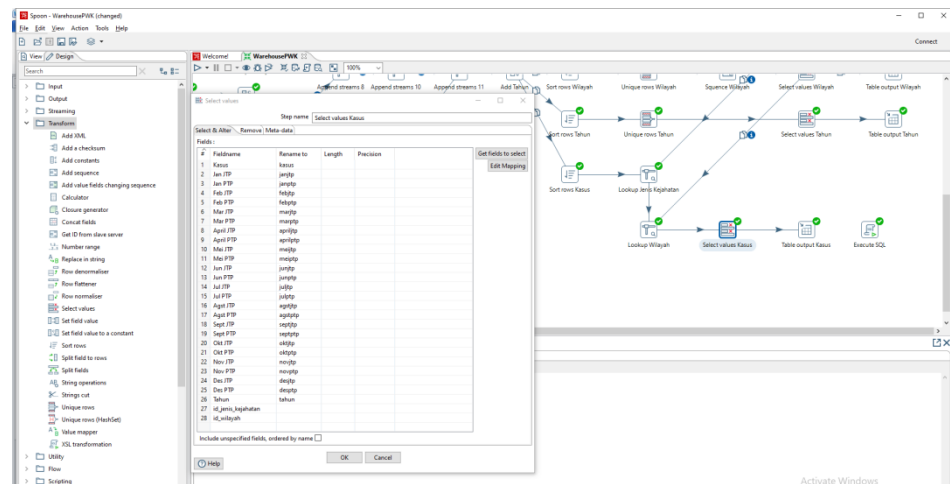


Gambar 4.23 Tampilan Jendela *Calculator*

e) Membuat *Select Values Kasus*

Setelah menambahkan *lookup* jenis kejahatan dan *lookup* wilayah, langkah selanjutnya adalah menambahkan *select values* kasus seperti pada Gambar 4.24. Fungsinya adalah untuk memilih *field* yang akan

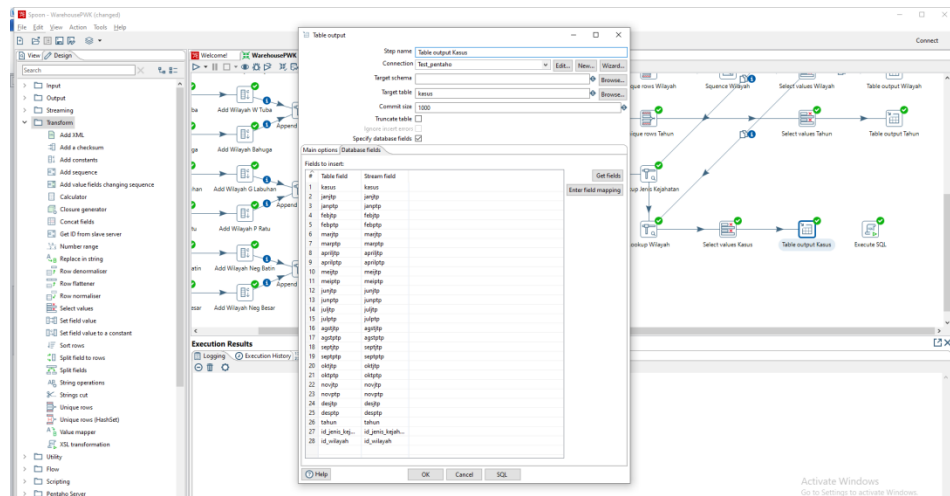
digunakan. *Field* yang digunakan dalam proses ini adalah *field* yang berkaitan dengan kasus dan berhubungan dengan jenis kejahatan serta wilayah yang berasal dari *lookup*. *Field* akan di *rename* sesuai dengan standar penulisan *field* di dalam *database* yaitu dengan menghapus *space* kosong dan menggunakan huruf kecil.



Gambar 4.24 Tampilan Jendela *Select Values Kasus*

f) Membuat *Table Output Kasus*

Setelah menambahkan *select values kasus*, langkah selanjutnya adalah menambahkan *table output kasus* seperti pada Gambar 4.25. Fungsinya adalah untuk mengirimkan hasil data kasus dari transformasi ke tabel di dalam *database*.



Gambar 4.25 Tampilan Jendela *Table Output* Kasus

7. *Execute SQL Script* Dimensi dengan Fakta

Setelah tahapan pembuatan data dimensi dan fakta berhasil dibuat, langkah selanjutnya adalah mengeksekusi data dimensi dan data fakta dengan *script* SQL. Fungsinya adalah untuk menjalankan perintah SQL dalam hal ini untuk membuat tabel di dalam *database*. Adapun tampilan jendela *execute script SQL* adalah seperti pada Gambar 4.26.

Isi *script* :

```
drop table if exists fact_kasus;
drop table if exists dim_jeniskejahatan;
drop table if exists dim_wilayah;
drop table if exists dim_tahun;
```

```
create table dim_jeniskejahatan(
idjeniskejahatan int(11) not null,
jeniskejahatan varchar(50) not null,
primary key(idjeniskejahatan)
);
```

```
create table dim_wilayah(
idwilayah int(11) not null,
wilayah varchar(30) not null,
primary key(idwilayah)
);
```

```
create table dim_tahun(
tahun int(4) not null,
primary key(tahun)
);
```

```
create table fact_kasus(
```

```

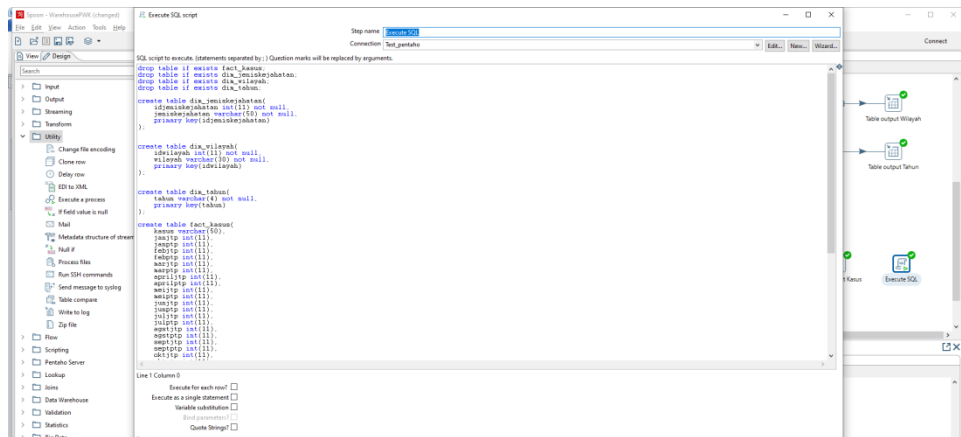
        kasus varchar(50),
        janjtp int(11),
        janptp int(11),
        febjtp int(11),
        febjtp int(11),
        marjtp int(11),
        marptp int(11),
        apriljtp int(11),
        aprilptp int(11),
        meijtp int(11),
        meiptp int(11),
        junjtp int(11),
        junptp int(11),
        juljtp int(11),
        julptp int(11),
        agstjtp int(11),
        agstptp int(11),
        septjtp int(11),
        septptp int(11),
        oktjtp int(11),
        oktptp int(11),
        novjtp int(11),
        novptp int(11),
        desjtp int(11),
        desptp int(11),
        tahun int(4),
        idjeniskejahatan int(11),
        idwilayah int(11),
        jan int(11),
        feb int(11),
        mar int(11),
        april int(11),
        mei int(11),
        jun int(11),
        jul int(11),
        agst int(11),
        sept int(11),
        okt int(11),
        nov int(11),
        des int(11),
        jtp int(11),
        ptp int(11)
    );

```

```

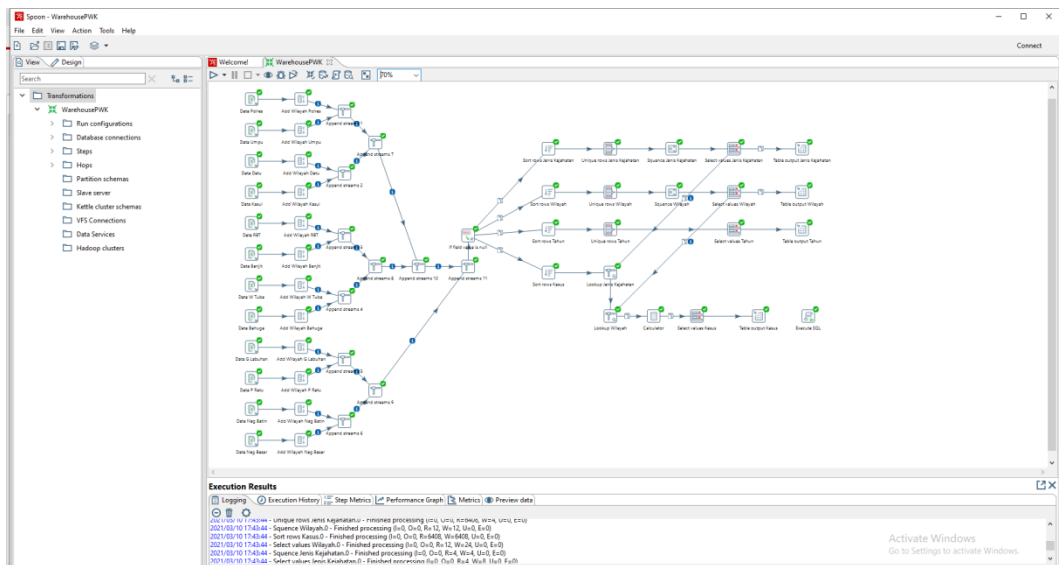
ALTER TABLE `fact_kasus` ADD CONSTRAINT `fk1` FOREIGN KEY
(`idjeniskejahatan`) REFERENCES `dim_jeniskejahatan`(`idjeniskejahatan`) ON DELETE CASCADE ON
UPDATE CASCADE;
ALTER TABLE `fact_kasus` ADD CONSTRAINT `fk2` FOREIGN KEY
(`idwilayah`) REFERENCES `dim_wilayah`(`idwilayah`) ON DELETE
CASCADE ON UPDATE CASCADE;
ALTER TABLE `fact_kasus` ADD CONSTRAINT `fk3` FOREIGN KEY
(`tahun`) REFERENCES `dim_tahun`(`tahun`) ON DELETE CASCADE ON
UPDATE CASCADE;

```



Gambar 4.26 Tampilan Jendela *Execute Script SQL*

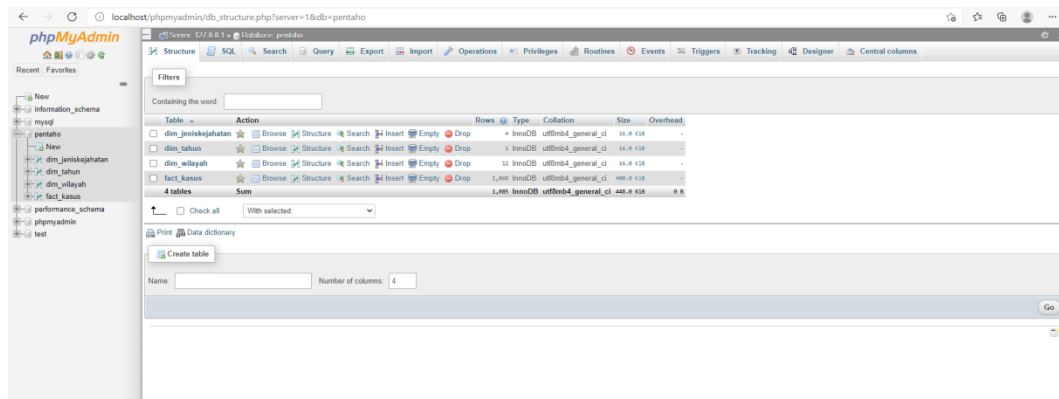
Adapun langkah/tahapan pembuatan data *warehouse* secara keseluruhan adalah dapat di lihat pada Gambar 4.27.



Gambar 4.27 Tampilan Keseluruhan Proses *Warehouse*

4.3.3 Tampilan Data di *Database*

Setelah tahap proses keseluruhan warehouse selesai, maka tampilan data warehouse di dalam database adalah seperti pada Gambar 4.28.

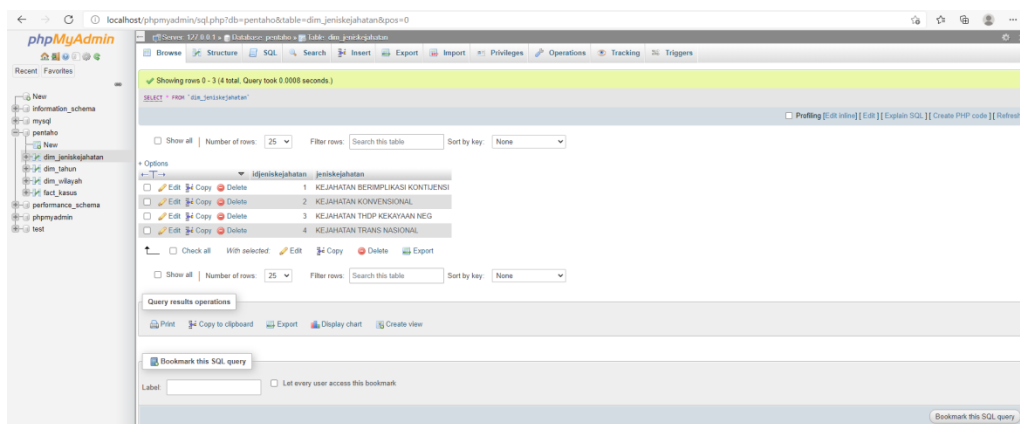


Gambar 4.28 Tampilan Data di *Database*

Adapun tampilan data pada tiap tabel adalah sebagai berikut :

1. Tabel Dimensi Jenis Kejahatan

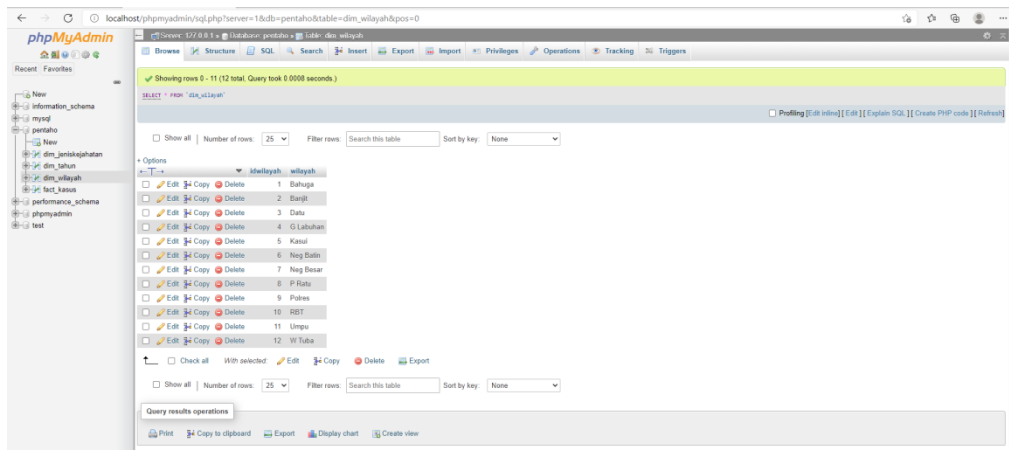
Tampilan data yang ada di tabel dimensi jenis kejahatan adalah seperti pada Gambar 4.29.



Gambar 4.29 Tampilan Data pada Tabel Dimensi Jenis Kejahatan di *Database*

2. Tabel Dimensi Wilayah

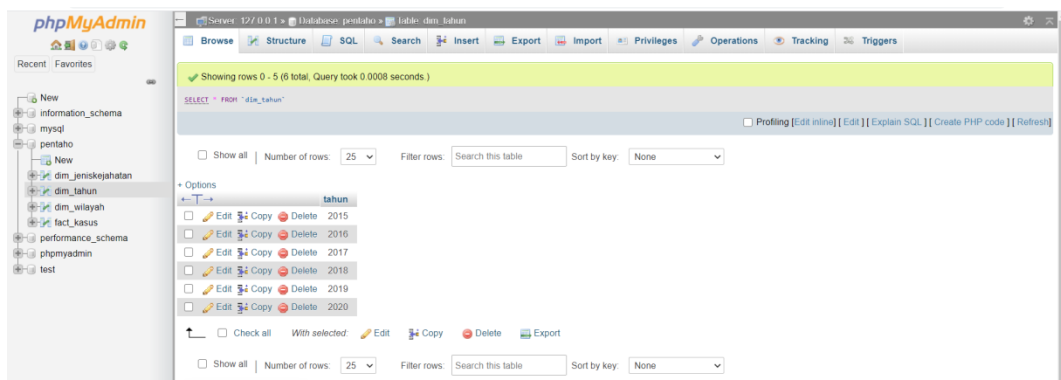
Tampilan data yang ada di tabel dimensi wilayah adalah seperti pada Gambar 4.30.



Gambar 4.30 Tampilan Data pada Tabel Dimensi Wilayah di *Database*

3. Tabel Dimensi Tahun

Tampilan data yang ada di tabel dimensi tahun adalah seperti pada Gambar 4.31.



Gambar 4.31 Tampilan Data pada Tabel Dimensi Tahun di *Database*

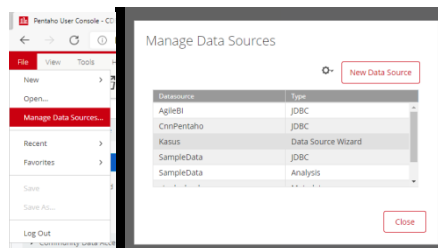
4. Tabel Fakta Kasus

Tampilan data yang ada di tabel fakta kasus adalah seperti pada Gambar 4.32.

selanjutnya adalah *login* menggunakan *user* admin. Tahapan-tahapan dalam pembuatan OLAP adalah sebagai berikut :

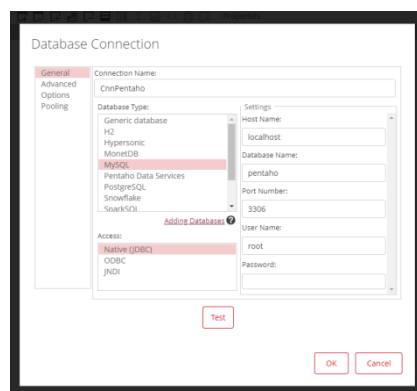
a. *Manage Data Sources*

Sebelum memasuki tahap CDE Dashboard, langkah pertama yang dilakukan adalah membuat koneksi dengan *database* yang telah dibuat dalam pengolahan data pada Pentaho Data *Intergration*. Tahap membuka jendela manage data source adalah pilih *file-manage data source*. Berikutnya sistem menampilkan jendela manage data source seperti pada Gambar 4.34.



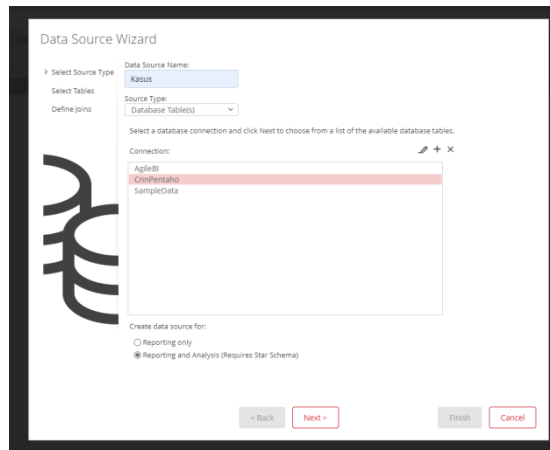
Gambar 4.34 Tampilan Jendela *Manage Data Sources*

Setelah jendela *manage data sources* terbuka, langkah selanjutnya dengan membuat koneksi baru dengan cara masuk ke jendela *setting* terlihat pada Gambar 4.35, kemudian isi data sesuai dengan *database* yang telah dibuat sebelumnya. Data connection disini diberi nama **CnnPentaho** (nama sesuai selera). Untuk memastikan *database* terkoneksi, tekan **test**.



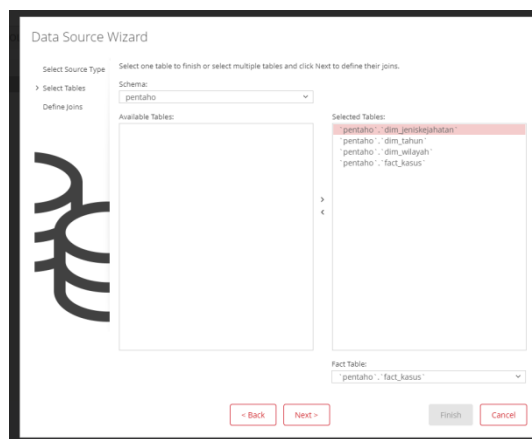
Gambar 4.35 Tampilan Jendela *Database Connection*

Setelah terlihat *datasource* **CnnPentaho** dengan *type* **JDBC**, maka selanjutnya adalah membuat *data source wizard* dengan cara menekan tombol *new data source* dan sistem menampilkan jendela *data source wizard* seperti pada Gambar 4.36.



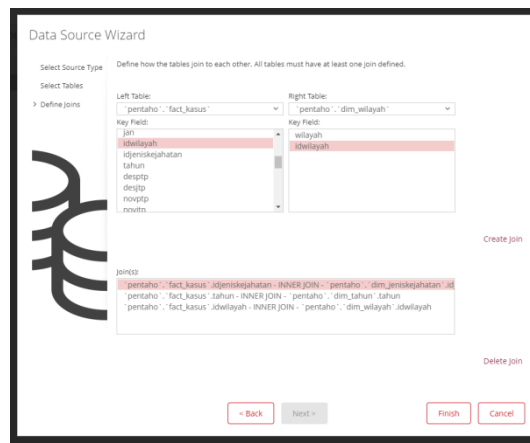
Gambar 4.36 Tampilan Jendela *Data Source Wizard*

Pada *data source name* diisi dengan Kasus, *connection* pilih CnnPentaho, kemudian pilih *reporting and analysis (requires star schema)*, *next*. Data pada kolom *available tables* dipindah ke kolom *selected tables*, pada *fact table* diisi “pentaho”, “fact_kasus), *next* terlihat seperti pada Gambar 4.37.



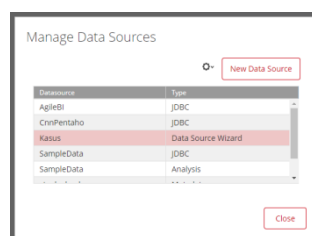
Gambar 4.37 Tampilan Jendela *Data Source Wizard Tables*

Langkah selanjutnya adalah join semua tabel yang ada pada *database* pentaho. Pada *left table* “pentaho , fact_kasus” dipilih **idjeniskejahatan** dan *right table* “pentaho, dim_jeniskejahatan” dipilih **idjeniskejahatan** kemudian *create join*. Setelah di *join*, maka akan muncul tulisan hasil *join* di kolom *join*. Pada *left table* “pentaho , fact_kasus” dipilih **idwilayah** dan *right table* “pentaho, dim_wilayah” dipilih **idwilayah** kemudian *create join*. Pada *left table* “pentaho , fact_kasus” dipilih **tahun** dan *right table* “pentaho, dim_tahun” dipilih **tahun** kemudian *create join* kemudian *finish*. Hasil dari *join* adalah seperti pada Gambar 4.38.



Gambar 4.38 Tampilan Jendela *Data Source Wizard Join Table*

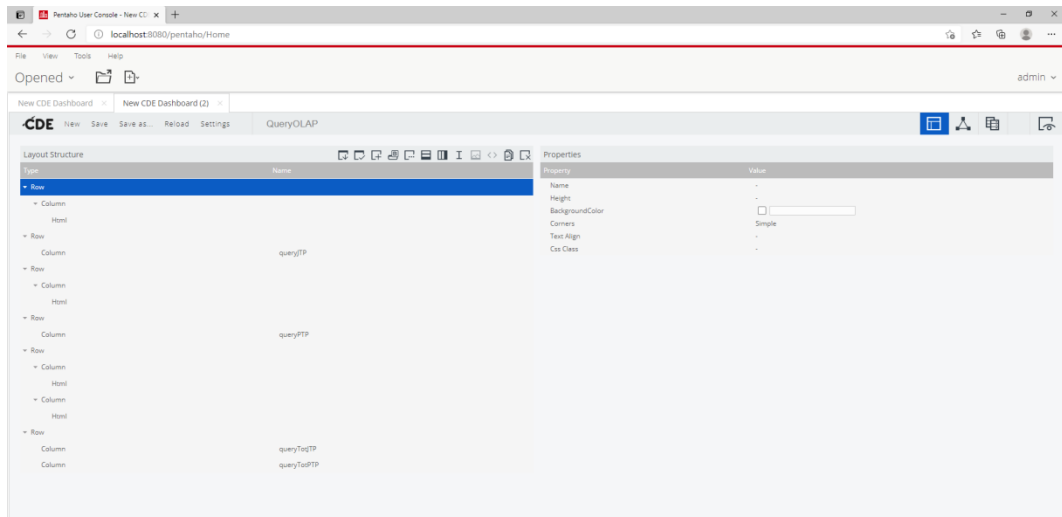
Setelah *finish*, maka akan muncul jendela yang terdapat isi *datasource* **Kasus** dengan *type data source wizard* seperti pada Gambar 4.39.



Gambar 4.39 Tampilan Jendela Koneksi *Data Source Wizard*

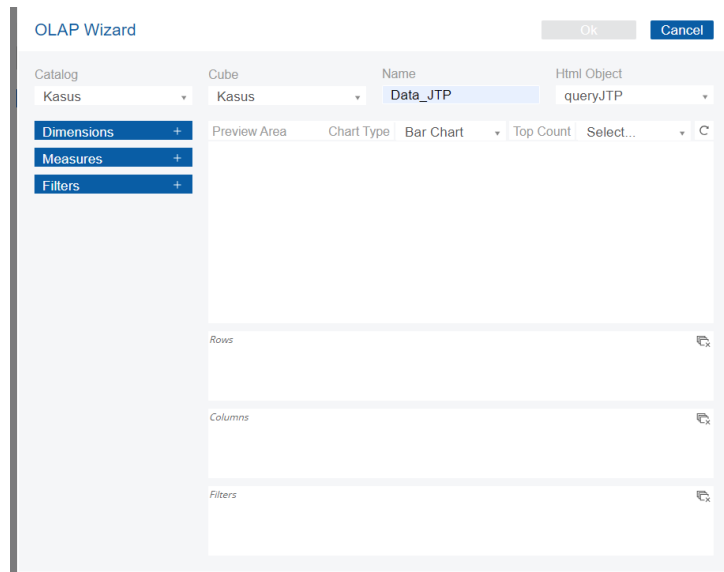
b. *CDE Dashboard*

Setelah koneksi selesai dibuat, langkah selanjutnya adalah mendesain *template* pada *CDE dashboard* dengan cara *file-new-CDE dashboard* seperti pada Gambar 4.40.



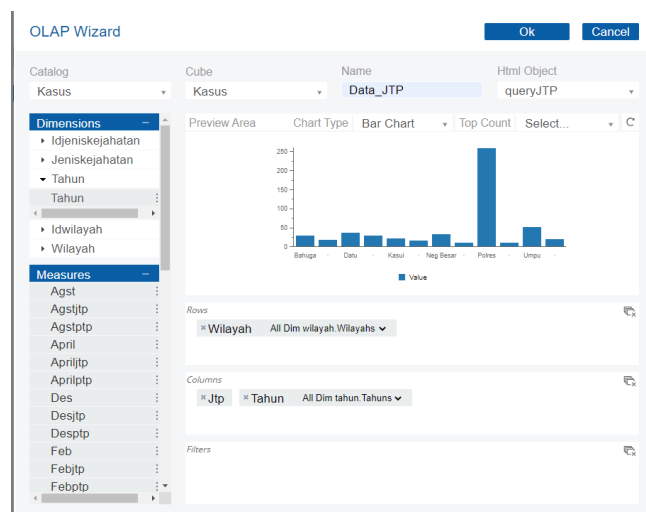
Gambar 4.40 Tampilan Jendela *CDE Dashboard Layout Panel*

Setelah *layout panel* dibuat, maka langkah selanjutnya adalah membuka *datasource panel*, pilih *wizard --> OLAP chart wizard*. Kemudian pada *catalog* diisi dengan **kasus**, *cube* otomatis akan terisi kasus, *name* diisi dengan *dataJTP* (penamaan sesuai selera) karena data yang akan ditampilkan adalah data jumlah kasus JTP per wilayah berdasarkan tahun, *html object* diisi dengan *queryJTP* (*row layout panel* yang telah dibuat sebelumnya) seperti pada Gambar 4.41.



Gambar 4.41 Tampilan Jendela *OLAP Wizard*

Setelah itu data OLAP yang akan ditampilkan pada penelitian ini adalah *bar chart* data jumlah kasus JTP per wilayah berdasarkan tahun. Oleh karena itu, maka pada *dimensions* pilih **wilayah** dan geret ke dalam kolom *row*. Kemudian pada *measures* pilih **jtp dan tahun** dan geret ke dalam tabel *columns*. Pada *chart type* pilih **bar chart** dan sistem menampilkan data jumlah kasus seperti pada Gambar 4.42.

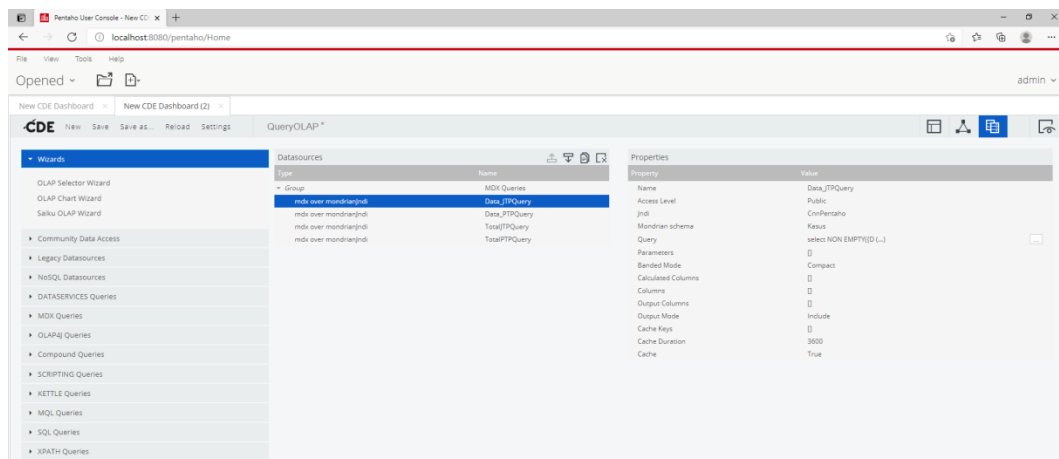


Gambar 4.42 Tampilan Jendela OLAP *Bar Chart* Kasus JTP Tiap Wilayah Berdasarkan Tahun

Untuk menampilkan data tabel kasus kejahatan, maka pada *query* yang ada di dalam wilayah pada *datasource panel* diisi dengan *script* : **SELECT * FROM dim_wilayah**. Sedangkan pada *query* yang ada di dalam kasus pada *datasource panel* diisi dengan *script* :

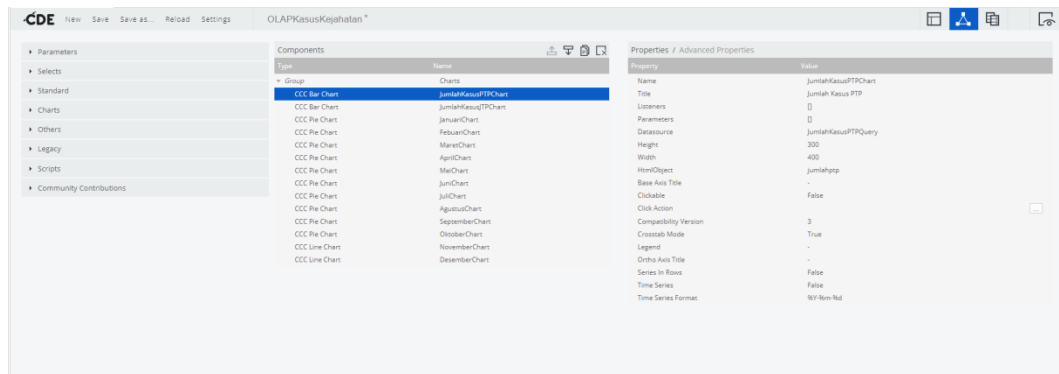
**SELECT tahun, kasus, jtp, ptp FROM fact_kasus
WHERE idwilayah = \${pWilayah}**

Adapun hasil keseluruhan dari data jumlah kasus per bulan adalah seperti pada gambar-gambar di bawah ini.



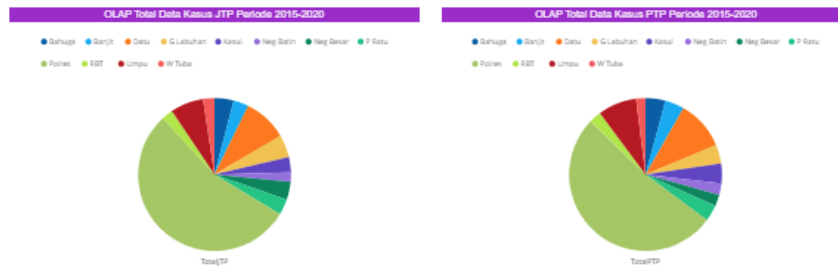
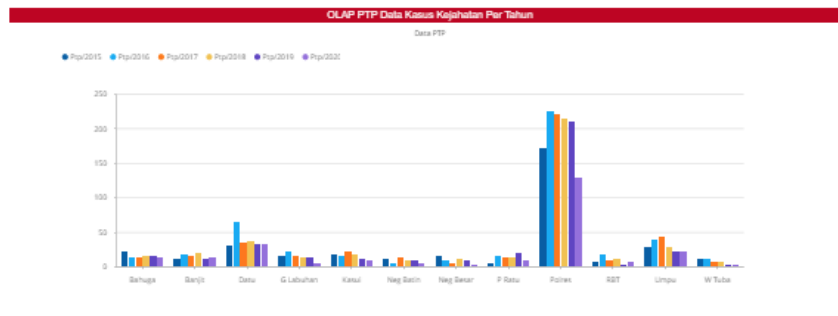
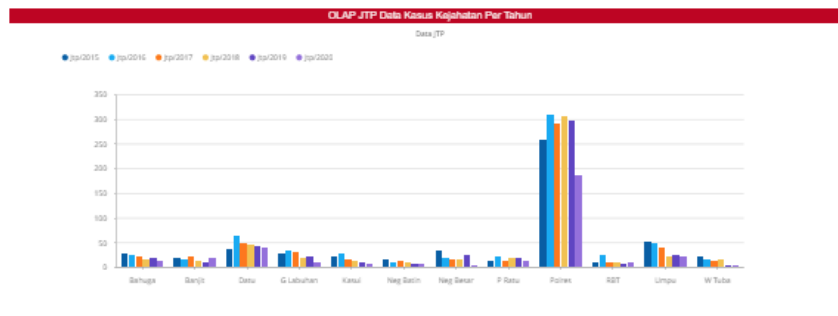
Gambar 4.43 Tampilan Jendela *Datasource Panel*

Gambar 4.43 menunjukkan item-item *data source* dari *data source panel* yang dibuat menggunakan *wizard*. Pada saat panel wizard muncul, hal yang harus diisi adalah nama data source, jndi, dan schema. Jika dibutuhkan konfigurasi data source dapat dirubah pada panel properti sebelah kanan.



Gambar 4.44 Tampilan Jendela *Components Panel*

Gambar 4.44 menunjukkan komponen *chart* berupa *bar chart*, *pie chart*, dan *line chart*. Seperti halnya pada *data source*, konfigurasi komponen dapat dilakukan pada panel properti sebelah kanan. Komponen *chart* yang ditampilkan berupa data kasus kejahatan beserta jumlah kasus jtp dan ptp.



Tabel Kasus Kejahatan

Wilayah: Polres

Show 10 entries

tahun	kasus	jtp	ptp
2015	ANIAYA RINGAN	0	0
2016	ANIAYA RINGAN	2	1
2017	ANIAYA RINGAN	9	9
2018	ANIAYA RINGAN	3	2
2019	ANIAYA RINGAN	1	1
2020	ANIAYA RINGAN	6	6
2015	ANIRAT	1	0
2016	ANIRAT	0	0
2017	ANIRAT	0	0
2018	ANIRAT	19	18

Showing 1 to 10 of 534 entries

Previous 1 2 3 4 5 ... 54 Next

Gambar 4.45 Tampilan OLAP Bar Chart Jumlah Kasus Kejahatan PTP dan JTP

Gambar 4.5 menunjukkan hasil dari OLAP kasus kejahatan. Pada *bar chart* dan *pie chart* menunjukkan hasil kasus kejahatan per tahun dimulai dari tahun 2015 sampai tahun 2020 di tiap wilayah. Sedangkan untuk tabel kasus kejahatan menampilkan data jumlah kasus kejahatan berdasarkan wilayah yang dipilih melalui pilihan wilayah yang ada di tengah atas. Data yang ditampilkan berupa tahun, kasus, jumlah jtp dan jumlah ptp.