

**Tugas Project Data Warehouse**  
**Analisis Pola Kehadiran Janji Temu Pasien dari Data Medical Appointment**



Disusun oleh :

Nandana Rifqi Irfansyah

**DEPARTEMEN SISTEM INFORMASI**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS BRAWIJAYA**  
**MALANG**  
**2023**

## DAFTAR ISI

<b>FP01-SID: Capaian Proyek Identifikasi Kebutuhan dan Penyusunan Paket Informasi</b>	<b>2</b>
Pembagian Role Kelompok Data Warehouse	2
Information Package	2
Kategori umur:	3
<b>FP02-SID: Capaian Proyek Penyusunan Desain DW</b>	<b>4</b>
A. Desain Logikal	4
B. Desain Fisikal	5
C. DDL	6
<b>FP03-SID: Capaian Proyek Penyusunan ETL</b>	<b>7</b>
a. Job DIMENSI_JENIS_KELAMIN	7
b. Job DIMENSI_UMUR	9
c. Job DIMENSI_KANAL	11
d. Job DIMENSI_TIPE_JANJI	14
e. Job FACT	15
<b>FP04-SID: Capaian Proyek Penyusunan SQL OLAP</b>	<b>19</b>
A. Rank	19
B. Row_Number	19
C. Rollup	20
D. Cube	21

## FP01-SID: Capaian Proyek Identifikasi Kebutuhan dan Penyusunan Paket Informasi

### Pembagian Role Kelompok Data Warehouse

Agnes Imanuela Chanani	Product Manager
Irsya Salim Fadhilah	System Architect
Sekar Ria Amanda	System Design
Nandana Rifqi Irfansyah	Business Analyst

### Information Package

**Information Subject :** *Analisis Pola Kehadiran Janji Temu Pasien dengan Dokter*

Dimensi				
Hierarki	Tipe Janji	Kanal	Umur	Jenis Kelamin
	Kategori Janji	Kategori Kanal	Kategori Umur	Kategori Jenis Kelamin
			Batas Atas	
			Batas Bawah	
<b>Measurement:</b> Jumlah pertemuan berdasarkan tipe janji, jenis kanal, jenis kelamin, dan kategori umur				

Dimensi	Penjelasan
<b>Tipe Janji</b>	Dimensi ini dapat memberikan informasi tentang jenis janji temu yang dibuat antara pasien dan dokter. Terdapat 2 kategori tipe janji yaitu medis dan prosedural
<b>Kanal</b>	Dimensi ini mencakup kanal atau cara komunikasi yang digunakan untuk membuat dan mengkonfirmasi janji temu. Misalnya, janji temu dapat dibuat melalui call center, personal, dan web

<b>Umur</b>	Dimensi ini melibatkan usia pasien yang membuat janji temu. Dengan mempertimbangkan faktor ini, pengelolaan janji temu dapat disesuaikan untuk memenuhi kebutuhan pasien berdasarkan kelompok usia mereka. Sesuai dengan batas atas dan batas bawah yang telah ditentukan pada tabel dibawah.
<b>Jenis Kelamin</b>	Dimensi ini melibatkan jenis kelamin pasien. Berdasarkan jenis kelamin, pola kehadiran pasien mungkin berbeda antara laki-laki dan perempuan.

Kategori umur:

<b>Kategori Umur</b>	<b>Batas Atas</b>	<b>Batas Bawah</b>
Balita	4	0
Anak-anak	16	5
Remaja	34	17
Dewasa	54	35
Lansia	200	55

## FP02-SID: Capaian Proyek Penyusunan Desain DW

### A. Desain Logikal

Kelompok kami menggunakan skema star (*star schema*) karena memiliki beberapa kelebihan, antara lain:

- **Simplicity:** Skema star memiliki struktur yang sederhana dan mudah dipahami. Dimensi terhubung ke tabel fakta melalui foreign key, sehingga memudahkan pengguna untuk melakukan query data.
- **Query performance:** Skema star memiliki query performance yang baik, karena hanya memiliki satu tabel fakta dan dimensi yang terpisah. Hal ini memungkinkan pengguna untuk melakukan query data dengan lebih cepat.
- **Scalability:** Skema star lebih scalable, karena memungkinkan penambahan dimensi tanpa mempengaruhi kinerja sistem.

Skema star disusun dengan menggunakan draw.io sesuai dengan information package yang telah dibuat sebelumnya. Star schema ini terdiri dari 3 dimensi antara lain:

#### **Dimensi tipe janji**

- id\_tipe\_janji int
- kategori\_janji string

#### **Dimensi kanal**

- id\_kanal int
- kategori\_kanal string

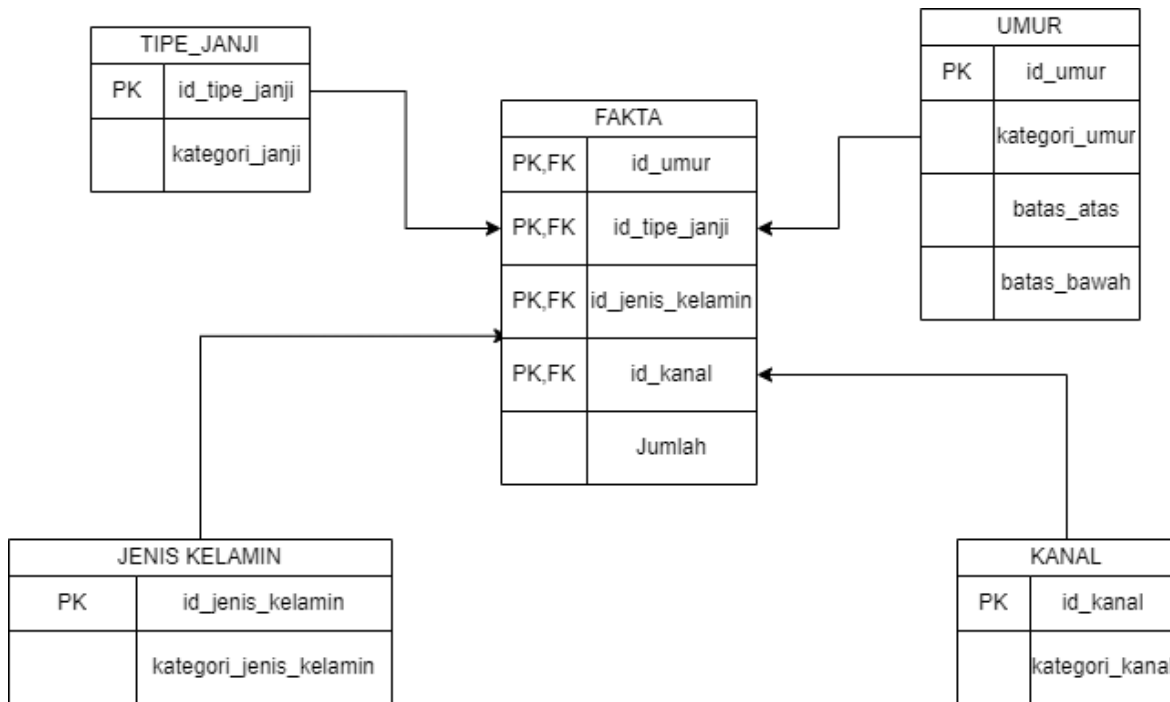
#### **Dimensi umur**

- id\_umur int
- kategori\_umur string
- batas\_atas int
- batas\_bawah int

#### **Dimensi jenis kelamin**

- id\_jenis\_kelamin int
- kategori\_jenis\_kelamin string

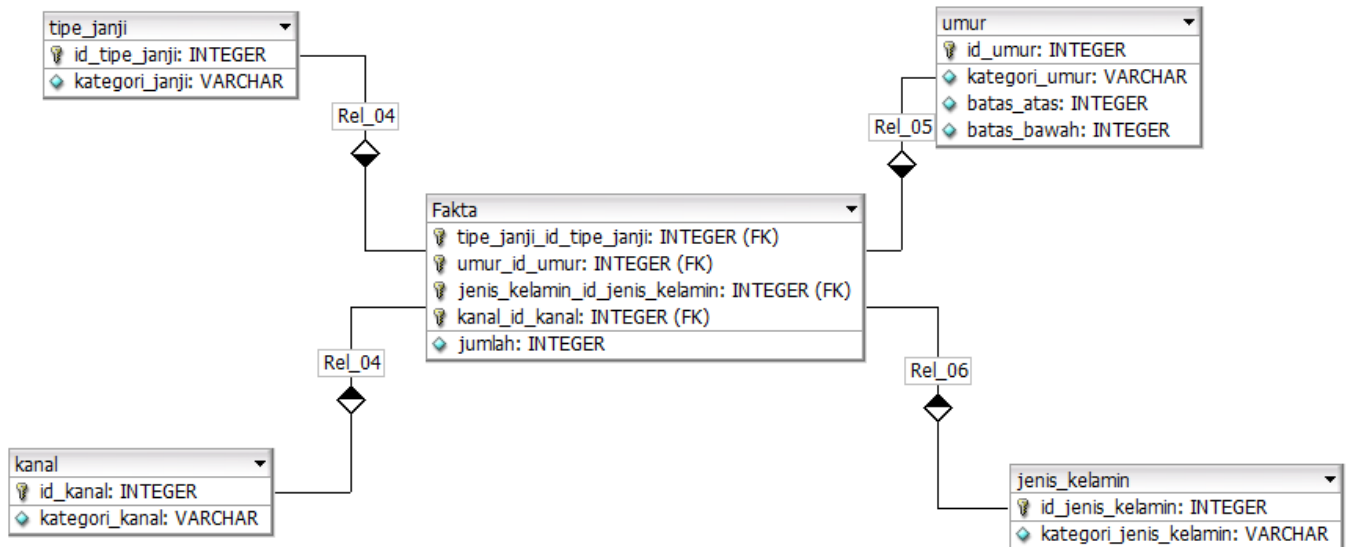
## SKEMA STAR



Gambar Star Schema

## B. Desain Fisikal

Desain fisikal *star schema* dengan menggunakan aplikasi DBDesigner :



Gambar Implementasi Desain Fisikal

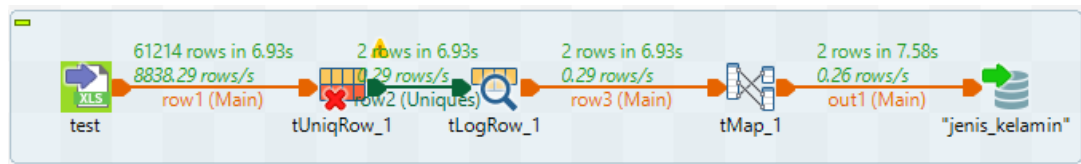
### C. DDL

```
CREATE TABLE TIPE_JANJI(  
ID_TIPE_JANJI INT NOT NULL PRIMARY KEY,  
KATEGORI_JANJI STRING  
)  
  
CREATE TABLE UMUR(  
ID_UMUR INT NOT NULL PRIMARY KEY,  
KATEGORI_UMUR STRING  
BATAS_ATAS INT  
BATAS_BAWAH INT  
)  
  
CREATE TABLE KANAL(  
ID_KANAL INT NOT NULL PRIMARY KEY,  
KATEGORI_KANAL STRING  
)  
  
CREATE TABLE JENIS_KELAMIN(  
ID_JENIS_KELAMIN INT NOT NULL PRIMARY KEY,  
KATEGORI_JENIS_KELAMIN STRING  
)  
  
CREATE TABLE FAKTA(  
ID_TIPE_JANJI INT NOT NULL,  
ID_UMUR INT NOT NULL,  
ID_KANAL INT NOT NULL,  
ID_JENIS_KELAMIN INT NOT NULL,  
FOREIGN KEY (ID_TIPE_JANJI) REFERENCES TIPE_JANJI(ID_TIPE_JANJI),  
FOREIGN KEY (ID_UMUR) REFERENCES UMUR(ID_UMUR),  
FOREIGN KEY (ID_KANAL) REFERENCES KANAL(ID_KANAL),  
FOREIGN KEY (ID_JENIS_KELAMIN) REFERENCES  
JENIS_KELAMIN(ID_JENIS_KELAMIN),  
JUMLAH INT  
)
```

## FP03-SID: Capaian Proyek Penyusunan ETL

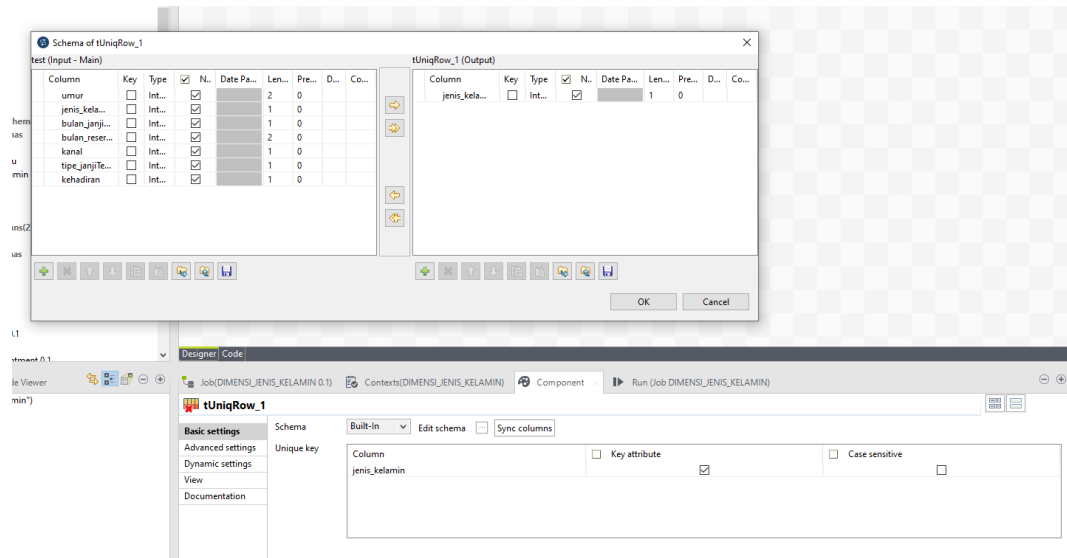
Pada proses Extract Transform Load (ETL) mencakup dari bagaimana data diambil, data dibersihkan, dan bagaimana data diformat sesuai dengan perancangan fisikal dari data warehouse. Pembuatan data untuk tabel dimensi adalah langkah pertama dari memasukkan data. Langkah yang diperlukan adalah pencarian data unik serta melakukan surrogate key untuk primary key pada tabel dimensi. Kemudian adalah pembuatan data untuk tabel Fakta dengan melakukan pencocokan key dari masing-masing tabel Dimensi serta agregasi untuk kolom JUMLAH.

1. Membuka Talend Open Studio for Data Integration dan membuat project baru.
2. Membuat file excel sumber data dan mencari lokasi file sumber data.
3. Membuat koneksi ke basis data.
4. Melakukan test connection ke database.
5. Melakukan *retrieve schema*.
6. Membuat job dengan menekan tombol **create job**.
  - a. Job **DIMENSI\_JENIS\_KELAMIN**
    - 1) Melakukan drag and drop metadata yang sudah dibuat ke dalam workspace Job.

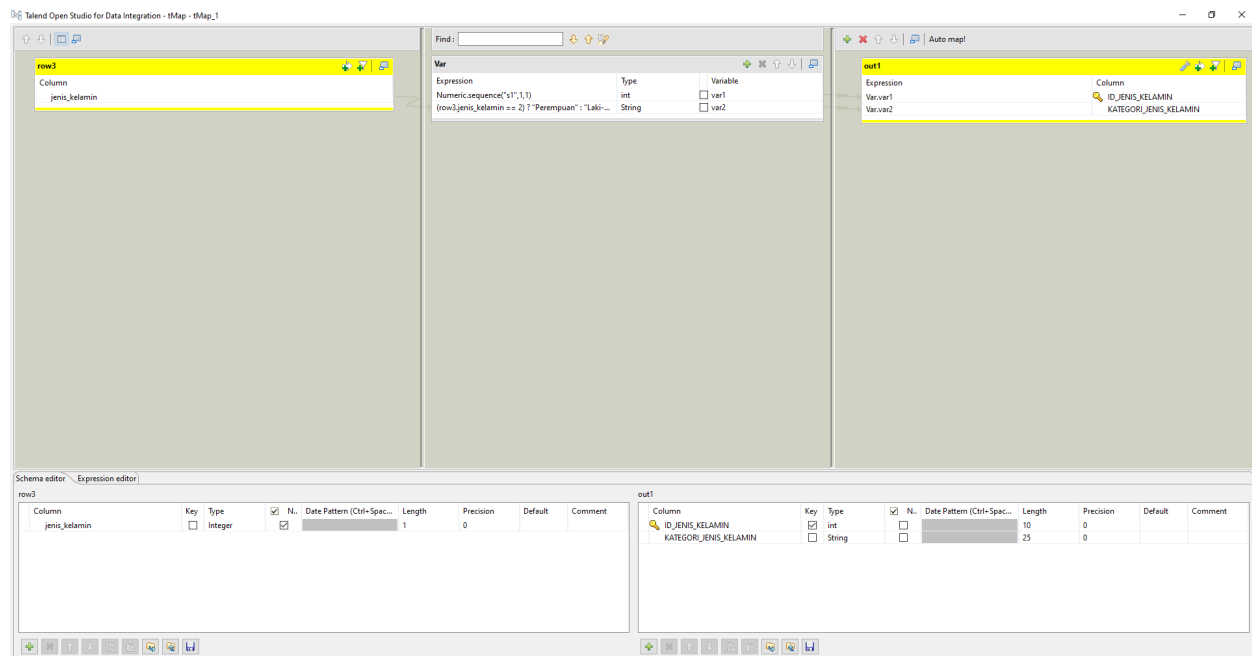


- 2) Menghubungkan metadata ke **tUniqRow\_1**. Pada **tUniqRow\_1**, memilih tab Component dan melakukan edit *Schema*.





- 3) Menghubungkan **tUniqRow\_1** dengan **tLogRow\_1** untuk melihat hasilnya dengan melakukan Run Job.
- 4) Menghubungkan **tLogRow\_1** dengan **tMap\_1** kemudian melakukan double-click pada **tMap\_1** dan membuat output baru. Pada *Schema Editor*, ubah *Schema Output* out1 dengan menambahkan 2 kolom. Menghubungkan variabel dengan skema output dan skema input ke variabel.



Expression:

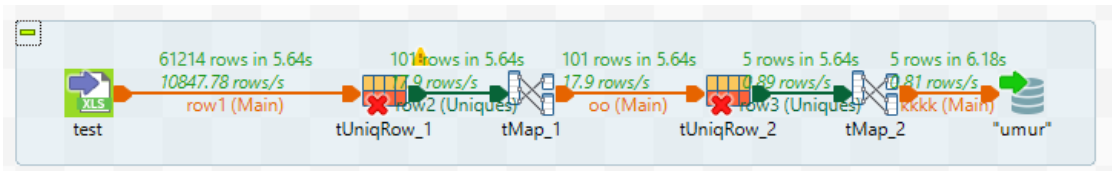
```
(row3.jenis_kelamin == 2) ? "Perempuan" : "Laki-Laki"
```

- 5) Menghubungkan **tMap\_1** dengan tabel **DIMENSI\_JENIS\_KELAMIN** yang sudah dibuat pada koneksi basis data dan memilih **tDBOutput**.
- 6) Melakukan run pada Job kemudian data akan masuk ke dalam tabel yang telah dibuat sebelumnya.

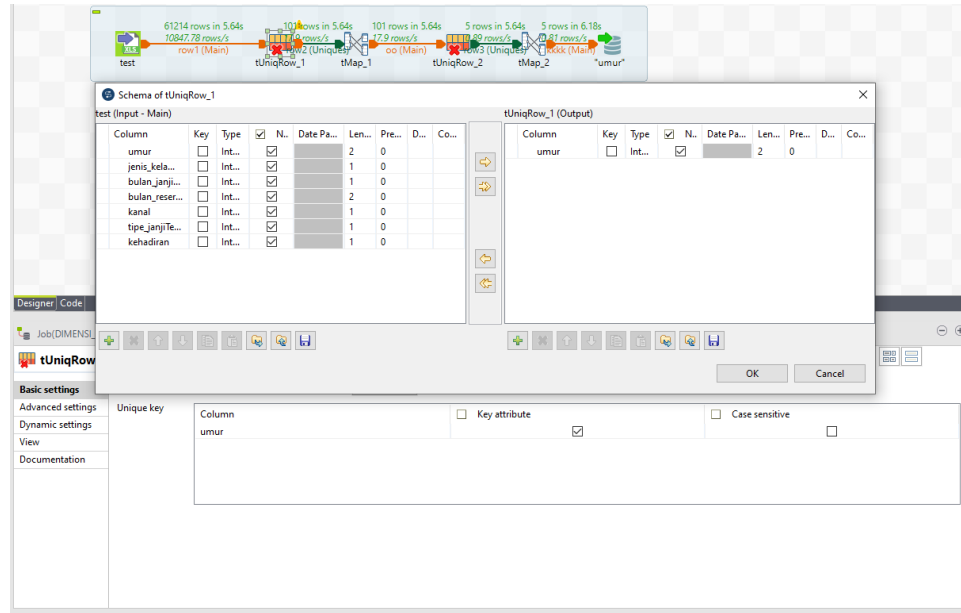
	ID_JENIS_KELAMIN	KATEGORI_JENIS_KELAMIN
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Laki-Laki
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Perempuan

b. Job **DIMENSI\_UMUR**

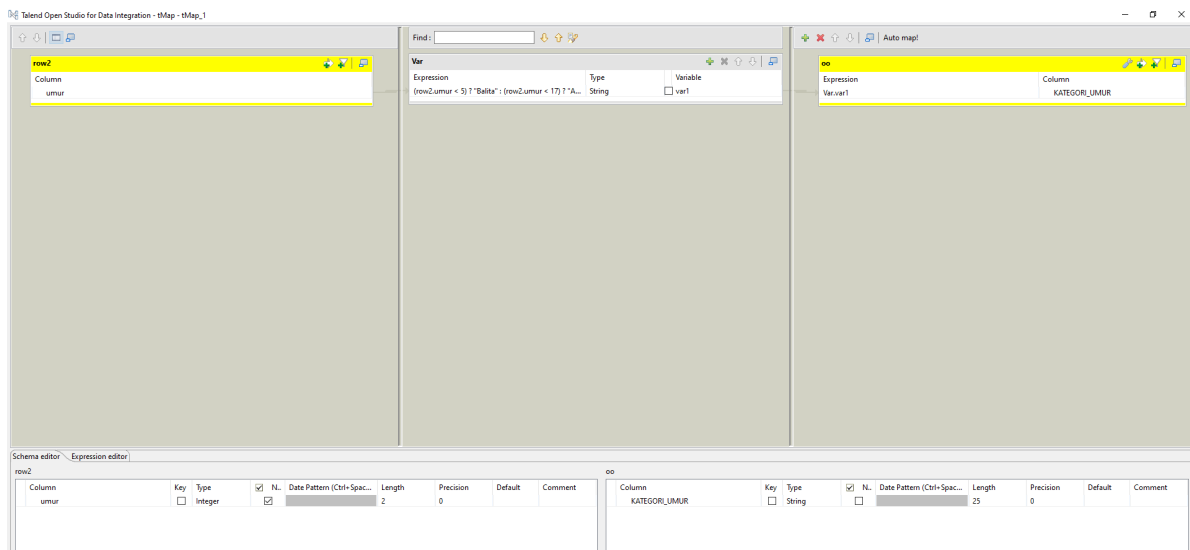
- 1) Melakukan drag and drop metadata yang sudah dibuat ke dalam workspace Job. Kemudian menghubungkannya dengan **tUniqRow\_1**.



- 2) Pada **tUniqRow\_1**, memilih tab Component dan melakukan edit *Schema*.



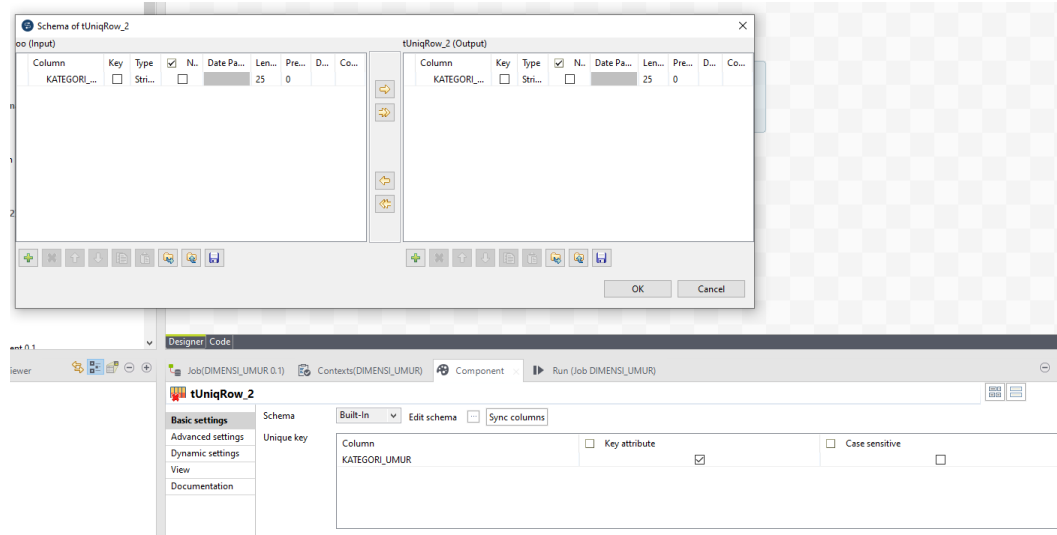
- 3) Menghubungkan **tUniqRow\_1** ke **tMap\_1** kemudian melakukan double-click pada **tMap\_1** dan membuat output baru. Pada *Schema Editor*, ubah *Schema Output* dengan menambahkan 1 kolom. Menghubungkan variabel dengan skema output dan skema input ke variabel.



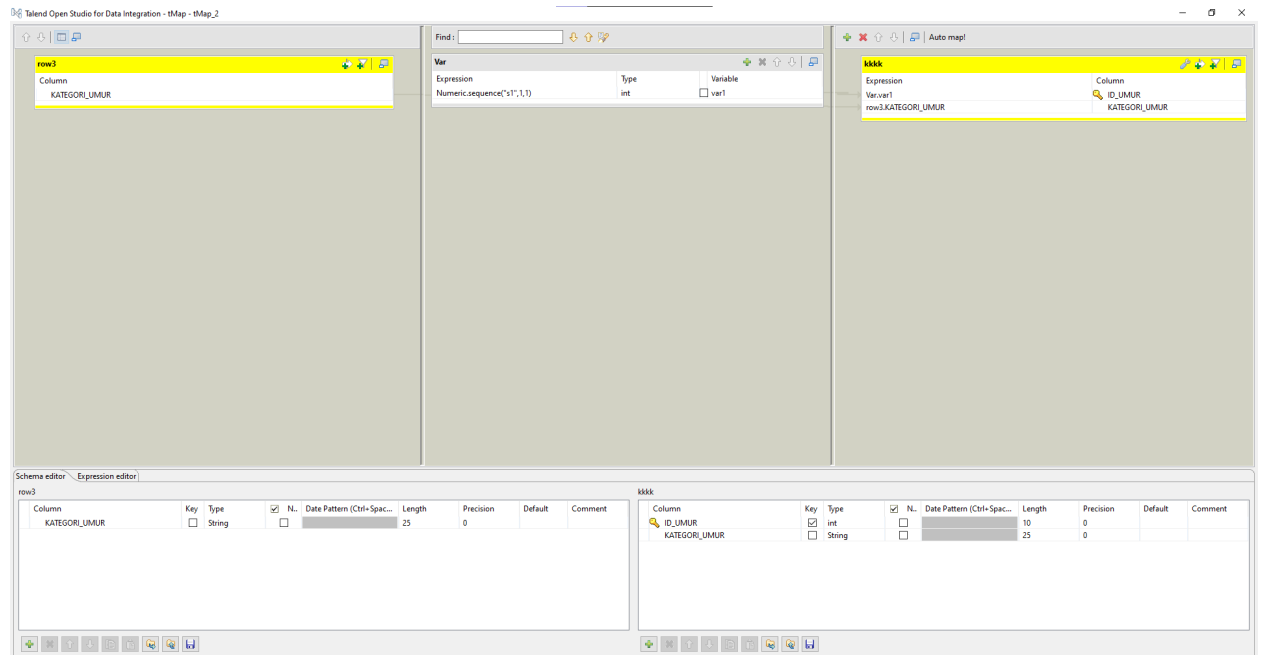
Expression:

(row2.umur < 5) ? "Balita" : (row2.umur < 17) ? "Anak-Anak" : (row2.umur < 25) ? "Remaja" : (row2.umur < 55) ? "Dewasa" : "Lansia"

- 4) Menghubungkan **tMap\_1** ke **tUniqRow\_2**, kemudian memilih tab Component dan melakukan edit *Schema*.



- 5) Menghubungkan **tUniqRow\_2** dengan **tMap\_2**.



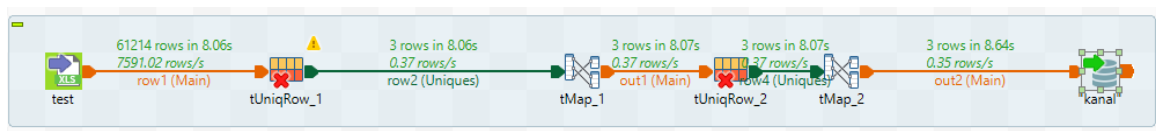
- 6) Kemudian menghubungkan **tMap\_2** dengan tabel **DIMENSI\_UMUR** yang sudah dibuat pada koneksi basis data dan memilih **tDBOutput**.

- 7) Melakukan run pada Job kemudian data akan masuk ke dalam tabel yang telah dibuat sebelumnya.

	ID_UMUR	KATEGORI_UMUR	BATAS_ATAS	BATAS_BAWAH
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Lansia	100	55
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Dewasa	54	35
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Remaja	34	17
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Anak-Anak	16	5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Balita	4	0

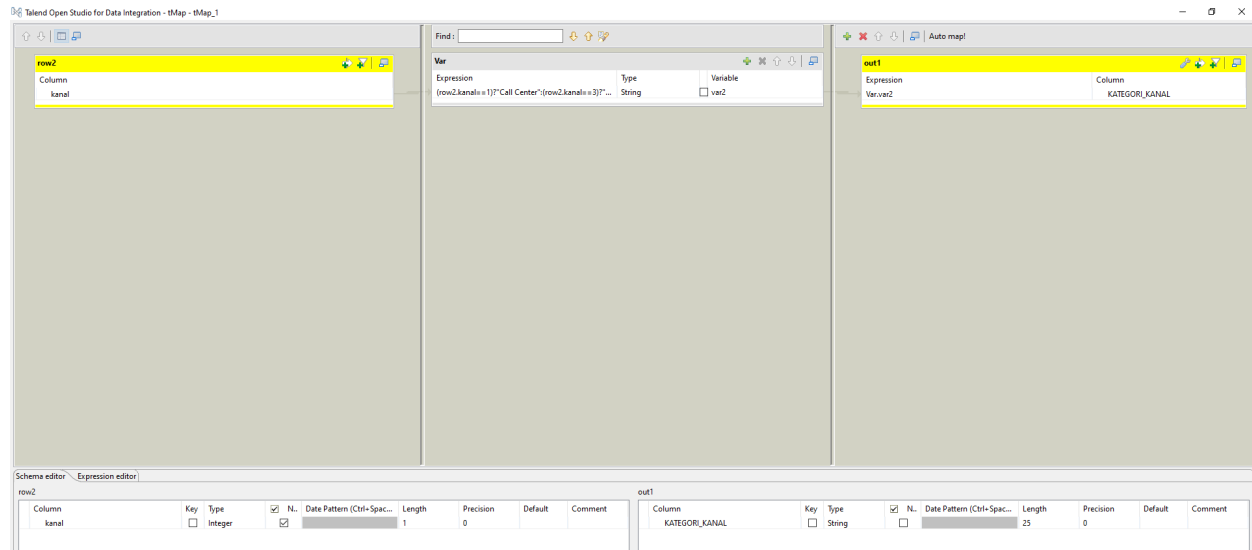
c. Job **DIMENSI\_KANAL**

- 1) Melakukan drag and drop metadata yang sudah dibuat ke dalam workspace Job.



- 2) Menghubungkan **tUniqRow\_1** ke **tMap\_1** kemudian melakukan double-click pada **tMap\_1** dan membuat output baru. Pada *Schema Editor*, ubah *Schema Output* out1 dengan menambahkan 1 kolom. Menghubungkan variabel dengan skema output dan skema input ke variabel.

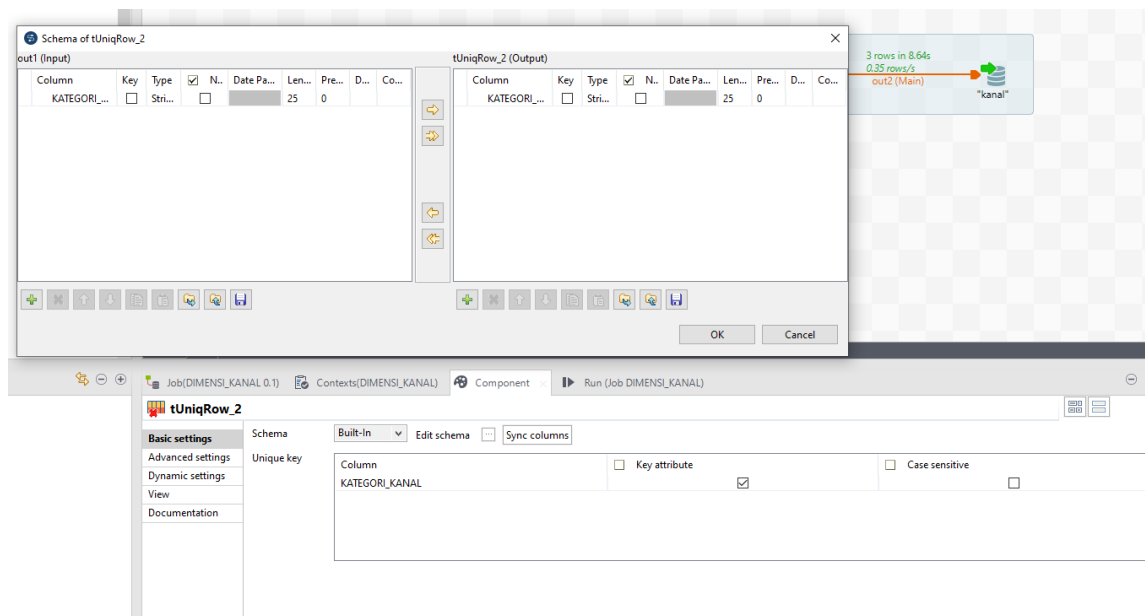
Column	<input type="checkbox"/> Key attribute	<input type="checkbox"/> Case sensitive
kanal	<input checked="" type="checkbox"/>	<input type="checkbox"/>



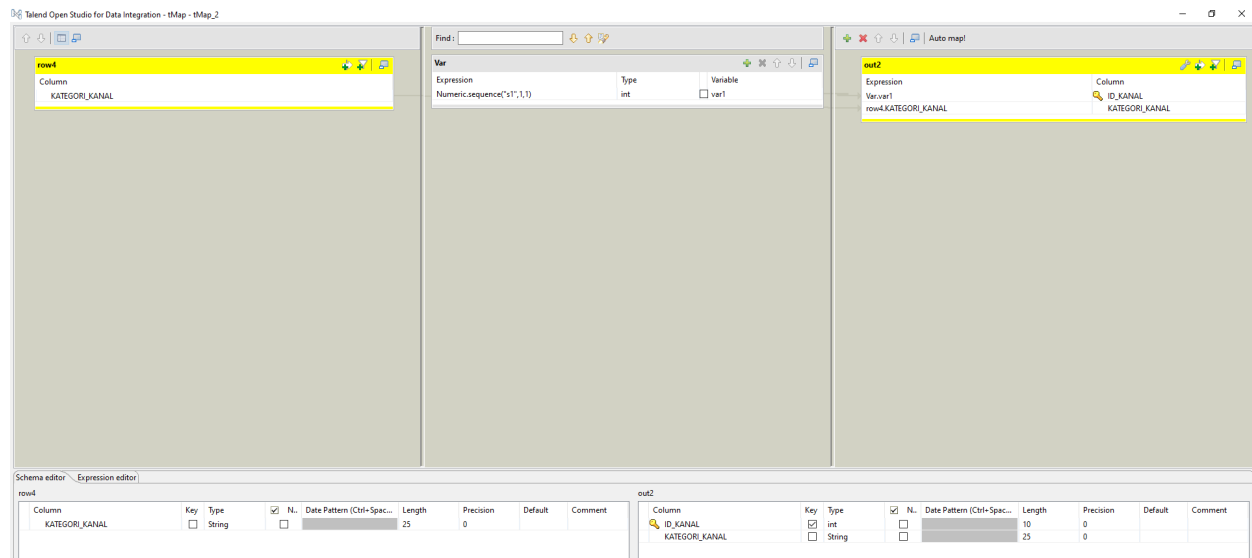
Expression:

`(row2.kanal == 1) ? "Call Center" : (row2.kanal == 2) ? "Personal" : "Web"`

- 3) Menghubungkan **tMap\_1** ke **tUniqRow\_2**.
- 4) Pada **tUniqRow\_2**, memilih tab Component dan melakukan edit *Schema*.



- 5) Kemudian menghubungkan **tUniqRow\_2** ke **tMap\_2** kemudian melakukan double-click pada **tMap\_2** dan membuat output baru. Pada *Schema Editor*, ubah *Schema Output out2* dengan menambahkan 2 kolom. Menghubungkan variabel dengan skema output dan skema input ke skema output.

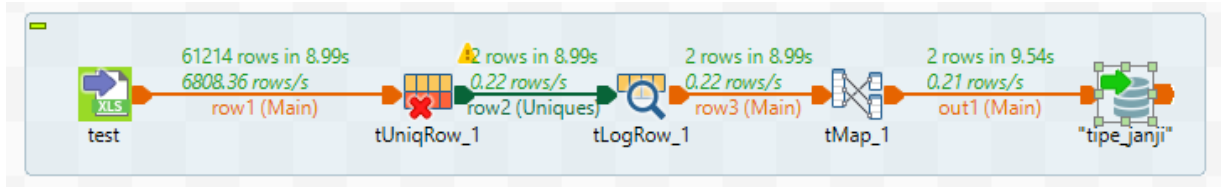


- 6) Menghubungkan **tMap\_2** dengan tabel **DIMENSI\_KANAL** yang sudah dibuat pada koneksi basis data dan memilih **tDBOutput**.
- 7) Melakukan run pada Job kemudian data akan masuk ke dalam tabel yang telah dibuat sebelumnya.

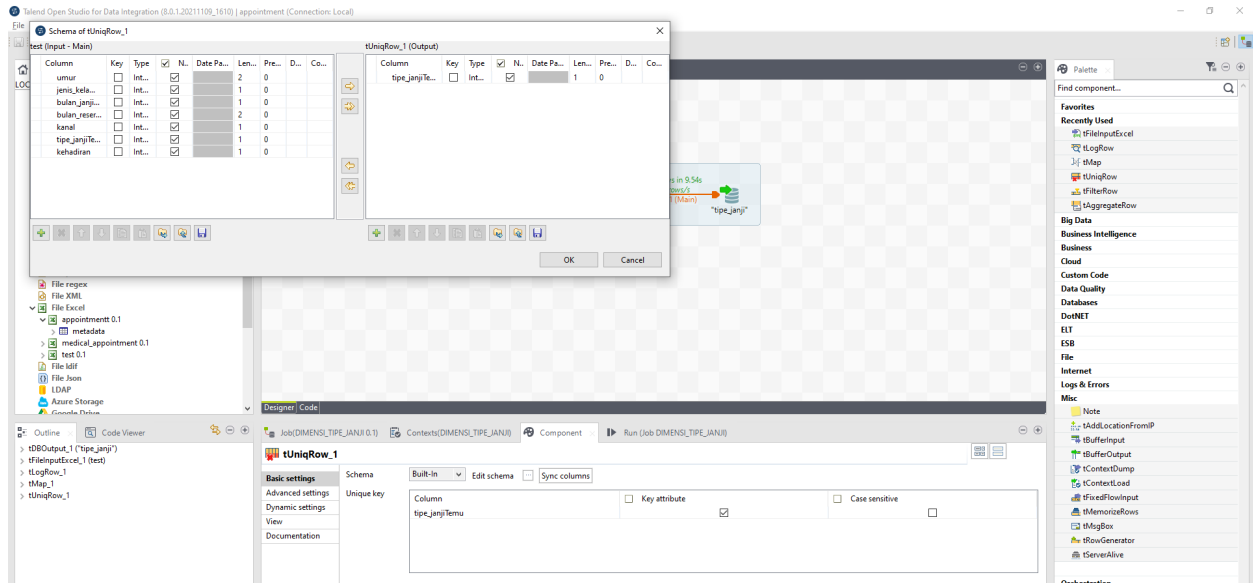
	ID_KANAL	KATEGORI_KANAL
<input type="checkbox"/> Edit Copy Delete	1	Call Center
<input type="checkbox"/> Edit Copy Delete	2	Personal
<input type="checkbox"/> Edit Copy Delete	3	Web

#### d. Job **DIMENSI\_TIPE\_JANJI**

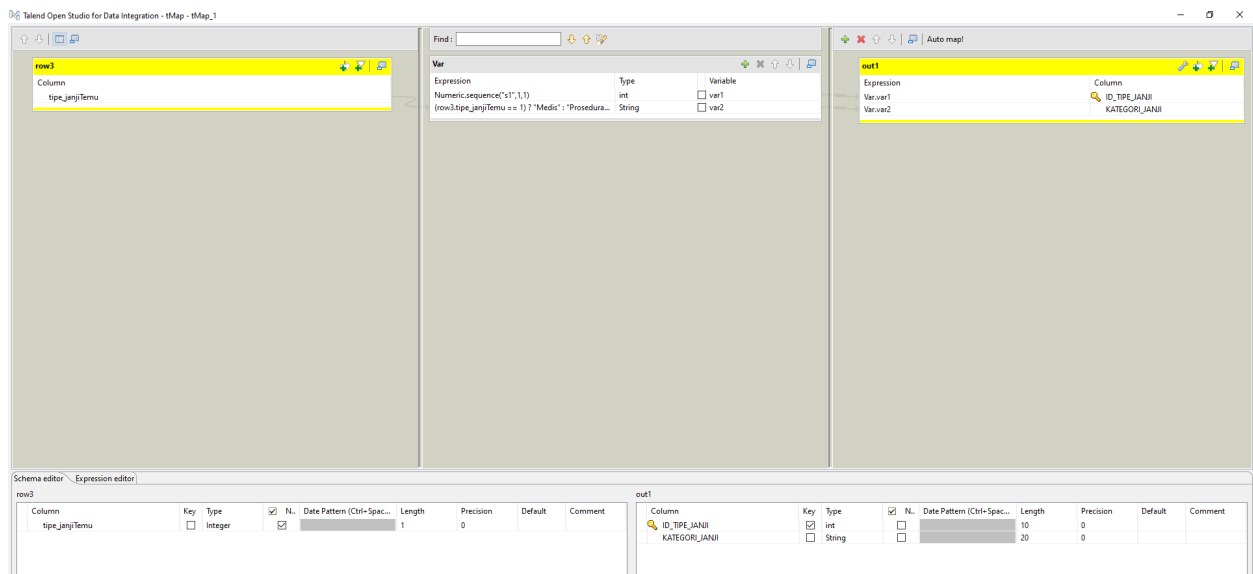
- 1) Melakukan drag and drop metadata yang sudah dibuat ke dalam workspace Job kemudian menghubungkannya ke **tUniqRow\_1**.



2) Pada **tUniqRow\_1**, memilih tab Component dan melakukan edit *Schema*.



3) Menghubungkan **tUniqRow\_1** ke **tMap\_1** kemudian melakukan double-click pada **tMap\_1** dan membuat output baru. Pada *Schema Editor*, ubah *Schema Output out1* dengan menambahkan 2 kolom. Menghubungkan variabel dengan skema output dan skema input ke variabel.





Expression:

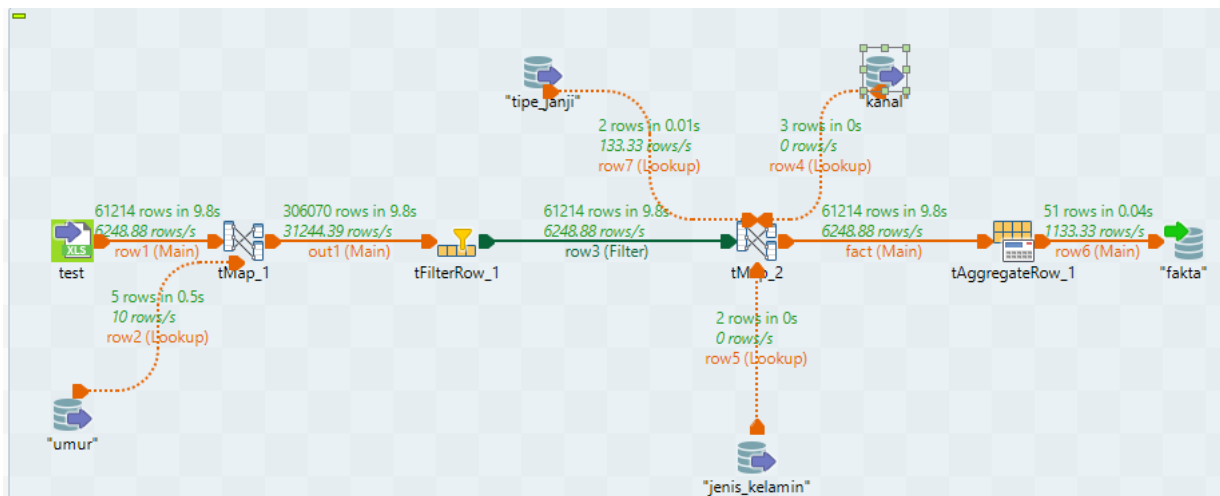
```
(row3.tipe_janji == 1) ? "Medis" : "Prosedural"
```

- 4) Menghubungkan **tMap\_1** dengan tabel **DIMENSI\_TPE\_JANJI** yang sudah dibuat pada koneksi basis data dan memilih **tDBOutput**.
- 5) Melakukan run pada Job kemudian data akan masuk ke dalam tabel yang telah dibuat sebelumnya.

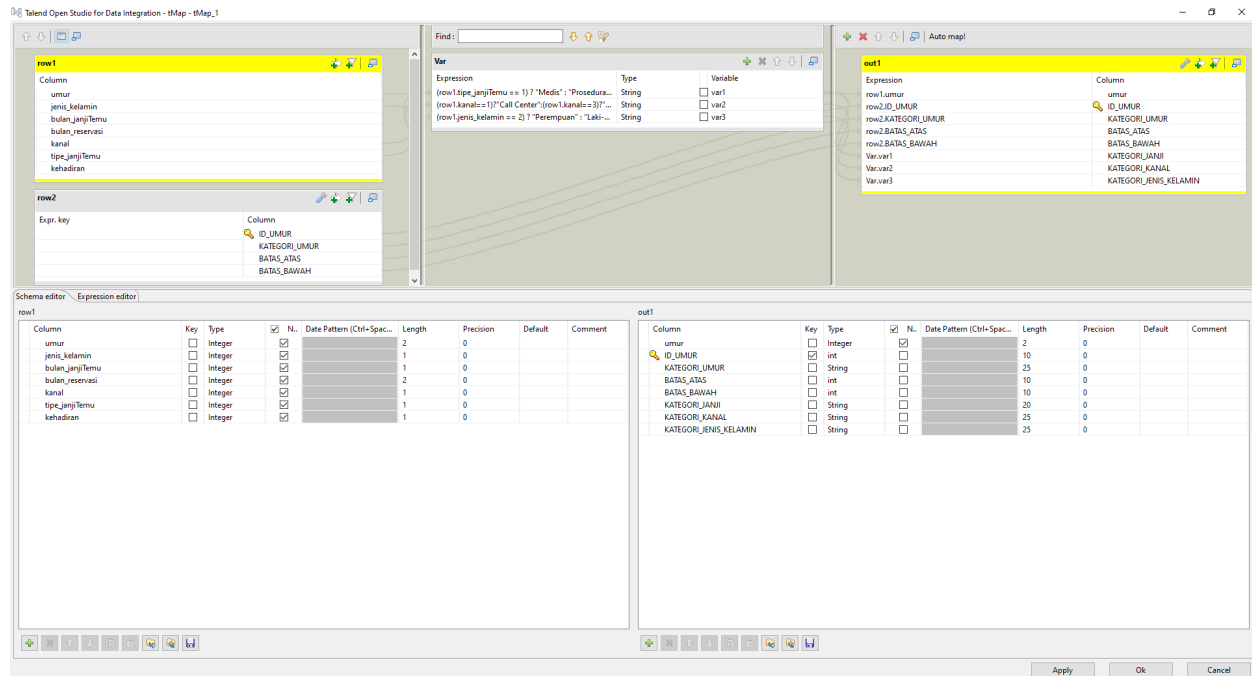
	ID_TPE_JANJI	KATEGORI_JANJI
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Medis
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Prosedural

e. Job **FACT**

- 1) Melakukan drag and drop metadata yang sudah dibuat ke dalam workspace Job kemudian menghubungkannya ke **tMap\_1** sebagai Main dan **tDBInput** untuk tabel umur sebagai Lookup.



- 2) Membuka **tMap\_1** dan buat output **out1** dengan mengatur Map Editor sebagai berikut:



Expression:

```
(row3.jenis_kelamin == 2) ? "Perempuan" : "Laki-Laki"
(row2.kanal == 1) ? "Call Center" : (row2.kanal == 2) ? "Personal" : "Web"
(row1.tipe_janji == 1) ? "Medis" : "Prosedural"
```

### 3) Menghubungkan **out1** dengan **tFilterRow\_1**.

Schema: Built-In Edit schema Sync columns

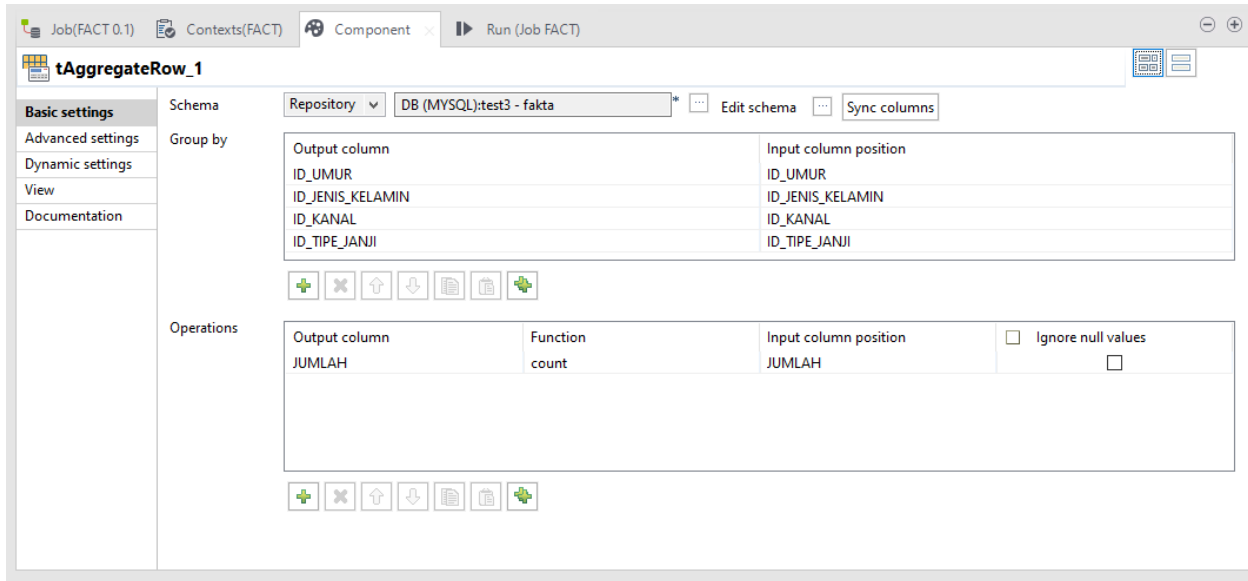
Logical operator used to combine conditions: And

Conditions	InputColumn	Function	Operator	Value
	umur	Empty	Lower or equal to	out1.BATAS_ATAS
	umur	Empty	Greater or equal to	out1.BATAS_BAWAH

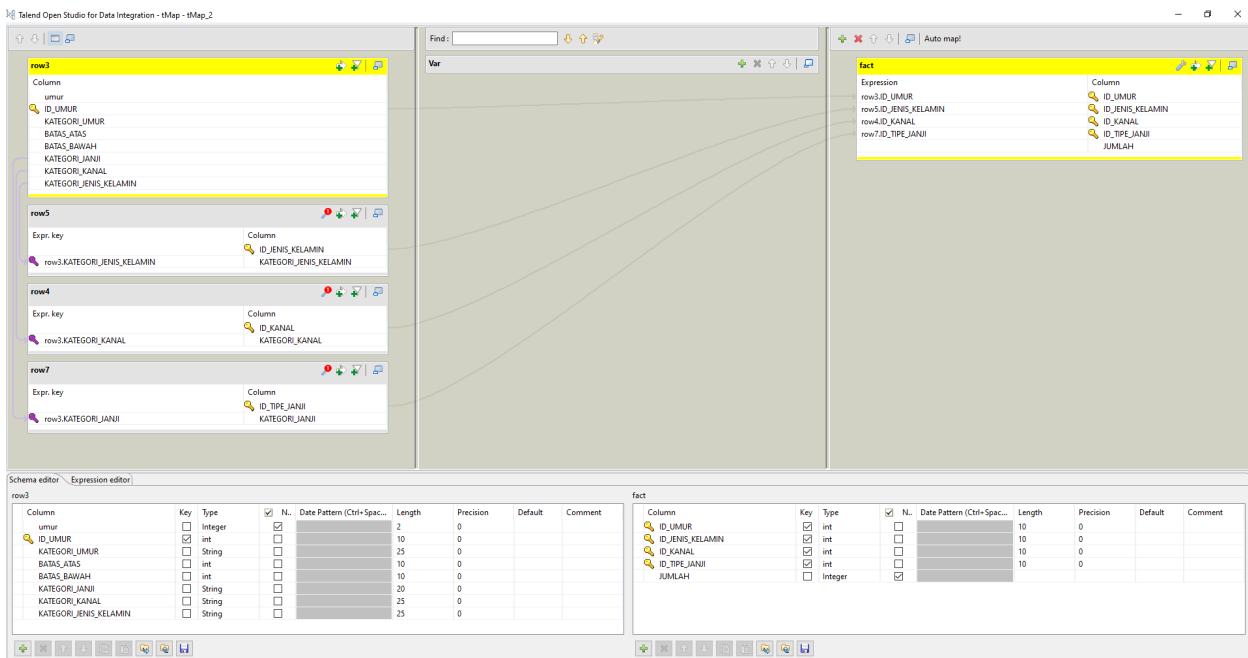
Use advanced mode

4) Menghubungkan **fakta** dan **tAggregateRow**, menyamakan skemanya kemudian menghubungkan **tFilterRow\_1** ke **tMap\_2** dengan Lookup tabel dimensi.

5) Mengatur **tAggregateRow** dengan pengaturan sebagai berikut:






6) Melakukan pengaturan pada **tMap\_2** sebagai berikut:



7) Melakukan run pada Job kemudian data akan masuk ke dalam basis data.

Execution

```
Starting job FACT at 17:53 27/05/2023.  
[statistics] connecting to socket on port 3955  
[statistics] connected  
[statistics] disconnected  
  
Job FACT ended at 17:53 27/05/2023. [Exit code = 0]
```

ID_UMUR	ID_JENIS_KELAMIN	ID_KANAL	ID_TIPE_JANJI	JUMLAH
1	2	1	1	4137
1	2	1	2	845
5	2	2	1	137
2	2	2	2	404
5	2	2	2	15
2	2	2	1	1651
3	1	1	2	123
2	2	3	1	7269
4	2	1	1	2462
3	1	2	2	44
3	1	2	1	476
4	1	3	1	1315
5	2	3	1	1622
1	2	2	2	334
5	1	1	1	1857
5	1	1	2	147
2	1	1	2	777
2	1	1	1	3506
1	2	2	1	957
4	2	1	2	207
2	1	2	1	1302
	2	2	2	65

Console

## FP04-SID: Capaian Proyek Penyusunan SQL OLAP

### A. Rank

Seorang user ingin menampilkan data jumlah pertemuan dan kategori jenis kelamin dengan kategori umur 'dewasa' dan kategori janji 'medis'. Ditampilkan dengan rangking yang berjarak pada jumlah (dari terbesar ke terkecil)

```
SELECT RANK() over(order by jumlah desc) AS rangking,  
jk.KATEGORI_JENIS_KELAMIN, u.KATEGORI_UMUR, tj.KATEGORI_JANJI,  
k.KATEGORI_KANAL, jumlah  
FROM jenis_kelamin as jk, kanal as k, tipe_janji as tj, umur as u, fakta as f  
WHERE jk.ID_JENIS_KELAMIN = f.ID_JENIS_KELAMIN  
AND k.ID_KANAL = f.ID_KANAL  
AND tj.ID_TIPE_JANJI = f.ID_TIPE_JANJI  
AND u.ID_UMUR = f.ID_UMUR  
AND u.KATEGORI_UMUR = "Dewasa"  
AND tj.KATEGORI_JANJI = "Medis"
```

rangking	KATEGORI_JENIS_KELAMIN	KATEGORI_UMUR	KATEGORI_JANJI	KATEGORI_KANAL	jumlah
1	Perempuan	Dewasa	Medis	Web	7269
2	Perempuan	Dewasa	Medis	Call Center	6950
3	Laki-Laki	Dewasa	Medis	Call Center	3506
4	Laki-Laki	Dewasa	Medis	Web	3075
5	Perempuan	Dewasa	Medis	Personal	1651
6	Laki-Laki	Dewasa	Medis	Personal	1302

### B. Row\_Number

Seorang user ingin menampilkan penomoran untuk setiap baris data yang akan dimulai dari 1 lagi apabila berbeda kategori umur, untuk kategori janji medis, kategori jenis kelamin 'perempuan', dan untuk seluruh kategori kanal

```
SELECT ROW_NUMBER() OVER(PARTITION BY u.KATEGORI_UMUR order by  
jumlah desc) AS NO, u.KATEGORI_UMUR, jk.KATEGORI_JENIS_KELAMIN,  
tj.KATEGORI_JANJI, k.KATEGORI_KANAL, jumlah  
FROM jenis_kelamin as jk, kanal as k, tipe_janji as tj, umur as u, fakta as f  
WHERE jk.ID_JENIS_KELAMIN = f.ID_JENIS_KELAMIN  
AND k.ID_KANAL = f.ID_KANAL
```

```

AND tj.ID_TIPE_JANJI = f.ID_TIPE_JANJI AND u.ID_UMUR = f.ID_UMUR
AND tj.KATEGORI_JANJI = "Medis"
AND jk.KATEGORI_JENIS_KELAMIN = "Perempuan";

```

NO	KATEGORI_UMUR	KATEGORI_JENIS_KELAMIN	KATEGORI_JANJI	KATEGORI_KANAL	jumlah
1	Anak-Anak	Perempuan	Medis	Call Center	2462
2	Anak-Anak	Perempuan	Medis	Web	1435
3	Anak-Anak	Perempuan	Medis	Personal	280
1	Balita	Perempuan	Medis	Call Center	1639
2	Balita	Perempuan	Medis	Web	1622
3	Balita	Perempuan	Medis	Personal	137
1	Dewasa	Perempuan	Medis	Web	7269
2	Dewasa	Perempuan	Medis	Call Center	6950
3	Dewasa	Perempuan	Medis	Personal	1651
1	Lansia	Perempuan	Medis	Call Center	4137
2	Lansia	Perempuan	Medis	Web	1273
3	Lansia	Perempuan	Medis	Personal	957
1	Remaja	Perempuan	Medis	Call Center	1694
2	Remaja	Perempuan	Medis	Web	1239
3	Remaja	Perempuan	Medis	Personal	427

### C. Rollup

Seorang user ingin menampilkan jumlah agregasi dari jumlah pertemuan yang dikelompokkan untuk setiap kategori kanal, kategori janji, kategori umur 'dewasa' dan untuk kategori jenis kelamin 'perempuan'.

```

SELECT KATEGORI_KANAL, KATEGORI_JANJI, KATEGORI_JENIS_KELAMIN,
SUM(jumlah) as jumlah_total
FROM jenis_kelamin as jk, kanal as k, tipe_janji as tj, umur as u, fakta as f
WHERE jk.ID_JENIS_KELAMIN = f.ID_JENIS_KELAMIN
AND k.ID_KANAL = f.ID_KANAL
AND tj.ID_TIPE_JANJI = f.ID_TIPE_JANJI
AND u.ID_UMUR = f.ID_UMUR
AND u.KATEGORI_UMUR = "Dewasa"
AND jk.KATEGORI_JENIS_KELAMIN = "Perempuan"
GROUP BY k.KATEGORI_KANAL,tj.KATEGORI_JANJI,
jk.KATEGORI_JENIS_KELAMIN WITH ROLLUP;

```

KATEGORI_KANAL	KATEGORI_JANJI	KATEGORI_JENIS_KELAMIN	jumlah_total
Call Center	Medis	Perempuan	6950
Call Center	Medis	NULL	6950
Call Center	Prosedural	Perempuan	1156
Call Center	Prosedural	NULL	1156
Call Center	NULL	NULL	8106
Personal	Medis	Perempuan	1651
Personal	Medis	NULL	1651
Personal	Prosedural	Perempuan	404
Personal	Prosedural	NULL	404
Personal	NULL	NULL	2055
Web	Medis	Perempuan	7269
Web	Medis	NULL	7269
Web	NULL	NULL	7269
NULL	NULL	NULL	17430

#### D. Cube

Seorang user ingin menampilkan total jumlah pertemuan untuk setiap jenis kelamin, total jumlah pertemuan untuk setiap kategori umur, serta agregasi total pertemuan untuk semua kategori umur dengan syarat kategori umur adalah Anak-anak dan Remaja, pada Kategori Kanal Web dan Kategori Janji Medis .

Note : Sebagai pengganti fungsi Cube agar dapat dijalankan di MySQL, maka digunakan pendekatan Union agar dapat menghasilkan output yang mendekati hingga setara dengan fungsi Cube.

```

SELECT KATEGORI_UMUR, KATEGORI_JENIS_KELAMIN, SUM(jumlah) as jumlah
FROM (
  SELECT KATEGORI_UMUR, KATEGORI_JENIS_KELAMIN, SUM(jumlah) as
jumlah
  FROM jenis_kelamin as jk, kanal as k, tipe_janji as tj, umur as u, fakta as f
  WHERE jk.ID_JENIS_KELAMIN = f.ID_JENIS_KELAMIN
  AND k.ID_KANAL = f.ID_KANAL
  AND u.ID_UMUR = f.ID_UMUR
  AND tj.ID_TIPE_JANJI = f.ID_TIPE_JANJI
  AND tj.KATEGORI_JANJI = 'Medis'

```

```

AND k.KATEGORI_KANAL = 'Web'
AND (u.KATEGORI_UMUR = 'Remaja' OR u.KATEGORI_UMUR = 'Anak-anak')
GROUP BY KATEGORI_UMUR, KATEGORI_JENIS_KELAMIN

UNION

SELECT NULL, KATEGORI_JENIS_KELAMIN, SUM(jumlah) as jumlah
FROM jenis_kelamin as jk, kanal as k, tipe_janji as tj, umur as u, fakta as f
WHERE jk.ID_JENIS_KELAMIN = f.ID_JENIS_KELAMIN
AND k.ID_KANAL = f.ID_KANAL
AND u.ID_UMUR = f.ID_UMUR
AND tj.ID_TIPE_JANJI = f.ID_TIPE_JANJI
AND tj.KATEGORI_JANJI = 'Medis'
AND k.KATEGORI_KANAL = 'Web'
AND (u.KATEGORI_UMUR = 'Remaja' OR u.KATEGORI_UMUR = 'Anak-anak')
GROUP BY KATEGORI_JENIS_KELAMIN

UNION

SELECT KATEGORI_UMUR, NULL, SUM(jumlah) as jumlah
FROM jenis_kelamin as jk, kanal as k, tipe_janji as tj, umur as u, fakta as f
WHERE jk.ID_JENIS_KELAMIN = f.ID_JENIS_KELAMIN
AND k.ID_KANAL = f.ID_KANAL
AND u.ID_UMUR = f.ID_UMUR
AND tj.ID_TIPE_JANJI = f.ID_TIPE_JANJI
AND tj.KATEGORI_JANJI = 'Medis'
AND k.KATEGORI_KANAL = 'Web'
AND (u.KATEGORI_UMUR = 'Remaja' OR u.KATEGORI_UMUR = 'Anak-anak')
GROUP BY KATEGORI_UMUR

UNION

SELECT NULL, NULL, SUM(jumlah) as jumlah
FROM jenis_kelamin as jk, kanal as k, tipe_janji as tj, umur as u, fakta as f
WHERE jk.ID_JENIS_KELAMIN = f.ID_JENIS_KELAMIN
AND k.ID_KANAL = f.ID_KANAL
AND u.ID_UMUR = f.ID_UMUR
AND tj.ID_TIPE_JANJI = f.ID_TIPE_JANJI
AND tj.KATEGORI_JANJI = 'Medis'
AND k.KATEGORI_KANAL = 'Web'
AND (u.KATEGORI_UMUR = 'Remaja' OR u.KATEGORI_UMUR = 'Anak-anak')
) AS subquery

```



GROUP BY KATEGORI\_UMUR, KATEGORI\_JENIS\_KELAMIN  
ORDER BY jumlah DESC;

KATEGORI_UMUR	KATEGORI_JENIS_KELAMIN	jumlah ▾ 1
NULL	NULL	4522
Anak-Anak	NULL	2750
NULL	Perempuan	2674
NULL	Laki-Laki	1848
Remaja	NULL	1772
Anak-Anak	Perempuan	1435
Anak-Anak	Laki-Laki	1315
Remaja	Perempuan	1239
Remaja	Laki-Laki	533