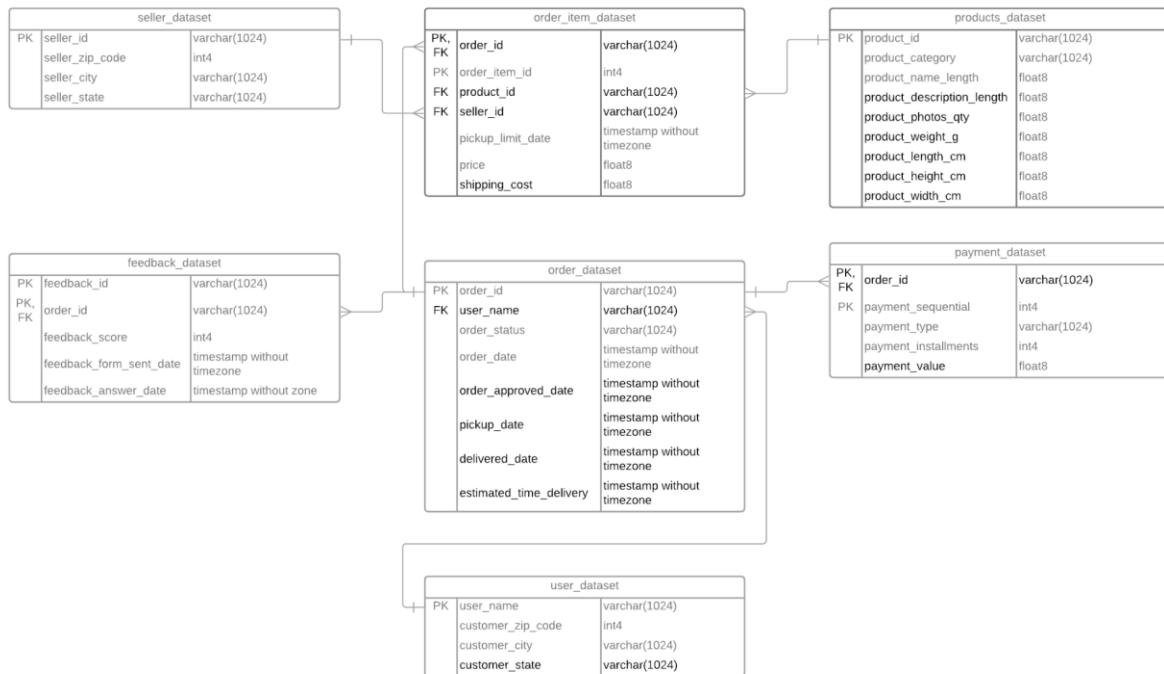


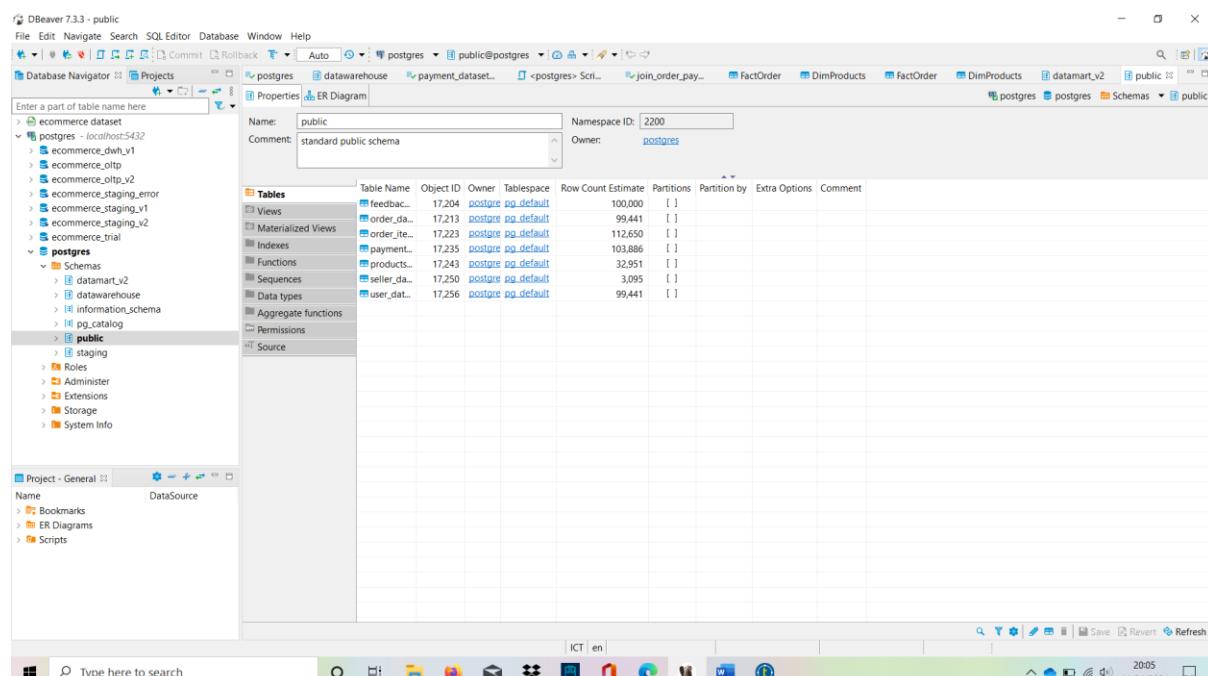
# PROGRESS 14 APRIL 2021

## OLTP

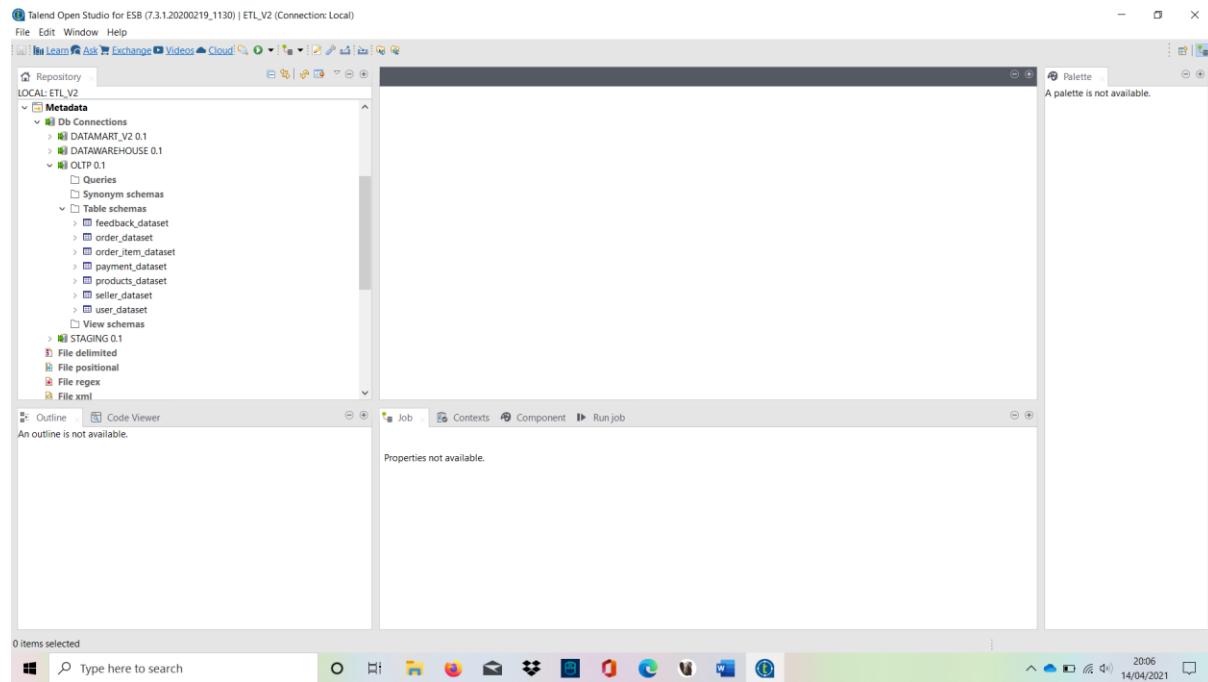
Untuk OLTP, saya langsung import data dari csv. Data tersebut antara lain feedback\_dataset, order\_dataset, order\_item\_dataset, payment\_dataset, seller\_dataset, user\_dataset, products\_dataset.



Pertama saya import dengan PostgreSQL pada db postgres, schema public.



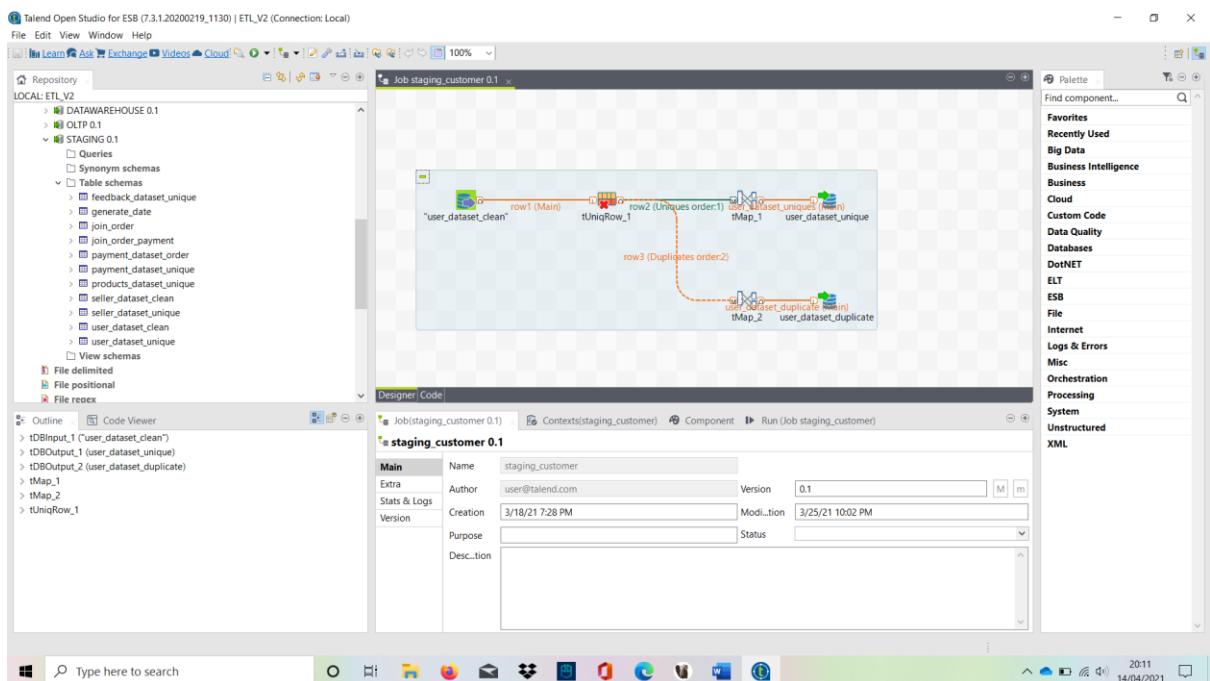
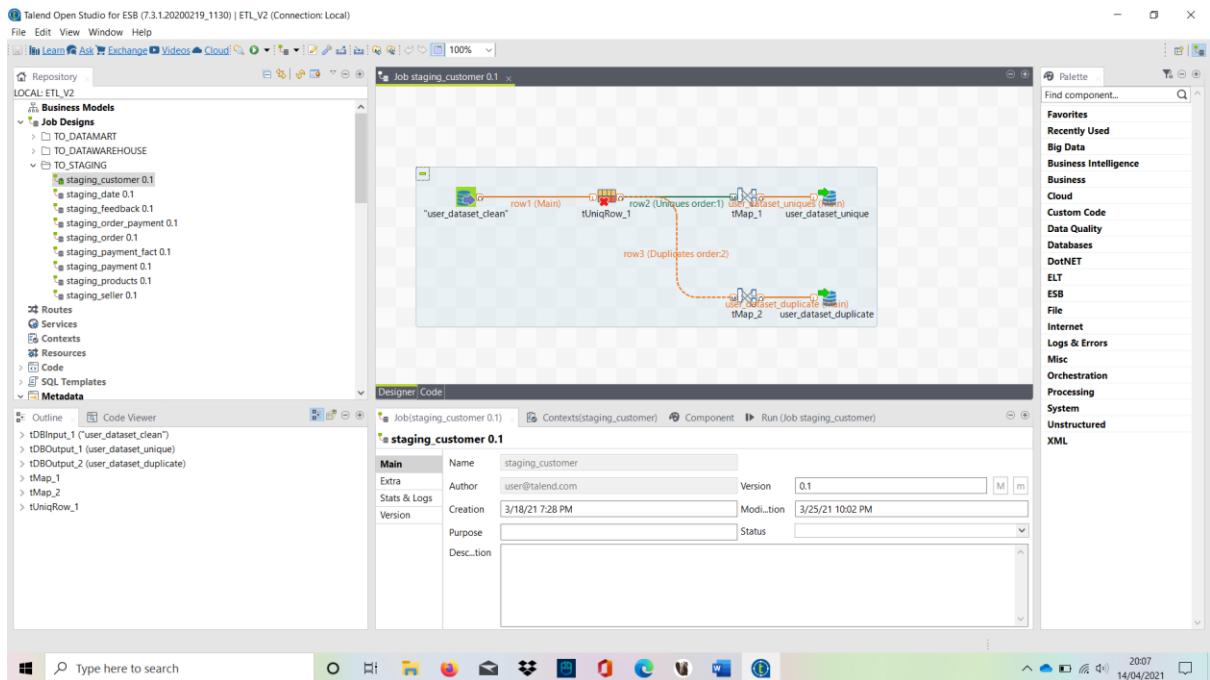
Kemudian saya lakukan db connection dengan talend.



Setelah itu saya melakukan cleaning terhadap user\_dataset dan seller\_dataset. Disini saya memastikan bahwa 1 zip code hanya merujuk pada satu kota. Disana saya lakukan cleaning di PostgreSQL.

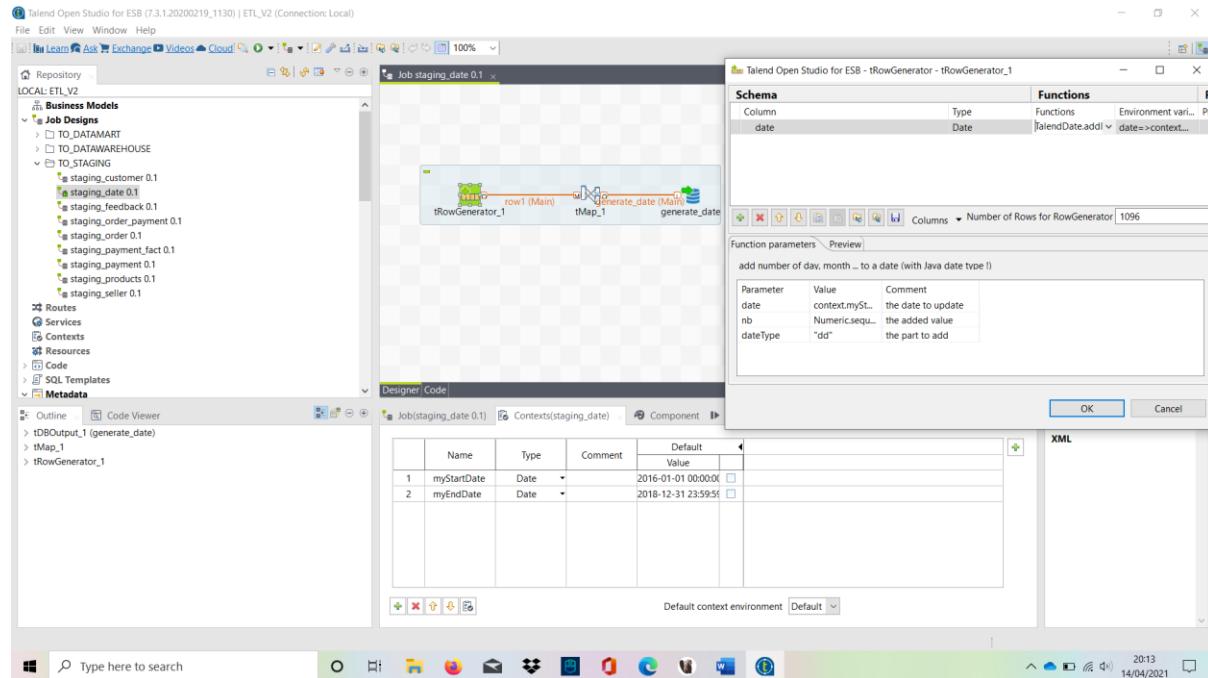
## STAGING

Untuk staging ada beberapa job yang dilakukan. Yang pertama ada job staging\_customer. Disana saya ingin memisahkan antara user\_dataset yang unique dengan yang duplicate. Dari staging (table user\_dataset\_clean yang ada pada staging) saya cari row yang unique untuk kemudian dimasukkan ke dalam staging. Sehingga baik yang unique ataupun duplicate masih ada.

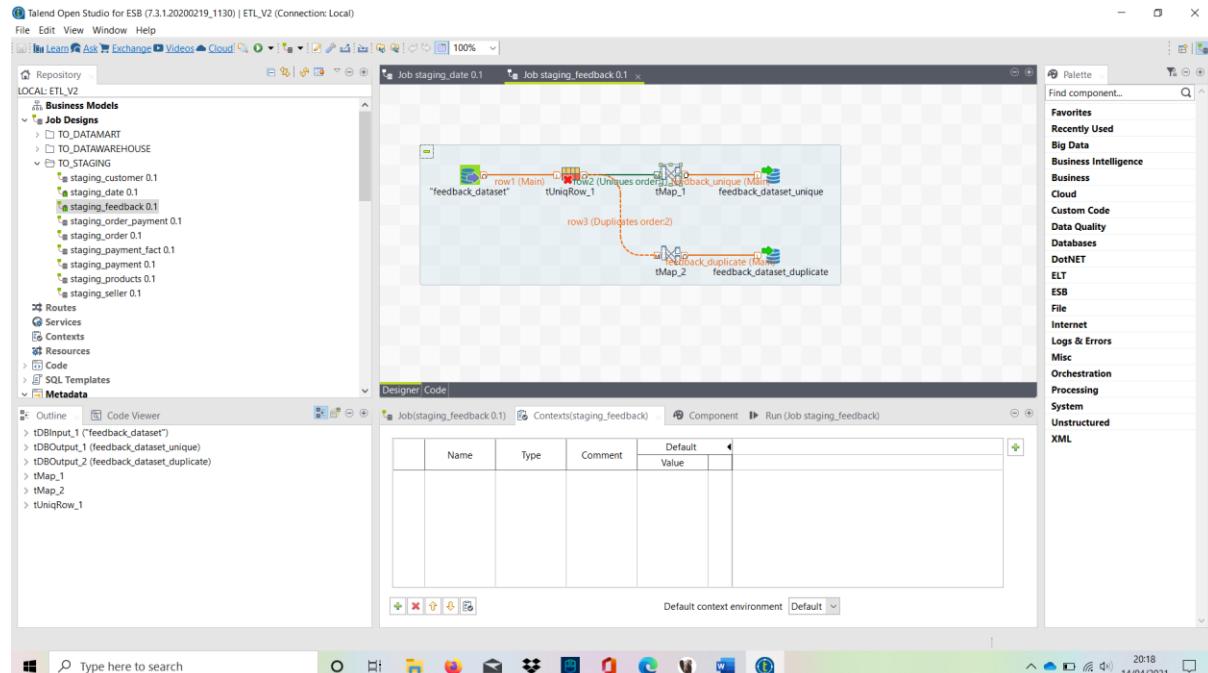


Kemudian yang kedua ada job `staging_date`. Disini saya membuat akan membuat dimensi date dengan Talend. Untuk start saya gunakan 1 Januari 2016 dan akhir yaitu 31 Desember 2018, berdasarkan min dan max date yang ada pada `order_date`. Kemudian menggunakan `tRowGenerator`, dimulai dari `context.myStartDate`, kemudian untuk urutan menggunakan `Numeric.sequence("s1",1,1) -1`, agar dari tanggal 2016-01-01 juga muncul. Kemudian date type adalah dd. Kemudian saya membuat berbagai macam kolom. Yang pertama ada `day_number_of_week`, yang merupakan minggu ke berapa pada tanggal tersebut (Week 1-4). Kemudian ada `day_number_of_month`, yang merupakan hari ke berapa pada suatu tanggal di bulan tersebut(1-31). Kemudian ada `day_number_of_year`, yang merupakan hari ke berapa pada suatu tanggal di tahun tersebut(1-366). `Month_year` merupakan bulan dalam bentuk string tanggal tersebut(January – December). Terdapat `month_number_of_year`, yaitu bulan ke berapa pada tanggal tersebut (1-12). `Calendar_quarter` merupakan kuarter ke berapa tanggal

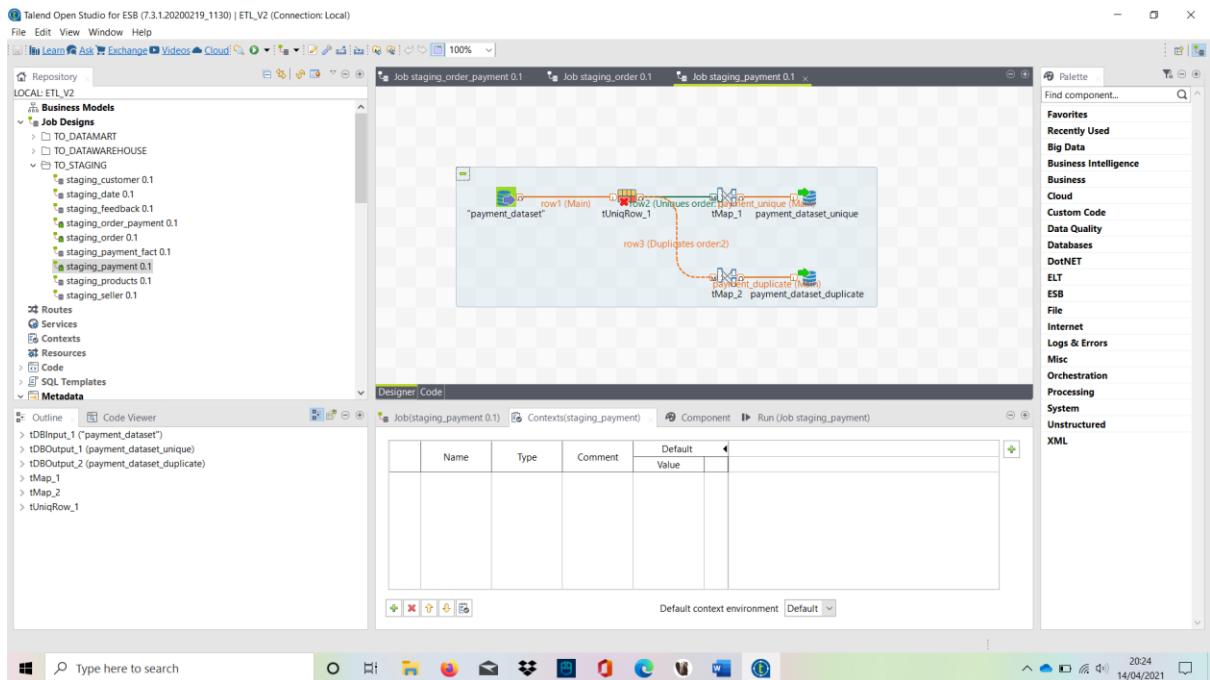
tersebut (1-4). Calendar\_year merupakan tahun berapa pada tanggal tersebut (2016-2018). Calendar\_semester merupakan semester ke berapa pada tanggal tersebut (1-2). Day name merupakan nama hari pada tanggal tersebut (Monday-Sunday).



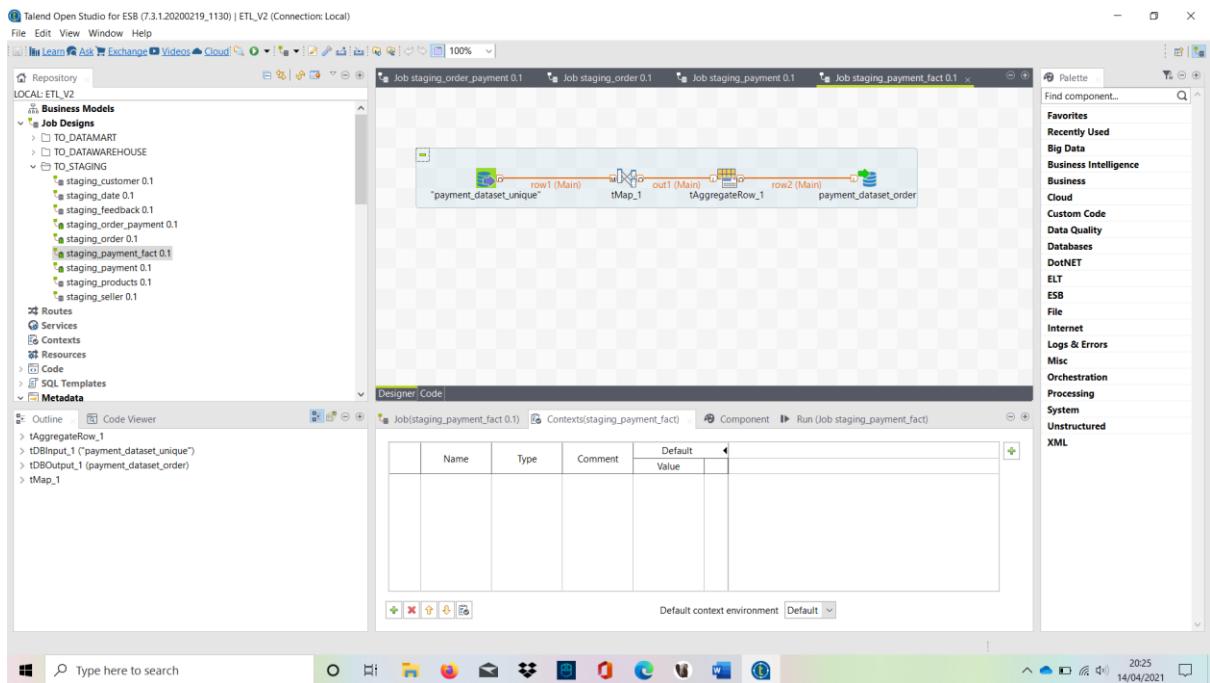
Staging\_feedback disini untuk memisahkan antara unique dan duplicate row yang ada pada feedback\_dataset.



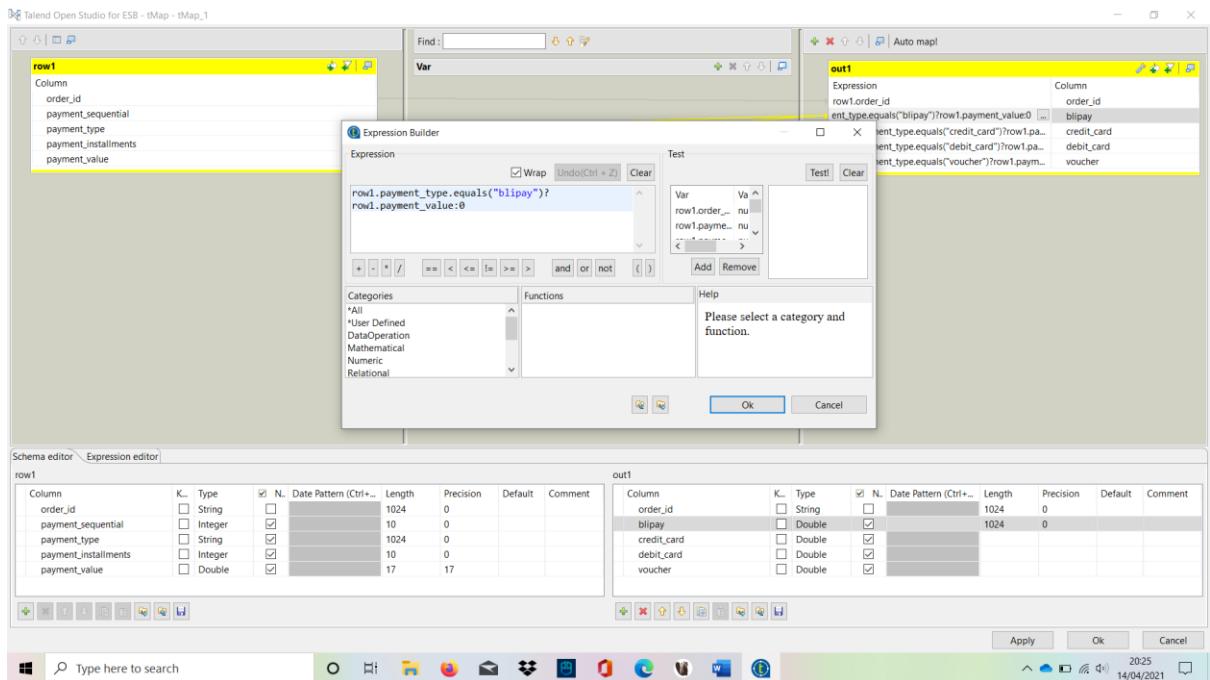
Staging\_payment disini saya memisahkan antara data payment yang duplicate dengan yang unique.



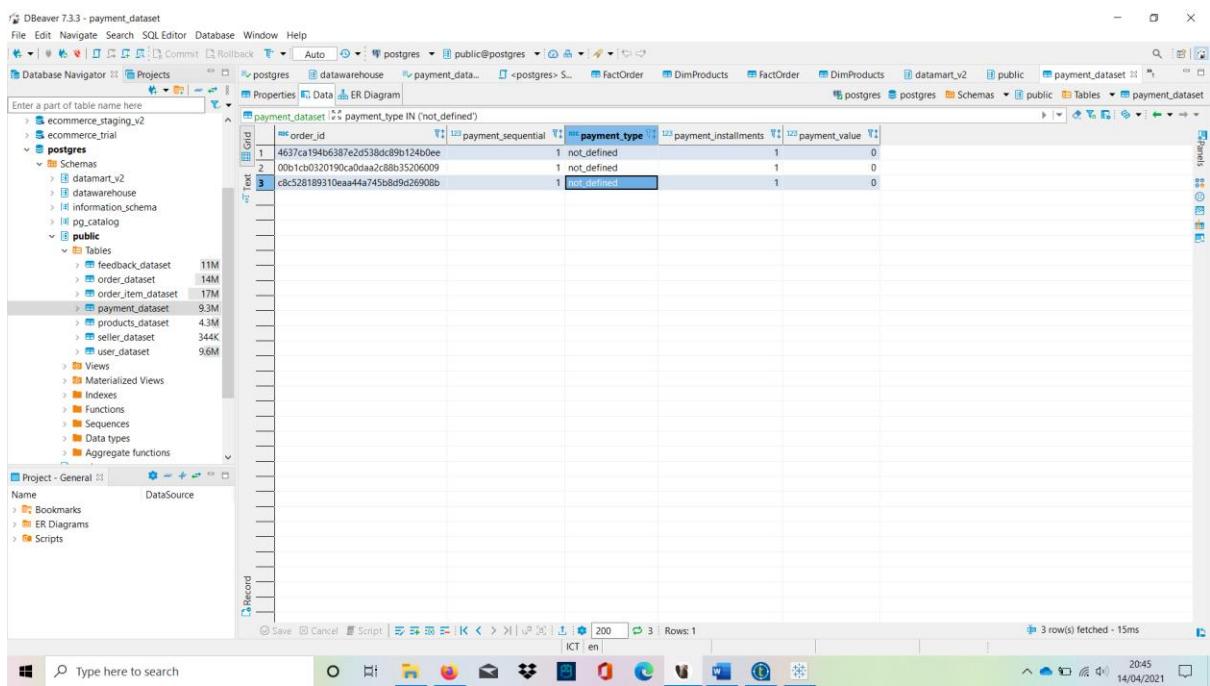
Staging\_payment\_fact, saya hanya menggunakan beberapa row saja yang akan menjadi pada lookup pada job staging\_order\_payment. Agar sesuai dengan granularity yang ada pada fact, maka saya sesuaikan.



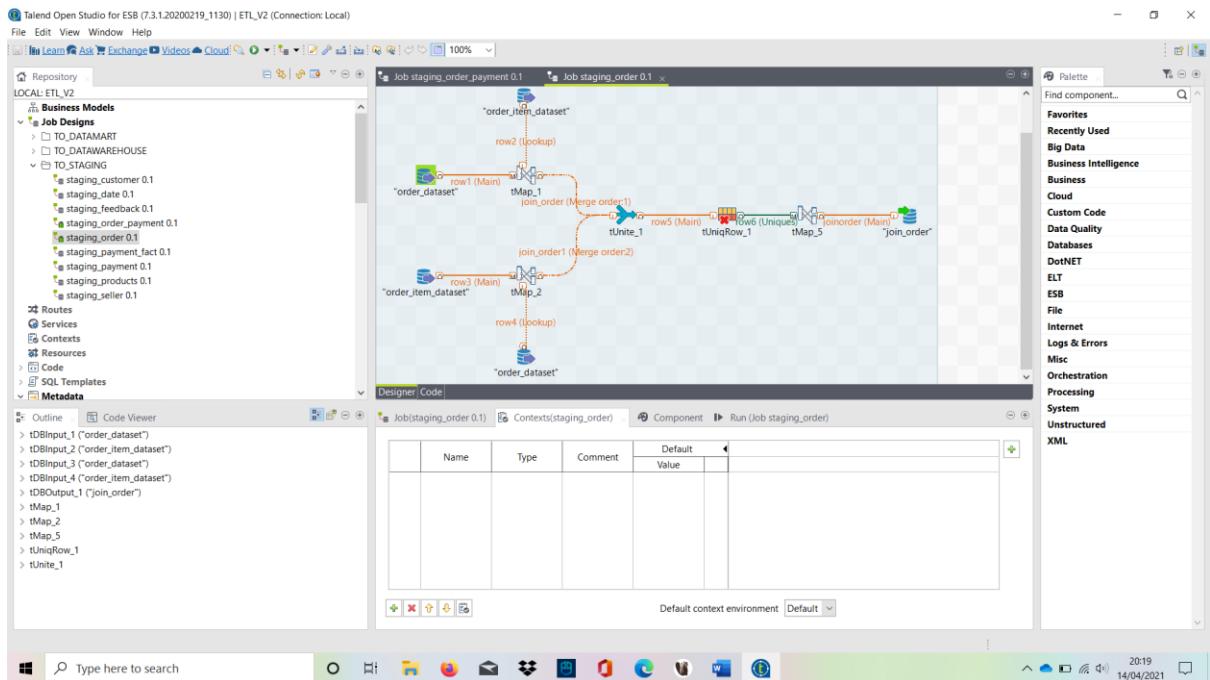
Disini terdapat 4 column yaitu blipay, credit\_card, debit\_card, dan voucher. Jika pembayarannya berupa blipay, maka nilai row adalah payment value. Setelah itu akan diaggregate dengan sum dan di group by menggunakan order\_id.



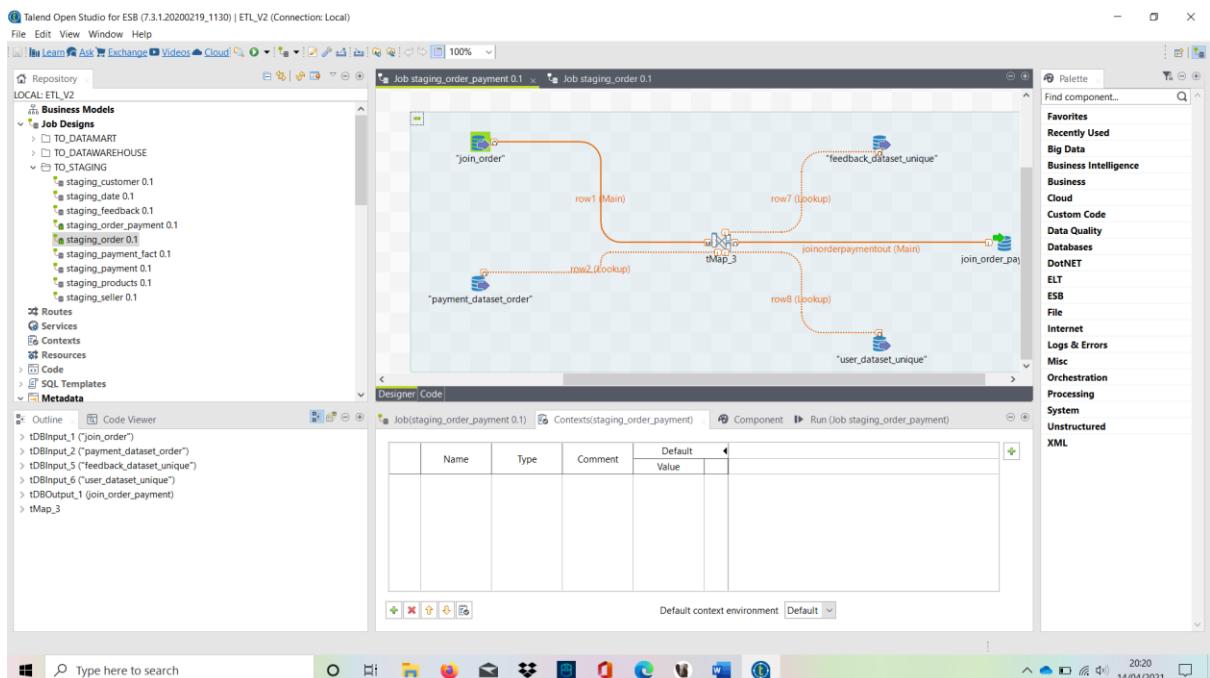
Sebenarnya ada satu tipe lagi yaitu not\_defined, namun karena hanya 3 dan tidak mengandung payment\_value, maka tidak dicantumkan.



Ada staging\_order untuk mengouter join order\_dataset dengan order\_item\_dataset karena saya ingin perorder item. Outputnya menuju ke staging dengan nama join\_order.

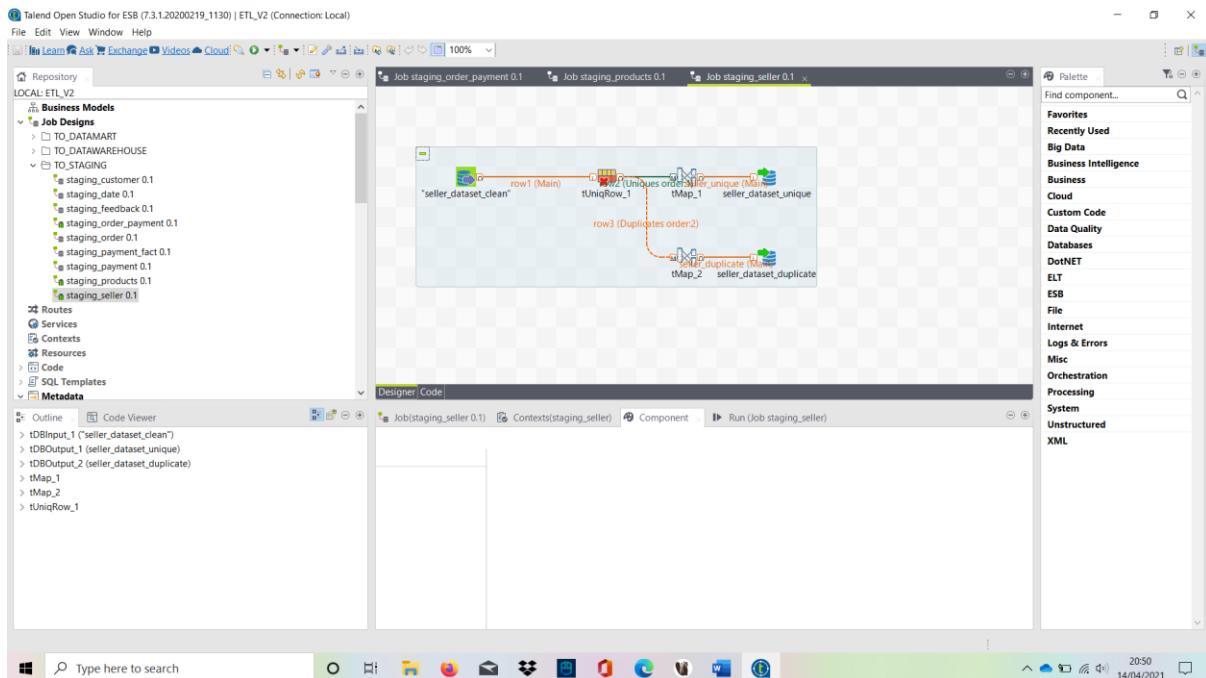


Job `staging_order_payment` dimana main disini adalah `join_order` dengan lookupnya adalah `payment_dataset_order`, `feedback_dataset_unique` dan `user_dataset_unique`. Outputnya ada pada `join_order_payment` di `staging`. Di `payment_dataset_order` diambil column `blipay`, `credit_card`, `debit_card`, dan `voucher`. Dari `feedback_dataset_unique` diambil `feedback_score`. Kemudian ada `user_dataset_unique`, diambil `user_name` dari table tersebut. Luarannya adalah `join_order_payment` yang masuk ke `staging`.

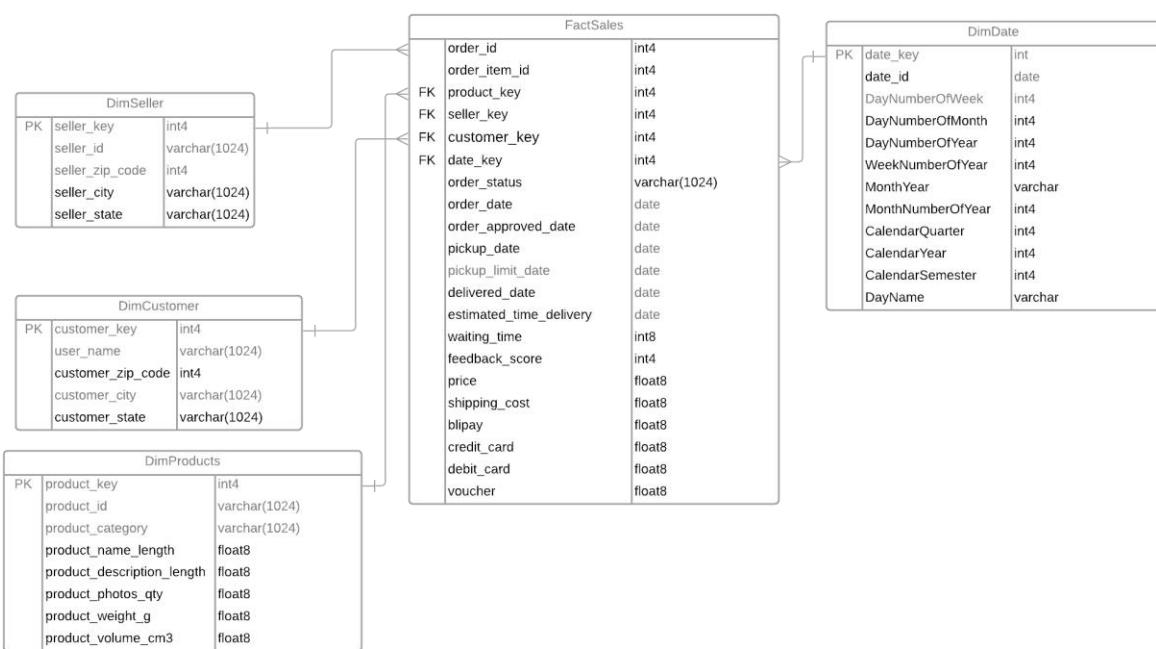


Kemudian terdapat job staging\_products untuk memisahkan data produk yang unique dan duplicate.

Terdapat job staging\_seller untuk memisahkan data produk yang unique dan duplicate.



## DATAWAREHOUSE



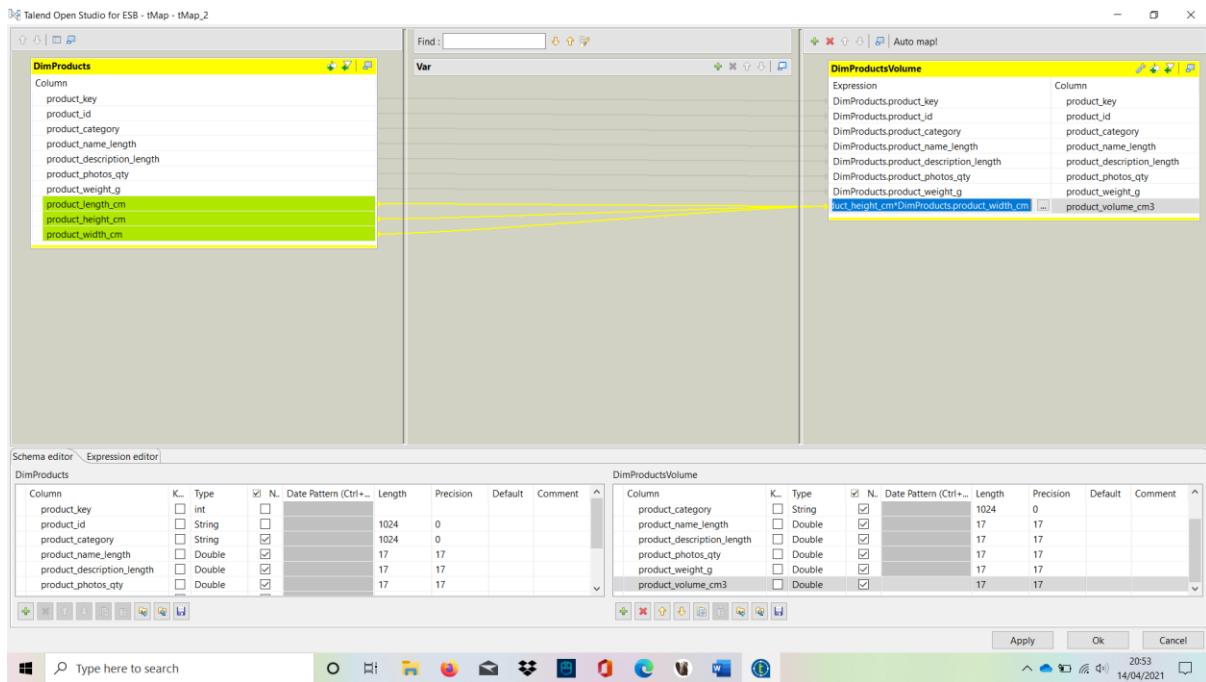
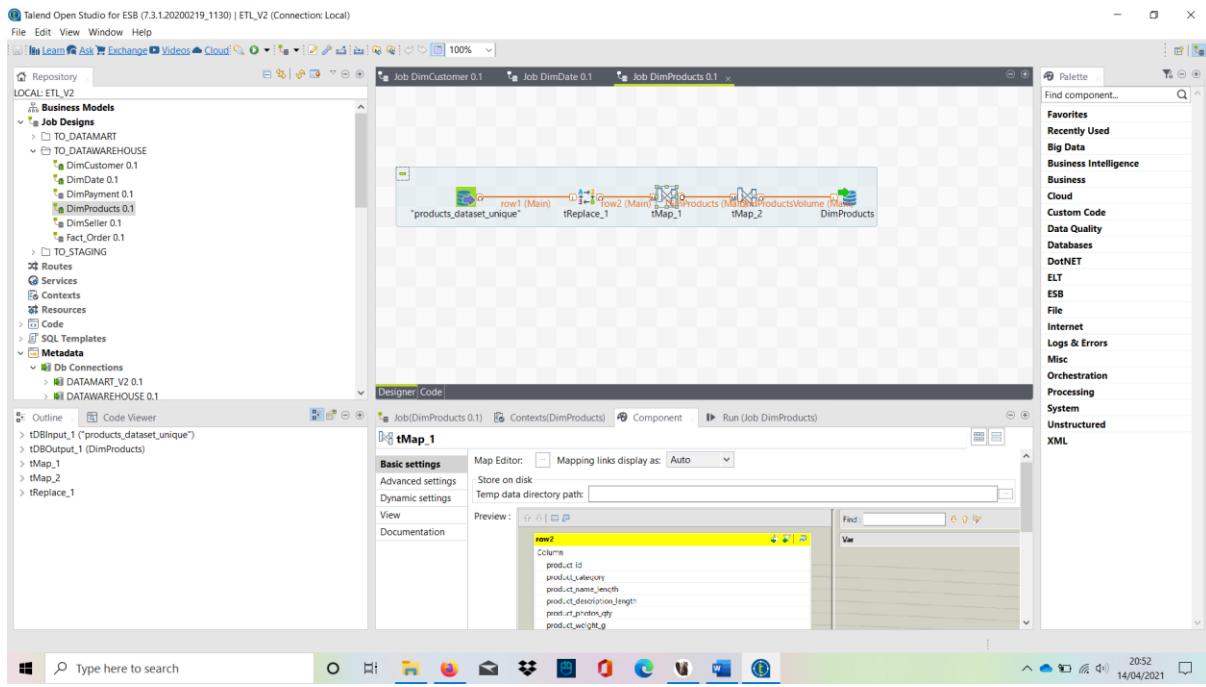
DimCustomer disini dari user\_dataset\_unique kemudian ditambah dengan surrogate key.

The screenshot shows the Talend Open Studio interface for a tMap component named 'tMap\_1'. The left panel displays the 'row1' dataset with columns: user\_name, customer\_zip\_code, customer\_city, and customer\_state. The right panel displays the 'DimCustomer' dataset with columns: customer\_key, user\_name, customer\_zip\_code, customer\_city, and customer\_state. The middle panel contains a 'Var' section where the expression 'Numeric.sequence("s1",1,1)' is mapped to the 'customer\_key' column. Below the panels are two schema editors. The first schema editor shows the mapping from 'row1' to 'DimCustomer' with specific data types and lengths. The second schema editor shows the detailed structure of the 'DimCustomer' table. At the bottom, a Windows taskbar is visible with various icons and the date/time '14/04/2021 20:51'.

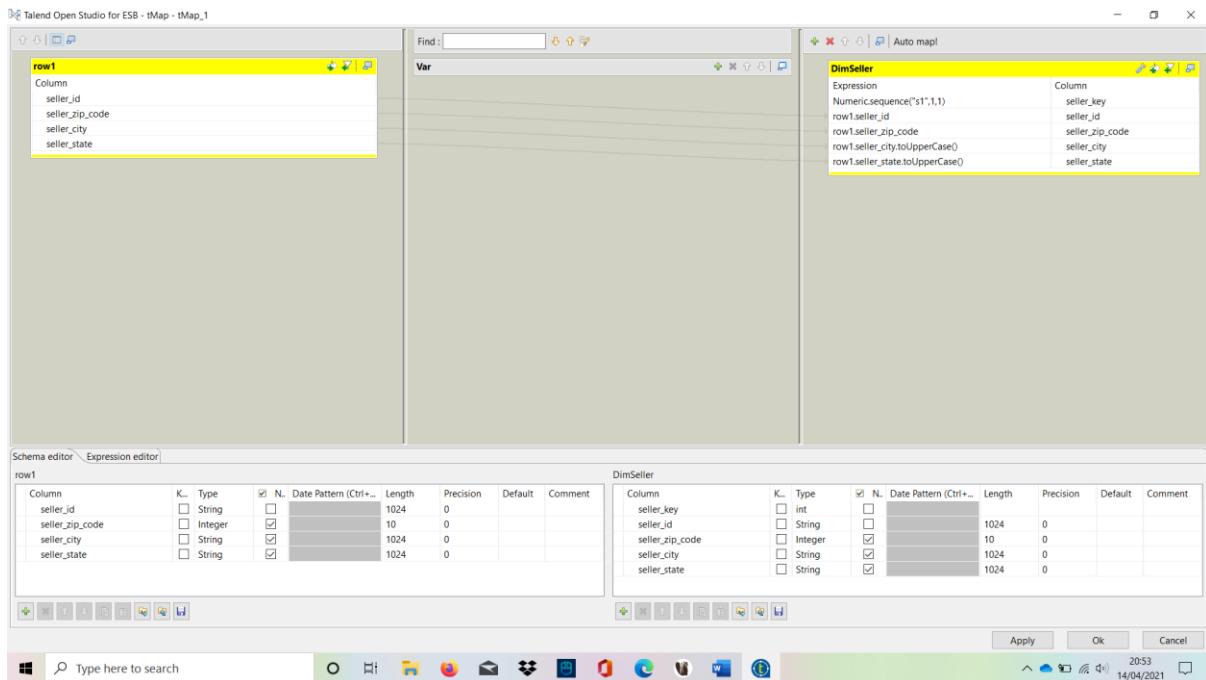
DimDate disini dari generate\_date kemudian ditambah dengan surrogate key.

This screenshot shows the same Talend Open Studio interface for the 'tMap\_1' component. The left panel shows the 'row1' dataset with various date-related columns like date\_id, DayNumberOfWeek, DayNumberOfMonth, DayNumberOfYear, WeekNumberOfYear, MonthYear, MonthNumberOfYear, CalendarQuarter, CalendarYear, CalendarSemester, and DayName. The right panel shows the 'DimDate' dataset with columns: date\_key, date\_id, DayNumberOfWeek, DayNumberOfMonth, DayNumberOfYear, WeekNumberOfYear, MonthYear, MonthNumberOfYear, CalendarQuarter, CalendarYear, CalendarSemester, and DayName. The middle panel shows the 'Var' section with the expression 'Numeric.sequence("s1",1,1)' mapped to the 'date\_key' column. Below the panels are two schema editors. The first schema editor maps 'row1' to 'DimDate' with specific data types and lengths. The second schema editor shows the detailed structure of the 'DimDate' table. The Windows taskbar at the bottom shows the date/time '14/04/2021 20:52'.

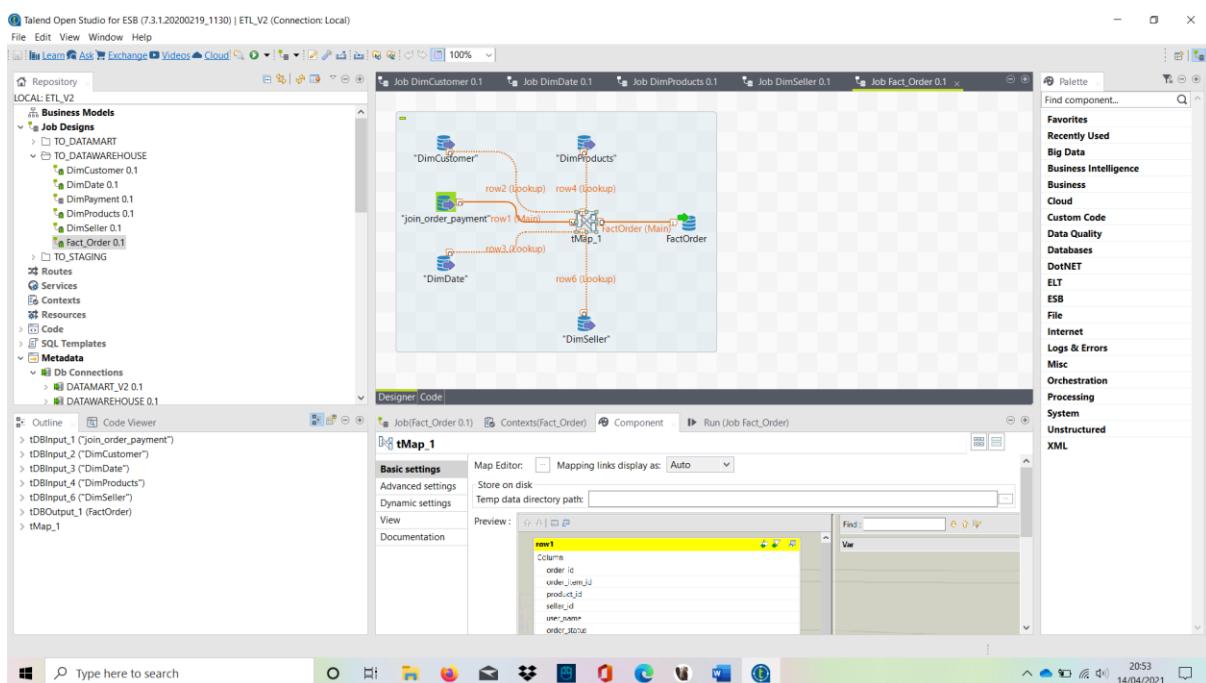
Dim Products disini dari products\_dataset\_unique kemudian ditambah dengan surrogate key pada tmap yang pertama dan menangkap null untuk diubah ke dalam 0. Pada tmap yang kedua, dibuat row baru yaitu volume yang merupakan product\_height\_cm dikali product\_length\_cm, dikali product\_width\_cm.



DimSeller disini dari seller\_dataset\_unique kemudian ditambah dengan surrogate key.



FactOrder, dimana join\_order\_payment sebagai main, dan DimCustomer, DimProducts, DimSeller, dan DimDate sebagai lookup.



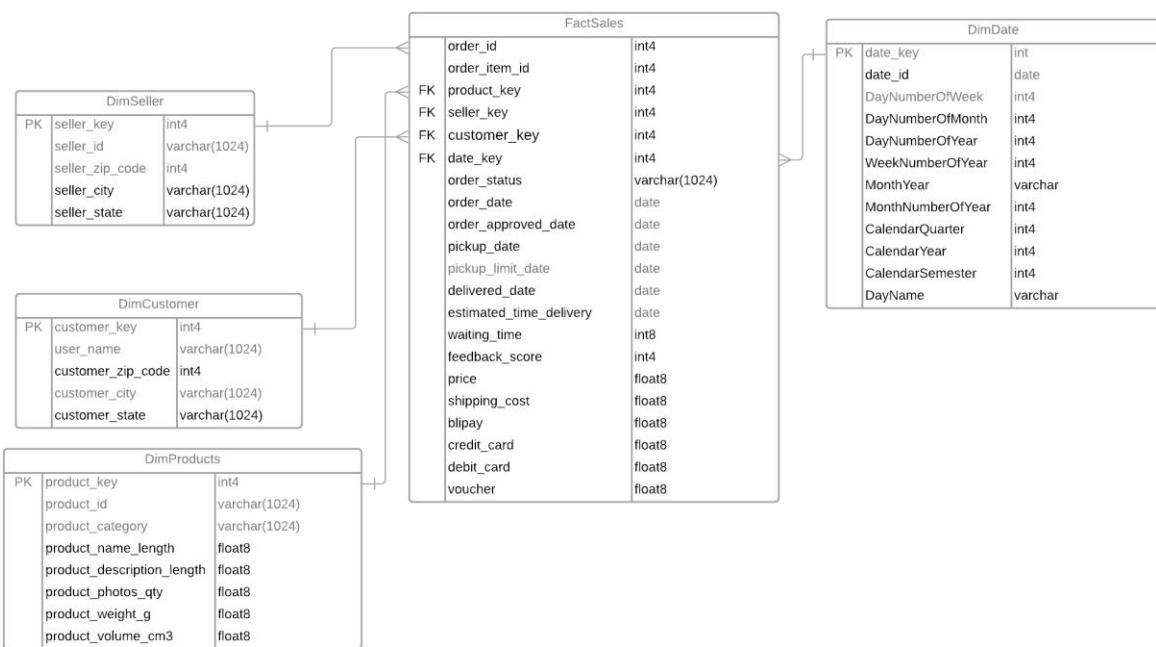
Disini untuk isinya sama dengan yang ada pada join\_order\_payment, namun yang berbeda adalah pada FactOrder, yang sebelumnya menggunakan natural key seperti product\_id, seller\_id, user\_name, diganti menggunakan surrogate key, yaitu product\_key, seller\_key, and customer\_key. Untuk date, order\_date yang ada pada join\_order\_payment dihubungkan dengan table DimDate pada column date\_id. Kemudian date\_key yang ada pada table DimDate dimasukkan ke dalam Fact Order.

The screenshot shows the Talend Open Studio interface for ESB, specifically the tMap component. It displays two rows of data mapping:

- Row 1:** Mappings from various columns like order\_id, order\_item\_id, product\_id, etc., to a single column named "customer\_key".
- Row 2:** Mappings from various columns to another set of columns, likely representing a fact table.

Below the rows, there are schema editors for both the source and target tables, showing detailed column definitions (e.g., type, length, precision) and constraints. The bottom of the window shows a Windows taskbar with icons for various applications.

## DATAMART



Untuk datamart sendiri tidak jauh dengan datawarehouse. Dan ini yang akan saya gunakan untuk visualisasi di Tableau.

