

**Strategi Meningkatkan Pendapatan dan Menstabilkan Bisnis Rental DVD Melalui  
Analisis Data**

Disusun guna memenuhi Tugas Final Project Studi Independen Celerates School

**Mentor:** Ryan Firmansyah



**Disusun oleh**

**Kelompok 3**

Ayu Nirmala

Berliani Salsabiilah

Muhammad Aulia Hana Putra

Rafie Ahza Ghaisan

**MAGANG DAN STUDI INDEPENDEN BERSERTIFIKAT  
CELERATES  
2024**

## **KATA PENGANTAR**

Puji syukur kami panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga makalah berjudul “Strategi Meningkatkan Pendapatan dan Menstabilkan Bisnis Rental DVD Melalui Analisis Data” dapat diselesaikan dengan baik. Kami mengucapkan terima kasih kepada Kak Ryan Firmansyah selaku mentor atas bimbingannya, Kak Putra Gema Nusa selaku trainer atas ilmu yang diberikan, serta seluruh anggota kelompok atas kerja sama dan dedikasinya selama penyusunan makalah ini. Kami menyadari bahwa makalah ini masih memiliki kekurangan, sehingga kritik dan saran yang membangun sangat kami harapkan.

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>DAFTAR ISI.....</b>	<b>3</b>
<b>DAFTAR GAMBAR.....</b>	<b>4</b>
<b>DAFTAR LAMPIRAN.....</b>	<b>5</b>
<b>ABSTRAK.....</b>	<b>6</b>
<b>BAB 1.....</b>	<b>7</b>
<b>PENDAHULUAN.....</b>	<b>7</b>
1.1 Latar Belakang.....	7
1.2 Tujuan Project.....	7
1.3 Manfaat Project.....	8
1.4 Lingkup Project.....	8
<b>BAB 2.....</b>	<b>10</b>
<b>METODOLOGI PROJECT.....</b>	<b>10</b>
2.1 Deskripsi Database.....	10
2.2 Tahapan Project.....	11
2.2.1 Input Database ke RDBMS (PostgreSQL).....	11
2.2.2 Pembuatan Staging Area.....	15
2.2.3 Desain Data Warehouse.....	21
2.2.4 Pembuatan Data Mart.....	24
2.2.5 Pembuatan Dashboard.....	25
2.3 Tools dan Teknologi.....	36
<b>BAB 3.....</b>	<b>39</b>
<b>IMPLEMENTASI DAN HASIL.....</b>	<b>39</b>
3.1 Input Database.....	39
3.2 Pembuatan Staging Area.....	39
3.3 Desain Data Warehouse.....	47
3.4 Pembuatan Data Mart.....	51
3.5 Visualisasi dan Analisis Dashboard.....	51
<b>BAB 4.....</b>	<b>58</b>
<b>PENUTUP.....</b>	<b>58</b>
4.1 Kesimpulan.....	58
4.2 Saran.....	58
<b>DAFTAR PUSTAKA.....</b>	<b>59</b>
<b>LAMPIRAN.....</b>	<b>60</b>

## DAFTAR GAMBAR

Gambar 2.1 Tabel ERD database DVD rental	12
Gambar 2.2 Create database DVD rental	14
Gambar 2.3 Kolom create-database	14
Gambar 2.4 Memberikan nama pada database DVD rental	14
Gambar 2.5 Restore database DVD rental	15
Gambar 2.6 Menambahkan file DVD rental untuk di restore	15
Gambar 2.7 Restore database	16
Gambar 2.8 Memastikan file yang di restore di Dbeaver	16
Gambar 2.9 Memilih Semua tabel di DVD rental	17
Gambar 2.10 Mengcopy DDL dari semua tabel	18
Gambar 2.11 Paste DDL yang sudah di copy ke script baru	18
Gambar 2.12 Memeriksa kerangka tabel yang sudah dibuat	19
Gambar 2.13 Proses pembuatan tabel log	20
Gambar 2.14 Transformasi list tabel staging	21
Gambar 2.15 Transformasi log tabel masuk	21
Gambar 2.16 Transformasi read data	22
Gambar 2.17 Job execute data	22
Gambar 2.18 Main job staging	23
Gambar 2.19 Entity Relation Diagram (ERD)	24
Gambar 2.20 Transformasi Dimensi Customer	24
Gambar 2.21 Transformasi Dimensi Date (waktu)	25
Gambar 2.22 Transformasi Dimensi Film	25
Gambar 2.23 Transformasi Dimensi Store	26
Gambar 2.24 Transformasi Fakta Rental	26
Gambar 2.25 Transformasi data mart	27
Gambar 2.26 Total Sales	28
Gambar 2.27 Total Customer	29
Gambar 2.28 Total Rentals	29
Gambar 2.29 Total Inventory	30
Gambar 2.30 Total Country	30
Gambar 2.31 Total Film	31
Gambar 2.32 Top 10 country by sales	31
Gambar 2.33 Bottom 10 country by sales	32
Gambar 2.34 Rentals by mouth	32
Gambar 2.35 Payment by month	33
Gambar 2.36 Film category by sales	33
Gambar 2.37 Sales by rating	34

Gambar 2.38 Customers by revenue	35
Gambar 2.39 Top actor by sales	35
Gambar 2.40 Bottom actor by sales	36
Gambar 2.41 Top film by sales	37
Gambar 2.42 Bottom film by sales	37
Gambar 2.43 Sales by staff	38
Gambar 2.44 Sales by day	38
Gambar 2.45 State	39
Gambar 2.46 Return status by distribution	39
Gambar 2.47 PostgreSQL	40
Gambar 2.48 DBeaver	40
Gambar 2.49 Pentaho	41
Gambar 2.50 Tableau	41
Gambar 3.1 Error last_updated	42
Gambar 3.2 Menghapus query table triggers	43
Gambar 3.3 Error public year	43
Gambar 3.4 Mengganti tipe public year	43
Gambar 3.5 Mengganti tipe public year	44
Gambar 3.6 Error public mpaa rating	44
Gambar 3.7 mengganti tipe data public mpaa rating	44
Gambar 3.8 Berhasil membuat kerangka tabel	45
Gambar 3.9 membuat schema log	45
Gambar 3.10 Error staging pentaho	46
Gambar 3.11Error staging pentaho	46
Gambar 3.12 Error staging pentaho	47
Gambar 3.13 Error staging pentaho	48
Gambar 3.14 Error staging pentaho	48
Gambar 3.15 Error berkurang saat foreign key dihapus	48
Gambar 3.16 Menghapus tabel views	49
Gambar 3.17 Uncekis district dan phone	49
Gambar 3.18 Menghapus Indexes	49
Gambar 3.19 Merunning query	50
Gambar 3.20 Terdapat null di kolom address	50
Gambar 3.21 mengatasi null	51
Gambar 3.22 Transformasi data mart	55
Gambar 3.23 Color Palette Visualisasi	56
Gambar 3.24 Desain Dashboard 1	57
Gambar 3.25 Desain Dashboard 2	58
Gambar 3.26 Desain Dashboard 3	58
Gambar 3.27 Dashboard 1	59
Gambar 3.28 Dashboard 2	60



## **DAFTAR LAMPIRAN**

Lampiran 1

65

## **ABSTRAK**

Industri persewaan DVD menghadapi tantangan dalam mempertahankan pendapatan yang stabil akibat perubahan preferensi pelanggan dan persaingan pasar yang semakin ketat. Analisis ini bertujuan untuk mengidentifikasi dan memahami faktor-faktor yang mempengaruhi fluktuasi pendapatan dengan pendekatan berbasis data menggunakan database DVD Rental. Melalui pembangunan infrastruktur data yang mencakup staging area, data warehouse, data mart, dan dashboard interaktif, analisis ini berhasil mengolah data dari berbagai dimensi, seperti pelanggan, genre film, lokasi, dan waktu. Hasil analisis memberikan wawasan strategis untuk memahami pola pendapatan, preferensi pelanggan, serta kinerja lokasi penyewaan.

Analisis ini menggunakan berbagai alat seperti PostgreSQL, Pentaho, DBeaver, dan Tableau untuk mendukung proses pengolahan dan visualisasi data. Dashboard interaktif yang dihasilkan memberikan gambaran menyeluruh tentang kinerja bisnis dan memudahkan pengambilan keputusan berbasis data. Hasil akhir analisis menunjukkan bahwa integrasi data yang baik dapat membantu bisnis mengoptimalkan layanan, meningkatkan pendapatan, dan mengurangi volatilitas pendapatan. Dengan beberapa pengembangan lebih lanjut, seperti penerapan model prediktif dan strategi pemasaran berbasis data, solusi ini dapat terus mendukung keberlanjutan bisnis persewaan DVD di masa depan.

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Industri persewaan DVD menghadapi tantangan besar dalam mempertahankan pendapatan yang stabil, terutama dalam menghadapi meningkatnya persaingan dan perubahan preferensi pelanggan. Salah satu permasalahan utama yang sering dihadapi masyarakat adalah fluktuasi pendapatan. Fluktuasi pendapatan dipengaruhi oleh berbagai faktor, termasuk musim, genre film populer dan loyalitas pelanggan.

Hal ini menciptakan kebutuhan mendesak bagi para manajer untuk memahami pola penjualan dan faktor-faktor yang mempengaruhinya sehingga mereka dapat mengembangkan strategi efektif untuk meningkatkan penjualan secara berkelanjutan. Untuk mengatasi permasalahan tersebut, pendekatan database merupakan solusi yang tepat. Memanfaatkan database DVD Rental di seluruh dimensi persewaan, pembayaran, pelanggan, dan waktu memberikan wawasan strategis yang mendalam bagi bisnis.

Database DVD Rental sangat penting untuk mendukung analisis bisnis. Data terstruktur ini memungkinkan administrator untuk melakukan berbagai analisis seperti: Identifikasi tren penjualan berdasarkan waktu, analisis profil pelanggan untuk memahami kontribusi mereka terhadap penjualan, memahami preferensi genre film pelanggan, dan banyak lagi. Hasil analisis ini dapat digunakan untuk menyusun strategi periklanan yang lebih efektif dan meningkatkan layanan pelanggan. Dengan mengintegrasikan data dari berbagai tabel ini ke dalam sistem gudang data, perusahaan dapat mengatasi tantangan volatilitas pendapatan dan memanfaatkan wawasan berbasis data untuk mengoptimalkan kinerja bisnis.

#### **1.2 Tujuan Project**

Project ini bertujuan untuk mengidentifikasi dan menganalisis faktor-faktor yang mempengaruhi fluktuasi pendapatan pada bisnis persewaan DVD. Dengan menggunakan pendekatan berbasis data, project ini juga bertujuan untuk:

1. Memahami pola pendapatan bisnis melalui analisis tren berdasarkan waktu, aktor, rating film, pelanggan, dan genre film.
2. Mengidentifikasi kontribusi utama terhadap pendapatan, termasuk pelanggan loyal, lokasi yang paling menguntungkan, serta genre film yang paling diminati.

### **1.3 Manfaat Project**

*Project* ini diharapkan memberikan manfaat langsung bagi bisnis persewaan DVD, terutama dalam meningkatkan pendapatan dan menjaga kestabilannya. Dengan menggunakan pendekatan database DVD Rental, bisnis dapat memperoleh wawasan strategis untuk mengidentifikasi peluang peningkatan pendapatan, seperti memahami preferensi pelanggan dan menentukan genre film yang paling diminati. Selain itu, project ini dapat membantu mengurangi fluktuasi pendapatan dengan strategi yang dirancang untuk menghadapi faktor-faktor musiman dan perubahan preferensi pelanggan.

### **1.4 Lingkup Project**

Project ini akan fokus pada pembangunan infrastruktur data yang mencakup pembuatan staging area, data warehouse, data mart, dan dashboard untuk analisis bisnis pada persewaan DVD. Adapun batasan proyek ini adalah sebagai berikut:

#### **1. Staging**

*Project* ini akan mencakup pembuatan staging area untuk menampung data mentah yang diperoleh dari berbagai sumber, seperti transaksi penyewaan dan pembayaran, data pelanggan, lokasi dan data film. Staging area akan berfungsi sebagai tempat sementara sebelum data diproses lebih lanjut ke dalam data warehouse.

#### **2. Data Warehouse**

Fokus utama *project* ini adalah mendesain dan membangun data warehouse yang menyimpan data terintegrasi dari berbagai dimensi customer, film, dan lain-lain. Data warehouse akan memfasilitasi proses analisis yang lebih efisien dan memungkinkan query yang lebih kompleks untuk mendukung pengambilan keputusan berbasis data.

#### **3. Datamart**

*Project* ini juga akan mencakup pembuatan data mart yang fokus pada area-area tertentu seperti analisis pendapatan, profil pelanggan, dan kinerja toko. Datamart ini akan membantu dalam proses analisis lebih spesifik dan terfokus pada kebutuhan bisnis tertentu.

#### **4. Dashboard**

Sebagai bagian dari *project*, akan dikembangkan dashboard interaktif menggunakan tableau untuk menampilkan hasil analisis yang dihasilkan dari data

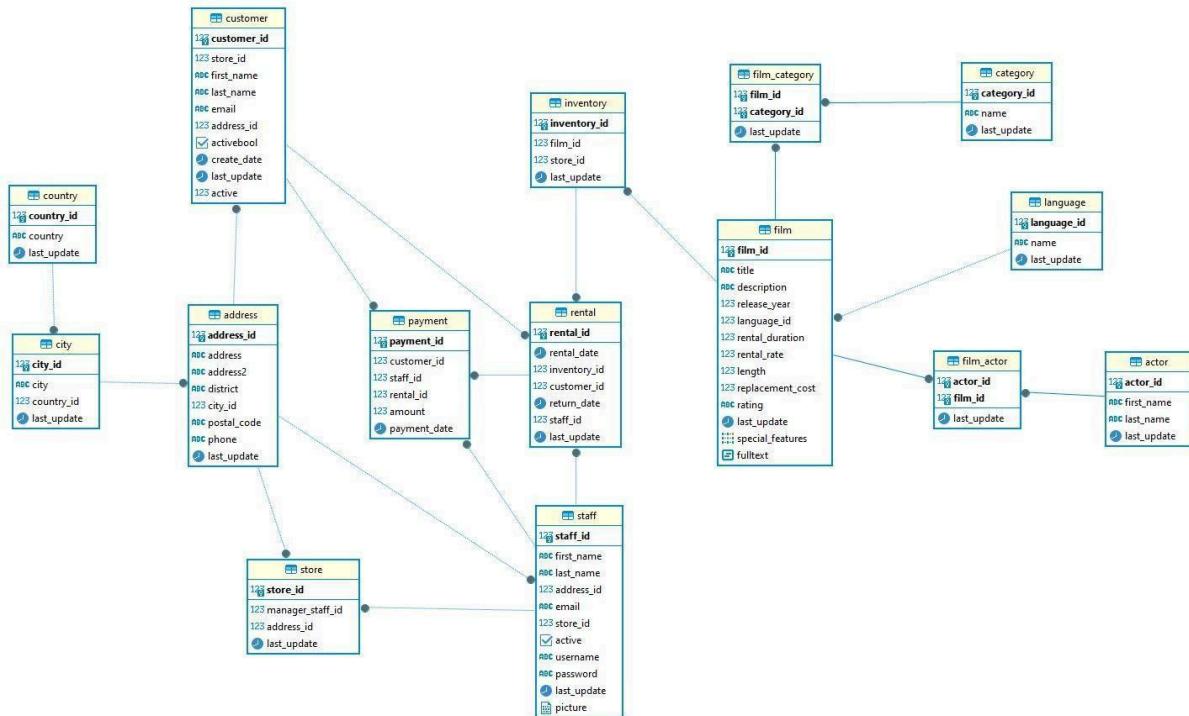
warehouse dan data mart. Dashboard ini akan menyajikan visualisasi data yang mudah dipahami, memungkinkan pemangku kepentingan untuk memantau kinerja bisnis secara real-time dan membuat keputusan yang lebih tepat.

## BAB 2

### METODO LOGI PROJEC T

#### 2.1 Deskripsi Database

Setelah berdiskusi via online menggunakan *discord*, kami memutuskan untuk memilih database DVD Rental. Menurut kami struktur tabel cukup sederhana dan tidak terlalu kompleks untuk dianalisis. Struktur database DVD Rental ini sendiri terdiri dari beberapa tabel utama yang saling terhubung melalui hubungan *primary key* (PK) dan *foreign key* (FK). Pada tabel film terdapat data utama tentang film, seperti judul, deskripsi, kategori, dan bahasa, serta berhubungan dengan tabel *film\_actor* untuk mencatat aktor yang membintangi film dan *film\_category* untuk menghubungkan film dengan kategorinya. Bahasa film disimpan tabel *language*, yang juga terhubung langsung ke tabel film.



Gambar 2.1 Tabel ERD database DVD rental

Data inventaris film dikelola melalui tabel `inventory`, yang menyimpan informasi mengenai stok film yang tersedia di berbagai lokasi store. Tabel ini berhubungan dengan tabel `store`, yang mencatat informasi tentang lokasi store, serta tabel `rental` yang mencatat transaksi penyewaan film. Tabel `rental` juga terhubung dengan tabel `customer` untuk

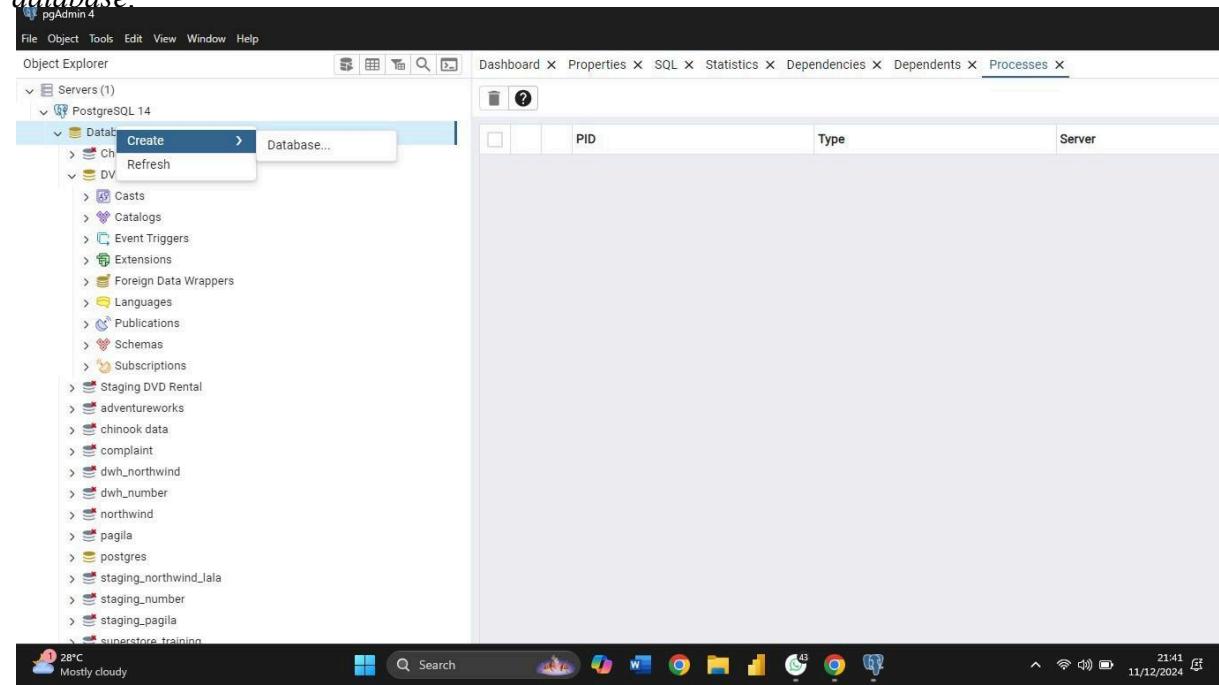
mengidentifikasi pelanggan yang melakukan penyewaan dan tabel payment untuk mencatat pembayaran terkait transaksi tersebut.

Hubungan mengenai lokasi berada di tabel address, city dan country. Tabel address menyimpan informasi alamat yang digunakan oleh tabel customer, staff, dan store. Tabel address ini terhubung dengan tabel city untuk mencatat kota tempat alamat, sementara tabel city terhubung ke tabel country untuk mencatat negara. Lebih lanjut lagi mengenai data pelanggan disimpan dalam tabel customer, termasuk nama, email, dan status keaktifan. Pelanggan dapat dihubungkan ke store tertentu melalui tabel store, yang juga terhubung dengan tabel staff untuk melihat staf yang bekerja di lokasi tersebut. Tabel staff menyimpan informasi staf, termasuk alamat dan store tempat mereka bekerja.

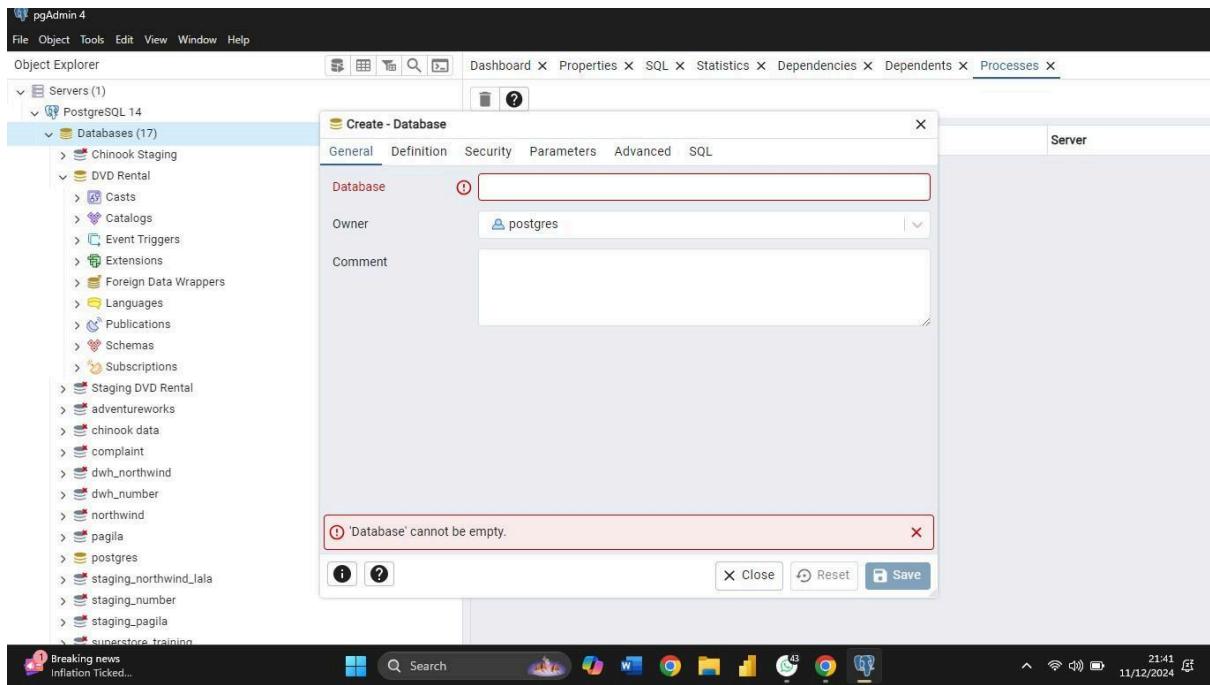
## 2.2 Tahapan Project

### 2.2.1 Input Database ke RDBMS (PostgreSQL).

Pertama-tama kami mengunduh data OLTP *database* DVD Rental yang berformat ‘tar’. Setelah di *download*, buka aplikasi pgAdmin untuk membuat dan merestore database DVD Rental. Selanjutnya klik kanan pada menu *database* dan pilih *create* dan pilih *database*.

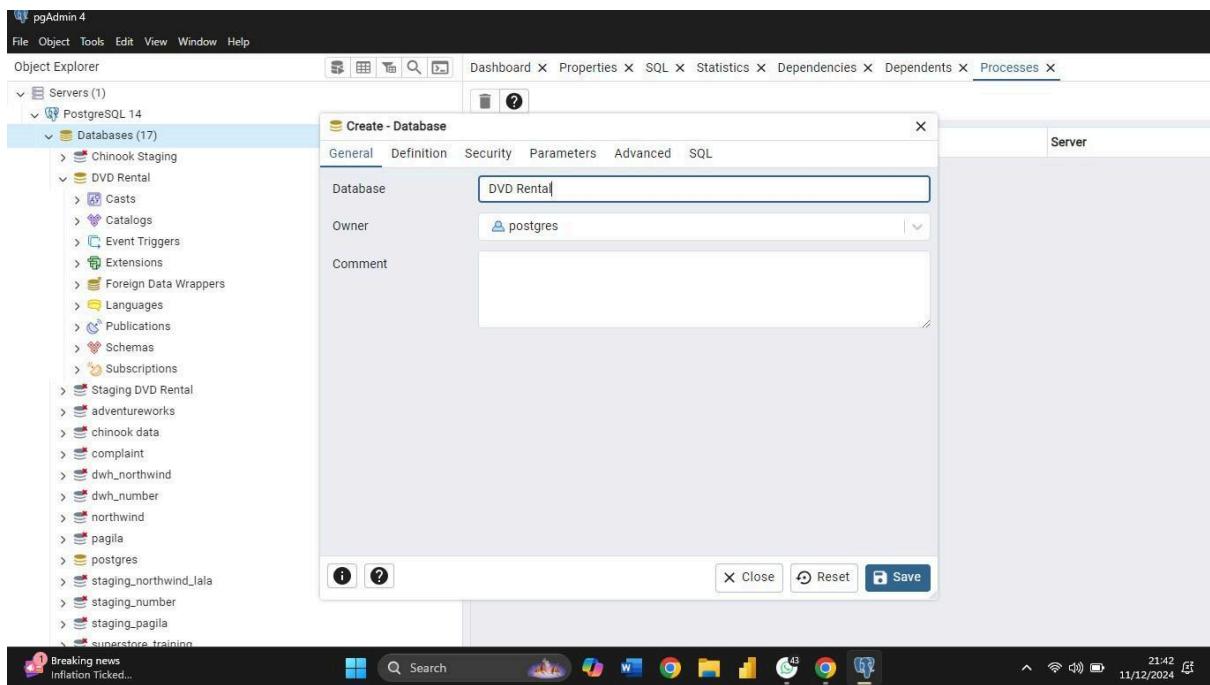


Gambar 2.2 Create database DVD rental



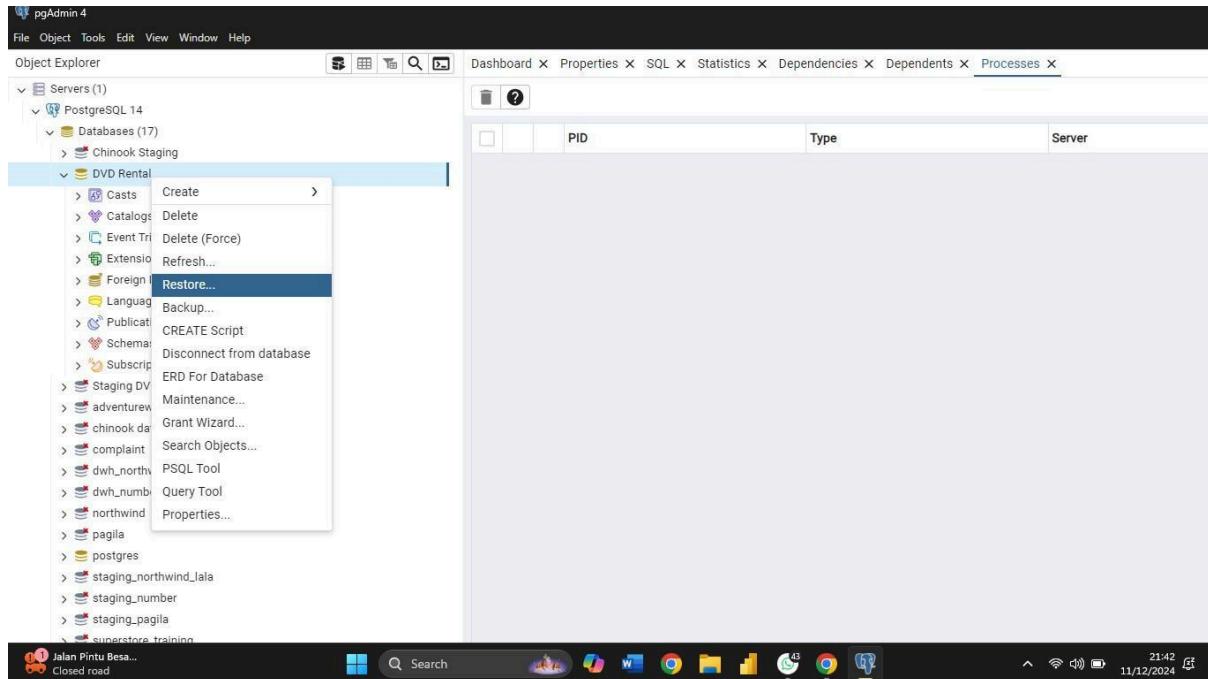
Gambar 2.3 Kolom create-database

Jika sudah muncul tampilan seperti di atas, berikan nama pada database sesuai dengan yang kita inginkan.



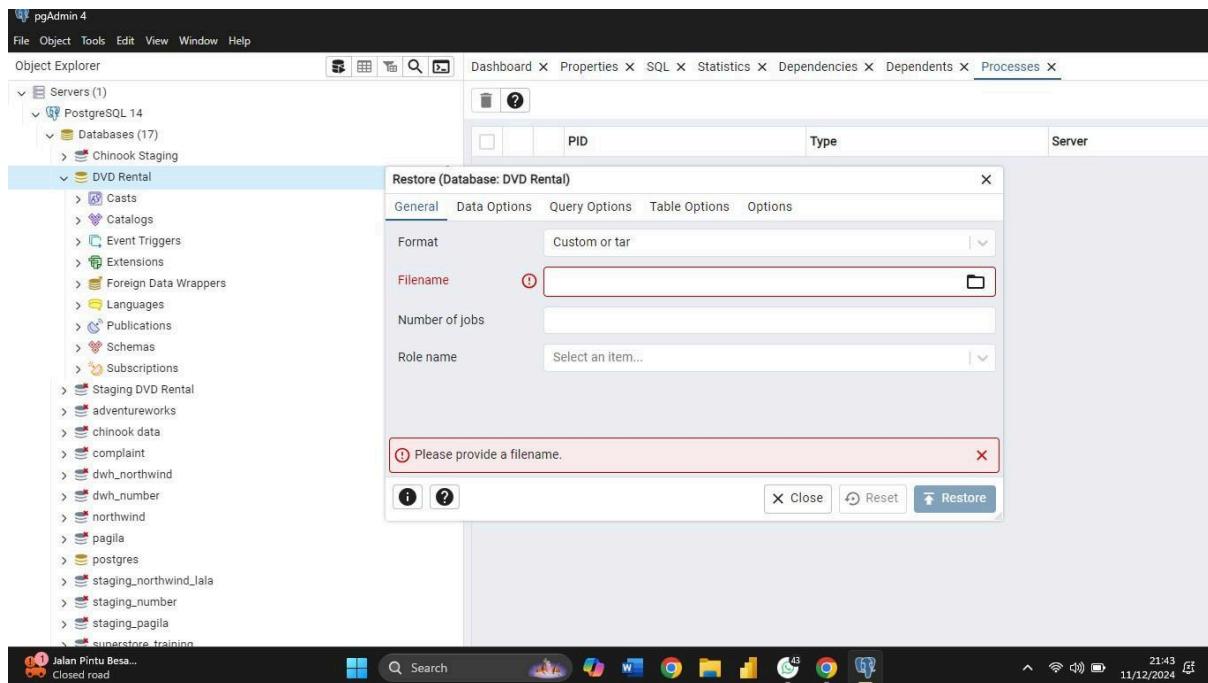
Gambar 2.4 Memberikan nama pada database DVD rental

Karena kami menggunakan database DVD Rental, maka berikan nama DVD Rental pada kolom database, Setelah itu tekan save.



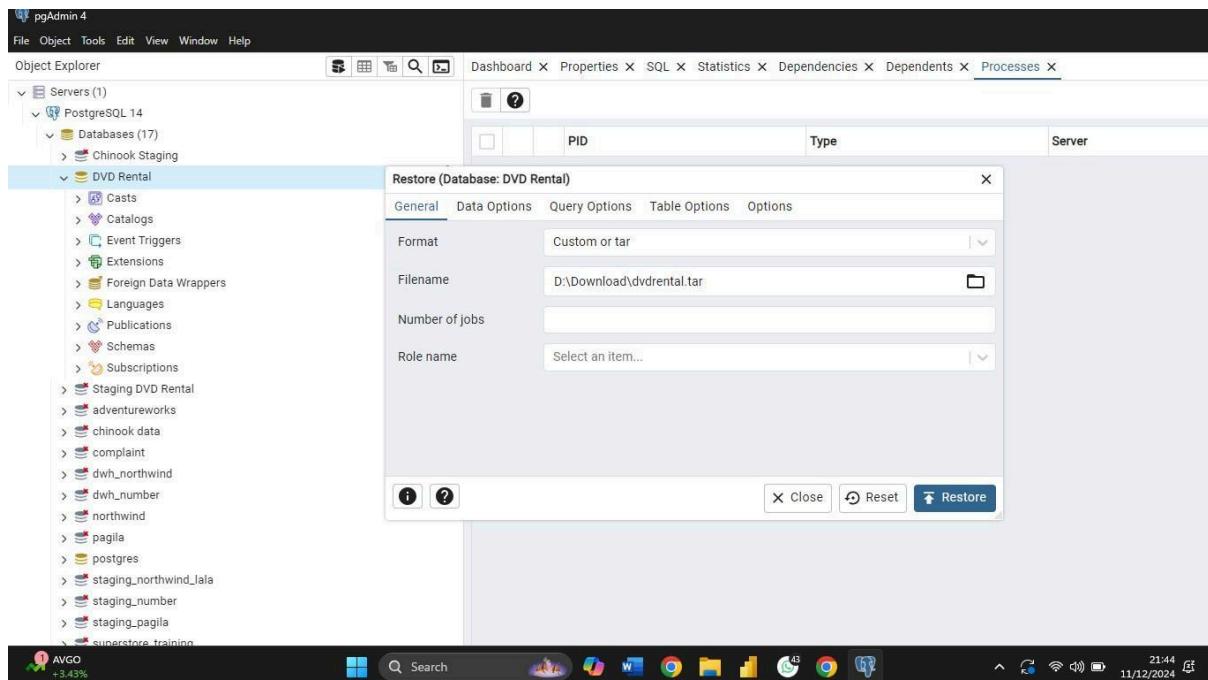
Gambar 2.5 Restore database DVD rental

Setelah *database* dibuat, klik kanan pada *database* lalu klik *restore*.



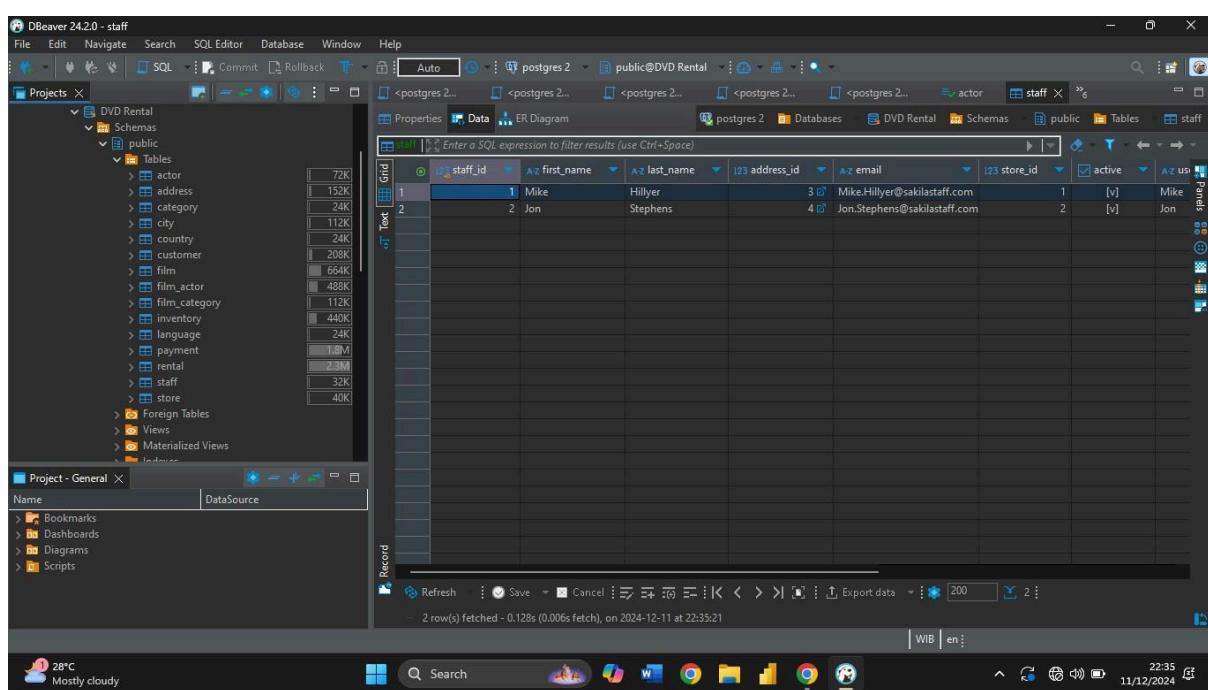
Gambar 2.6 Menambahkan file DVD rental untuk di restore

Jika sudah, akan terlihat kolom *restore* database DVD Rental seperti di atas.



Gambar 2.7 Restore database

Terakhir, tambahkan file tar *database* DVD Rental ke dalam kolom filename dan klik *restore*.



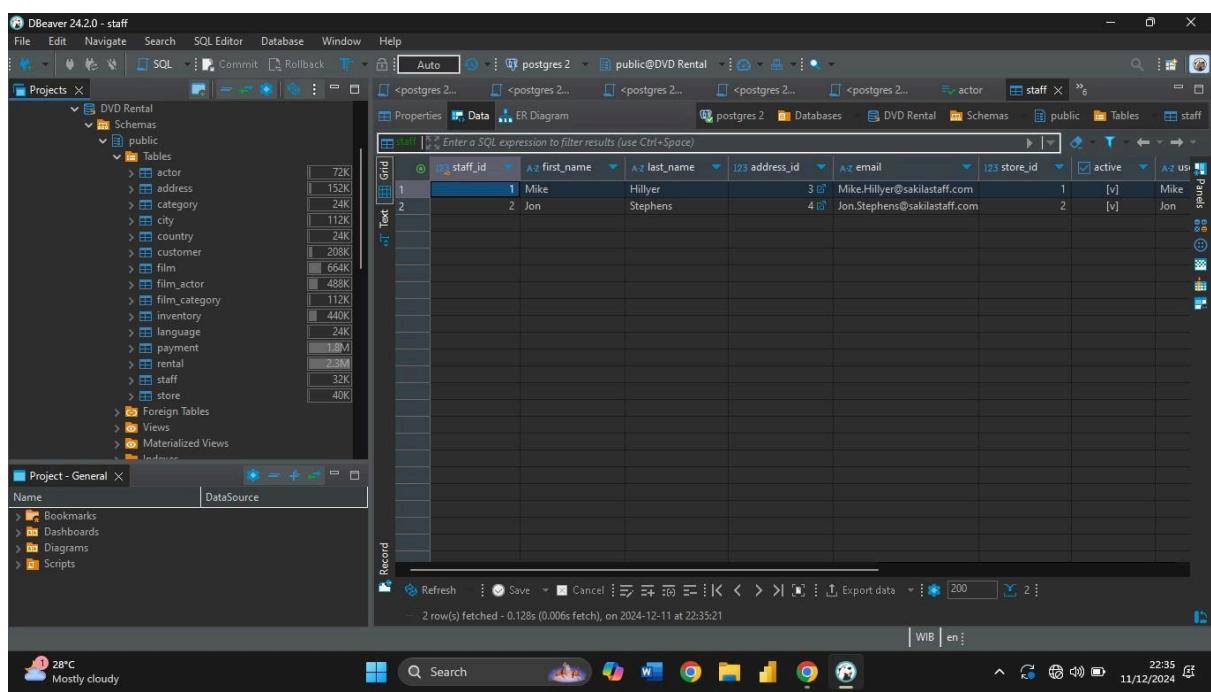
Gambar 2.8 Memastikan file yang di restore di Dbeaver

Untuk memastikan apakah file yang di *restore* sudah masuk atau belum, bisa dilihat di Dbeaver dan buka tabelnya satu-persatu.

## 2.2.2 Pembuatan Staging Area

*Staging* adalah proses mempersiapkan dan mengatur data dan di lingkungan sementara sebelum data tersebut digunakan dalam proses utama, seperti analisis, pelaporan, atau pemuatan ke dalam sistem produksi. Proses ini biasanya dilakukan dalam tahap ETL (Extract, Transform, Load) untuk memastikan data siap digunakan. Staging memiliki beberapa tujuan penting, seperti mempermudah pengumpulan data dari berbagai sumber, melakukan validasi dan pembersihan data, serta memastikan keamanan data sebelum diintegrasikan ke sistem utama.

Dalam mengolah database DVD Rental, kami melakukan proses staging terlebih dahulu menggunakan Pentaho. Kami membuat beberapa transformasi dan *job* untuk akhirnya bisa membuat *schema staging* di *database staging* DVD Rental. Namun sebelum beralih ke Pentaho, pertama-tama kita harus membuat *database staging* DVD Rental terlebih dahulu di Dbeaver.



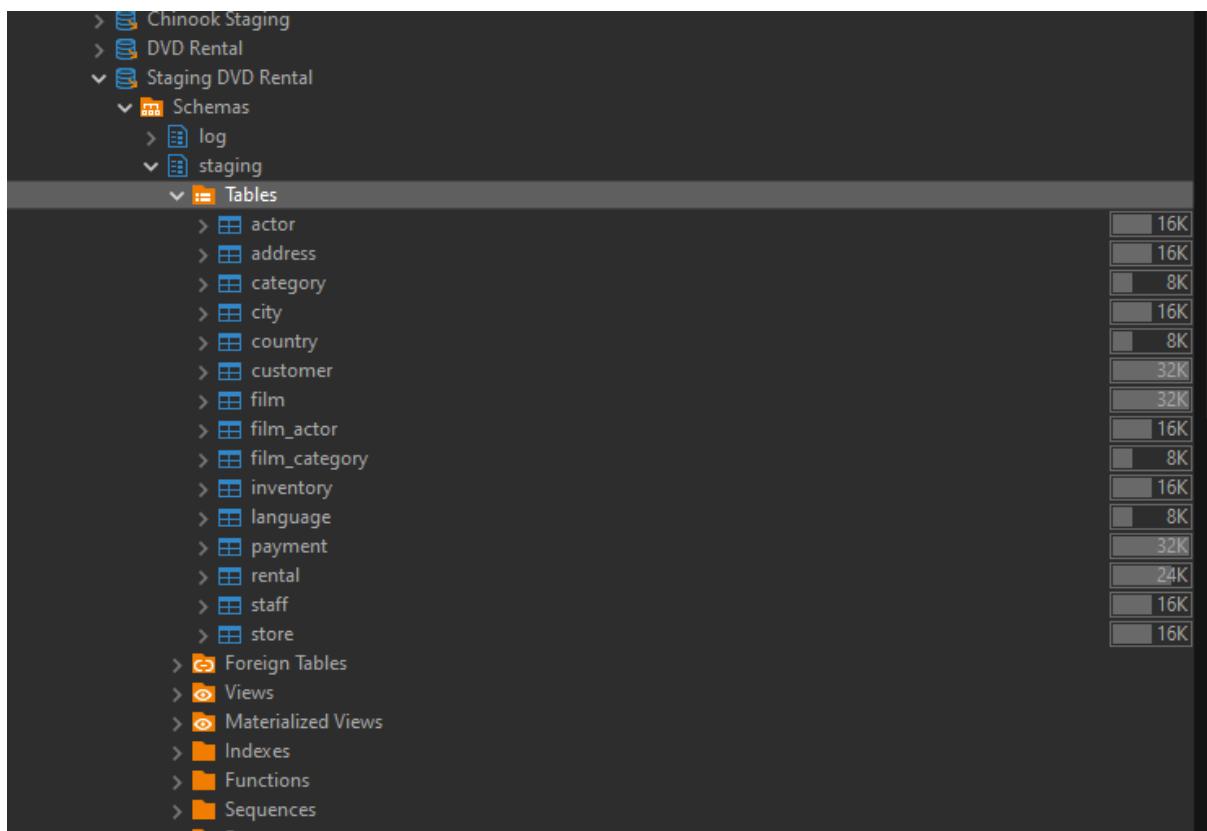
Gambar 2.9 Memilih Semua tabel di DVD rental

Lalu setelah membuat *database* untuk *staging*, *copy* DDL dari semua tabel yang ada di *database*. Klik semua tabel dengan menekan *shift*, lalu klik kanan dan *generate SQL*. Pada bagian *generate SQL* pilih DDL.

Gambar 2.10 Mengcopy DDL dari semua tabel

Gambar 2.11 Paste DDL yang sudah di copy ke script baru

Jika DDL sudah di *copy*, paste semua syntaxnya ke *database staging* dengan membuat *new SQL script*. Jika sudah di paste ke *script* baru, running semua syntax dengan menekan *ctrl + A* dan *enter*. Maka kerangka tabel database DVD Rental sudah tersedia di *database staging*.



Gambar 2.12 Memeriksa kerangka tabel yang sudah dibuat

Selanjutnya jika kerangka tabel sudah tersedia seperti gambar diatas, rename *schema public* menjadi *schema staging*. Setelah itu kita harus membuat *schema log* untuk memeriksa data yang akan masuk nantinya.

```

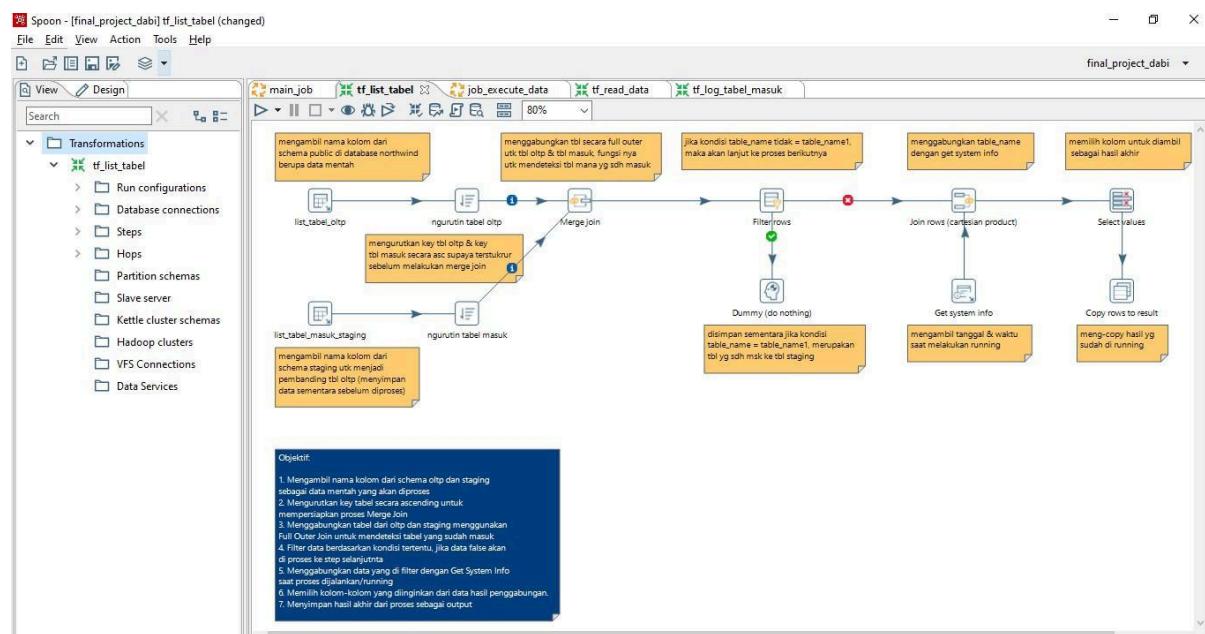
--log.log_table_masuk definition
--drop table
--drop table log.log_table_masuk;
create table log.log_table_masuk (
    table_name varchar(100) null,
    input_date timestamp null
);

--log.staging_log definition
--drop table
--drop table log.staging_log;
create table log.staging_log (
    table_name varchar(100) null,
    source_records varchar(100) null,
    source_date timestamp null,
    target_records varchar(100) null,
    target_date timestamp null,
    etl_date_start timestamp null,
    logical varchar(100) null
);

```

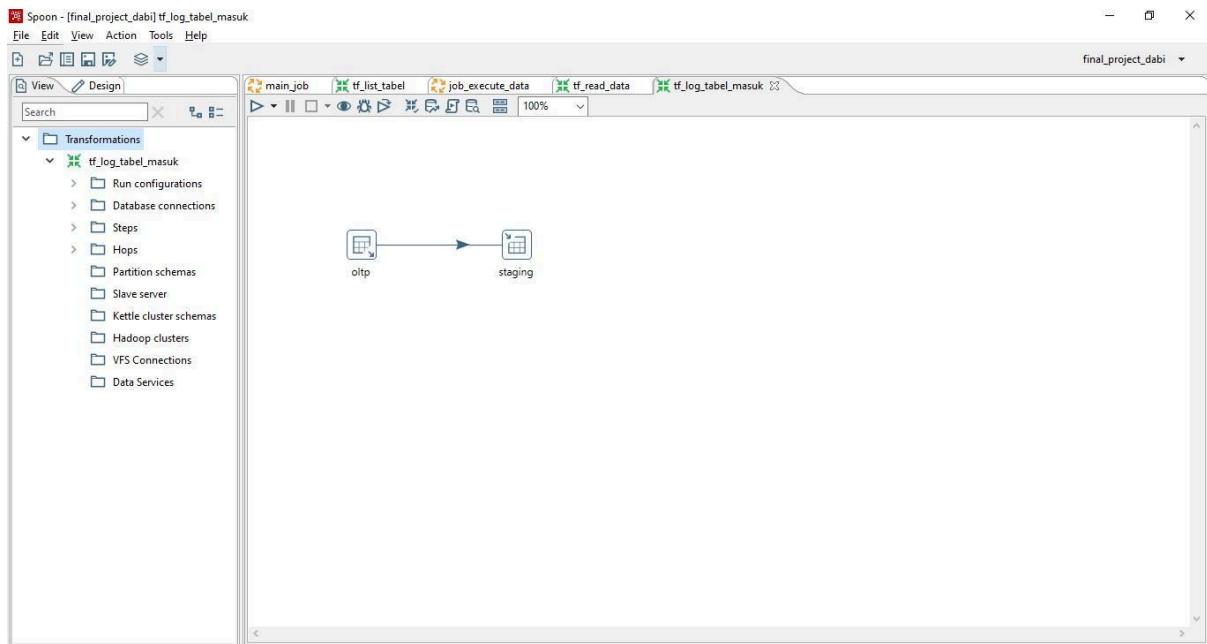
Gambar 2.13 Proses pembuatan tabel log

Buat *syntax* seperti gambar diatas untuk membuat tabel log\_table\_masuk dan tabel staging\_log. Membuat tabel log saat sedang *staging* data juga sangat penting, terutama dalam proses ETL. Fungsinya adalah untuk mencatat setiap langkah dalam proses ETL, seperti kapan data di-*extract*, di-*transform*, atau di-*load*. Setelah proses di Dbeaver selesai dan sudah dipastikan tidak ada *error*, baru bisa beralih ke Pentaho untuk proses *staging* lebih lanjut.



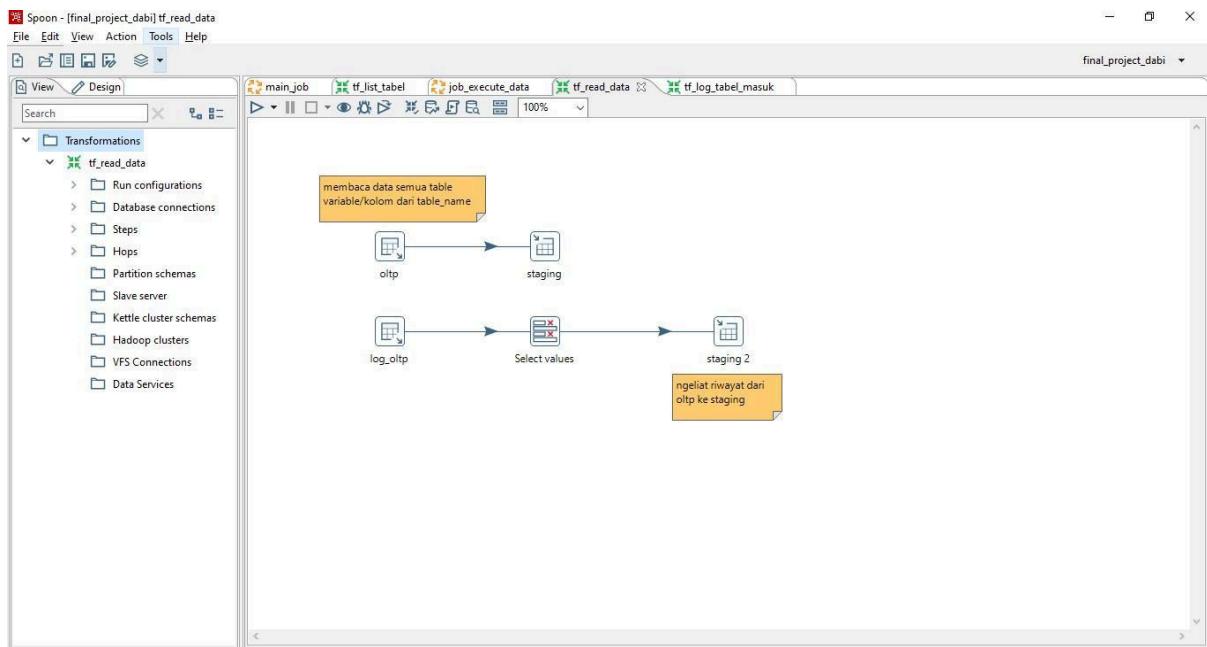
Gambar 2.14 Transformasi list tabel staging

Pada tahapan pertama, kami membuat transformasi list tabel untuk memastikan bahwa semua data dari tabel yang masuk sudah sesuai dan tidak ada data yang kurang. Jika diperhatikan, disana ada step list\_tabel\_masuk\_staging yang berguna sebagai pembanding tabel OLTP dan menyimpan data sementara sebelum diproses. Setelah itu, kami mengurutkan key tabel secara *ascending* agar terstruktur sebelum dilakukan merge join. Di tahap merge join kami menggabungkan tabel dari OLTP dan dari staging menggunakan full outer join agar terdeteksi tabel yang sudah masuk. Selanjutnya, filter data berdasarkan kondisi tertentu, jika ada false akan diproses ke step berikutnya. Lebih lanjut lagi, kami menggabungkan data yang di filter dengan Get System Info saat proses di *running*. Selanjutnya, kami menambahkan select value untuk memilih kolom-kolom mana yang ingin diambil dan terakhir copy hasil yang sudah di running menggunakan copy rows to result.



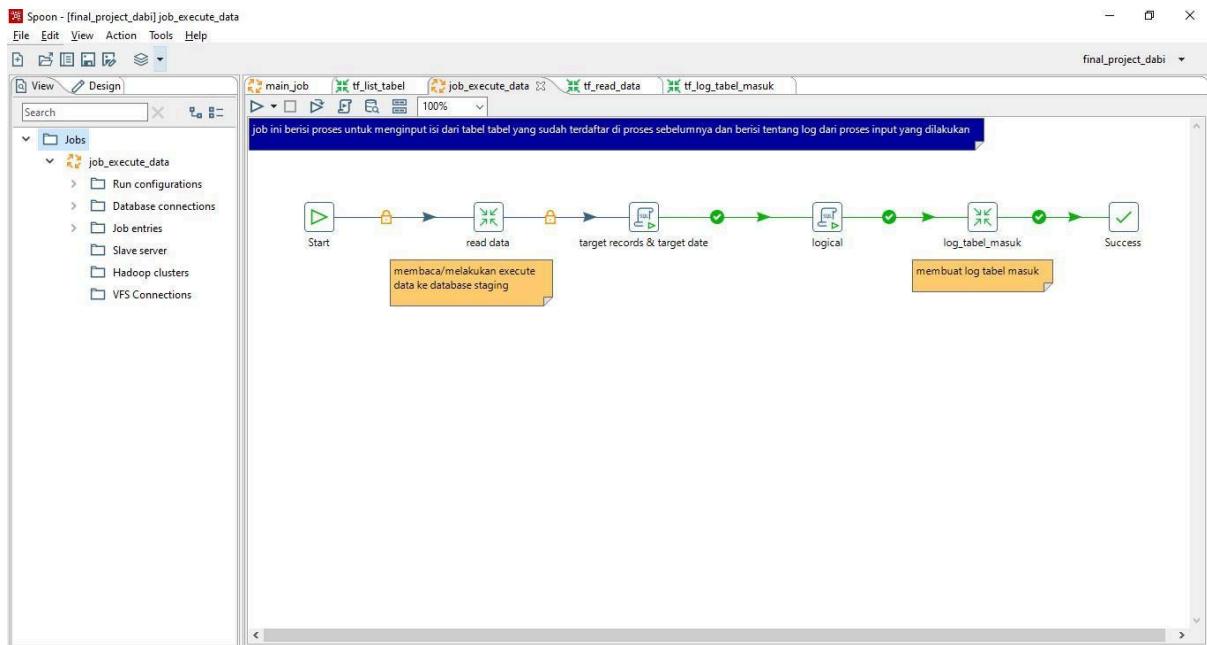
Gambar 2.15 Transformasi log tabel masuk

Setelah melakukan transformasi list tabel, selanjutnya kita membuat transformasi log tabel masuk. Pada tahapan ini hanya memindahkan data dari OLTP ke database staging schema log.



Gambar 2.16 Transformasi read data

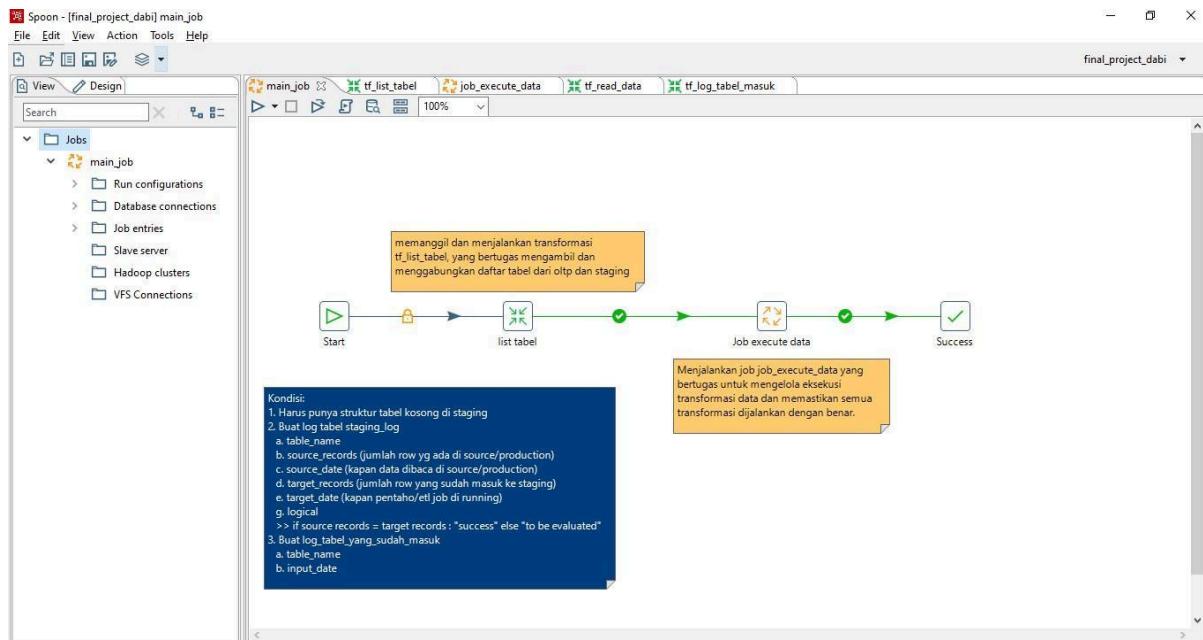
Pada transformasi read data, hubungkan kembali database OLTP dengan staging untuk membaca data semua tabel variabel/kolom dari table\_name. Disini kita juga ingin melihat riwayat perpindahan dari OLTP ke staging.



Gambar 2.17 Job execute data

Setelah semua transformasi siap, kita bisa langsung mengintegrasikannya menggunakan job. Seperti pada gambar diatas, kita membuat job\_execute\_data untuk menginput isi dari tabel-tabel yang sudah terdaftar di proses sebelumnya dan berisi tentang log dari proses input

yang dilakukan. fungsi dari job execute data adalah memastikan semua transformasi dijalankan dengan benar.



Gambar 2.18 Main job staging

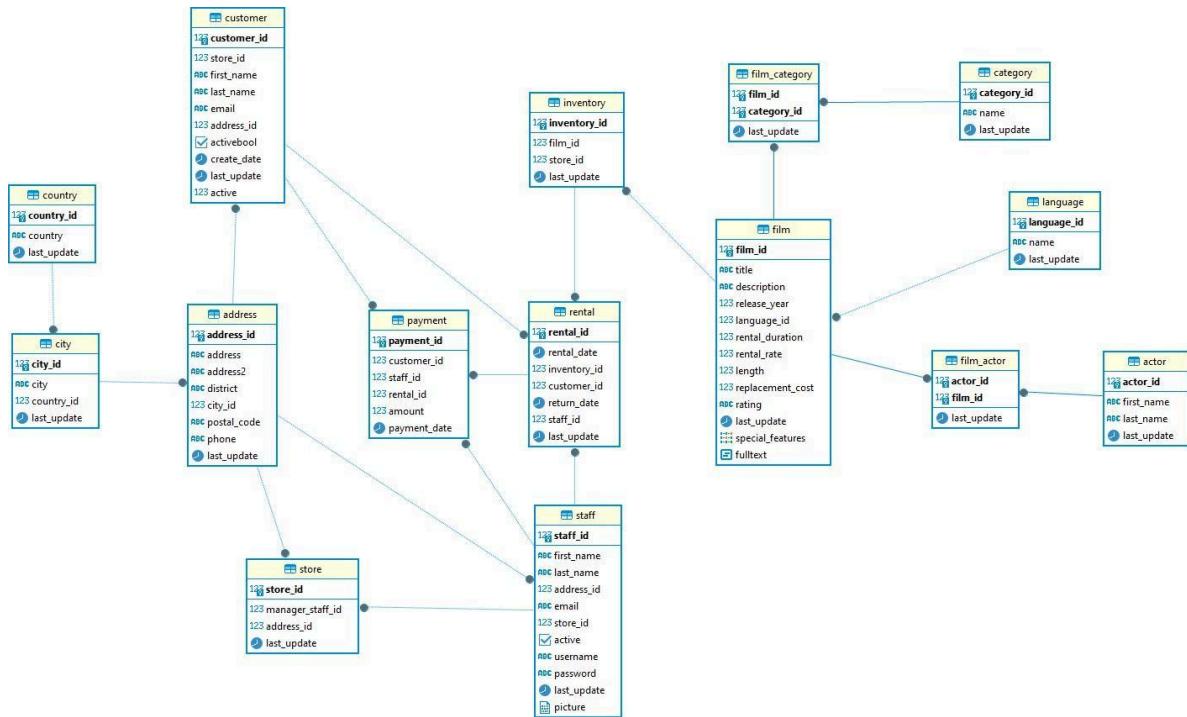
Klimaks dari proses staging berada pada main job yang mengintegrasikan tahapan list tabel dan job execute data. Di main job ini kita memanggil transformasi list tabel yang bertugas mengambil dan menggabungkan daftar tabel dari OLTP dan staging. Lalu kita juga memanggil job execute data yang bertugas untuk mengelola eksekusi transformasi data. Sebelum di *running*, pastikan dulu semua parameter sudah sesuai untuk menghindari *error*. Jika dirasa semuanya sudah sesuai baru dapat kita *running*. Setelah di *running* semua data sudah masuk ke database staging dan proses staging pun selesai.

### 2.2.3 Desain Data Warehouse

Data warehouse merupakan sistem penyimpanan data terpusat yang digunakan untuk mengumpulkan, menyimpan, dan mengelola data dari berbagai sumber untuk keperluan analisis dan pelaporan. Data warehouse dirancang untuk mendukung proses pengambilan keputusan dengan menyediakan data yang terstruktur, terorganisir, dan mudah diakses.

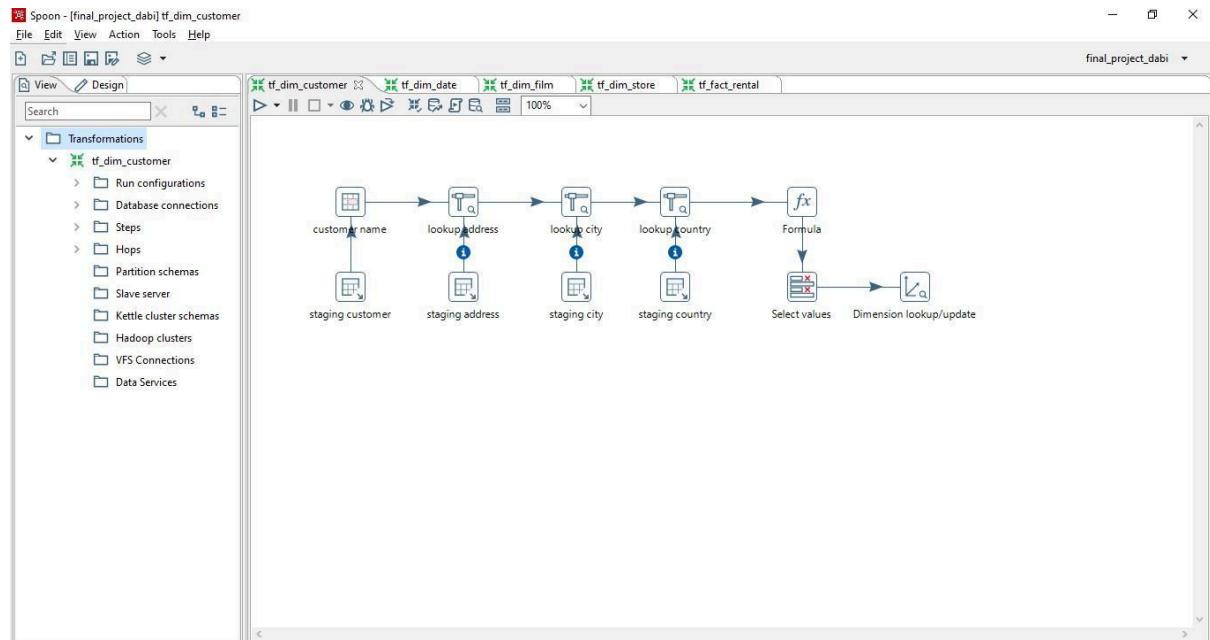
Dalam proses pembuatan Data Warehouse kami menggunakan kolaborasi antara dua tools yaitu Pentaho dan DBeaver. Sebelum menentukan tabel fakta dan dimensi dianjurkan untuk membentuk Entity Relation Diagram (ERD) untuk mempermudah penentuan tabel fakta dan dimensi. Kemudian, kami membentuk satu tabel fakta dan 4 tabel dimensi menggunakan

transformasi di software Pentaho. Berikut merupakan langkah-langkah dalam pembuatan Data Warehouse.



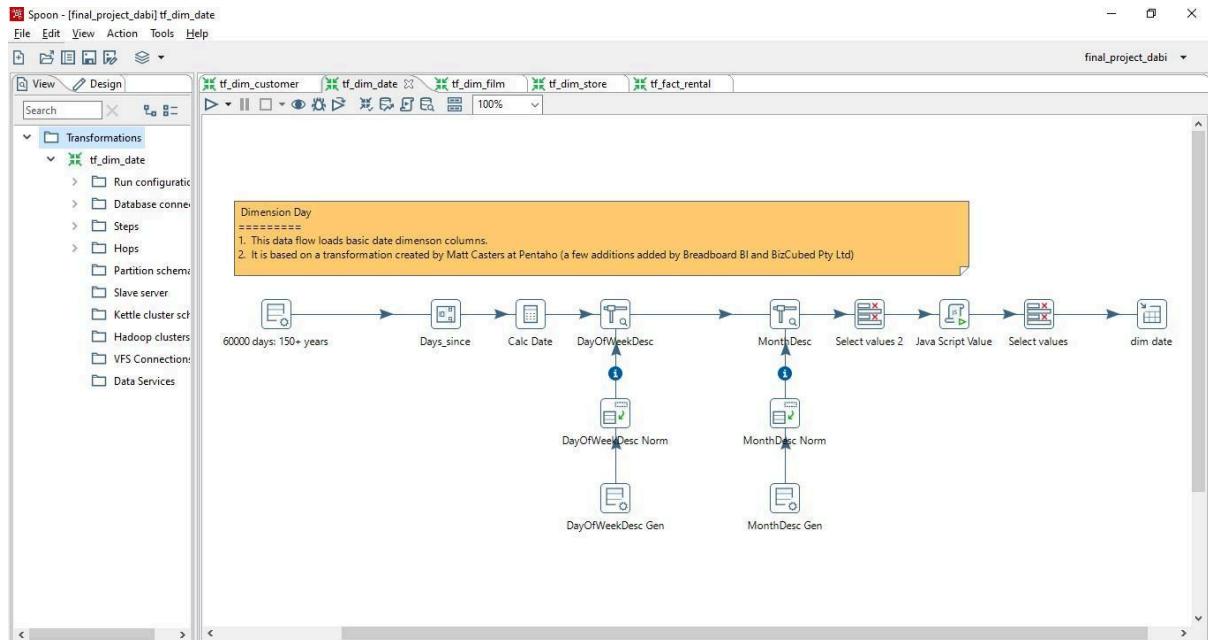
Gambar 2.19 Entity Relation Diagram (ERD)

ERD berguna untuk mempermudah penentuan tabel yang akan dilakukan proses Data Warehousing dengan membagi beberapa kelompok sehingga membentuk tabel yang terorganisir dan terintegrasi.



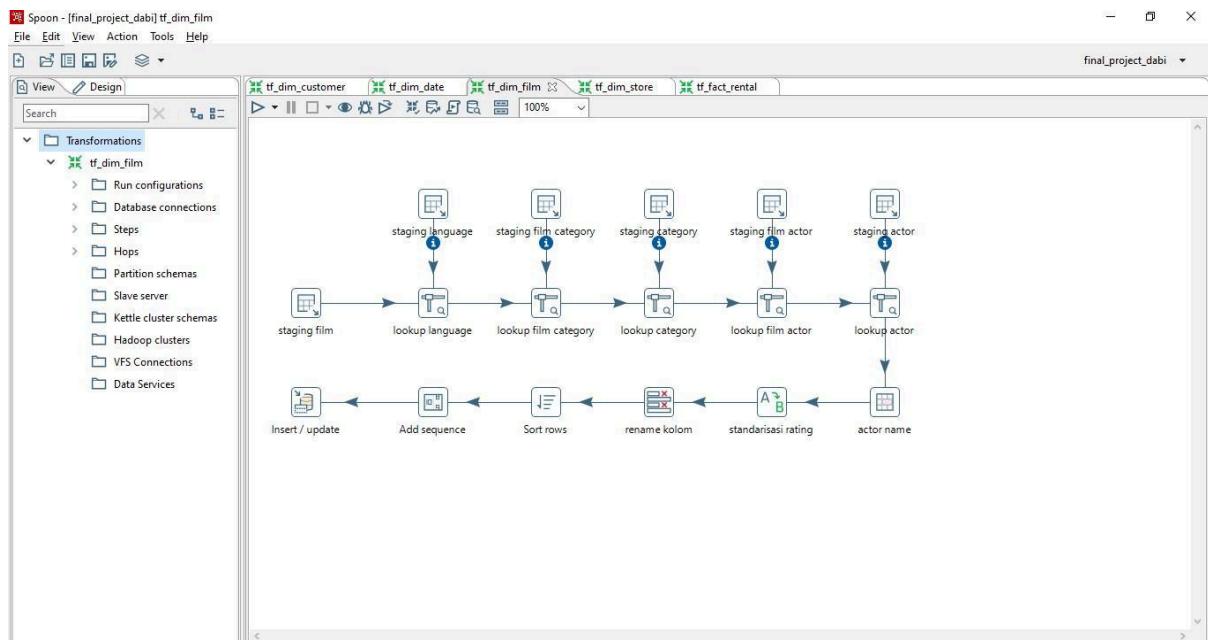
Gambar 2.20 Transformasi Dimensi Customer

Transformasi Dimensi Customer dimulai dengan memasukkan data staging customer, kemudian dilakukan langkah lookup untuk menggabungkan dengan data lainnya (staging data address, city, dan country).



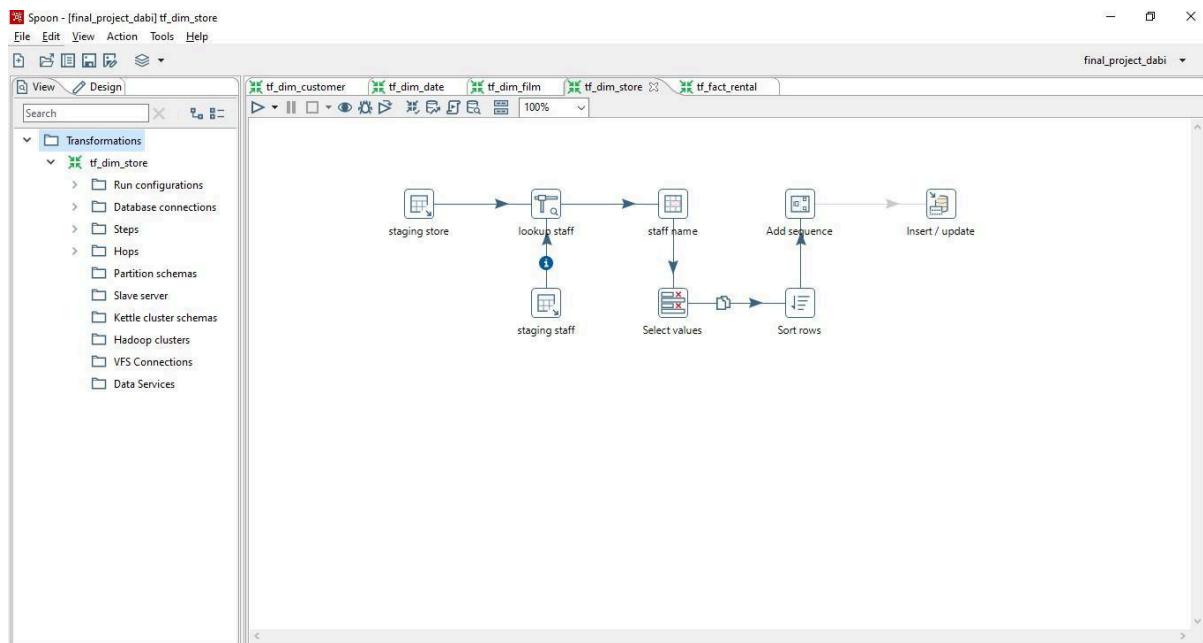
Gambar 2.21 Transformasi Dimensi Date (waktu)

Transformasi Dimensi Date (waktu) merupakan pembuatan tabel dengan atribut seperti tanggal, bulan, tahun, kuartal, dan lain sebagainya.



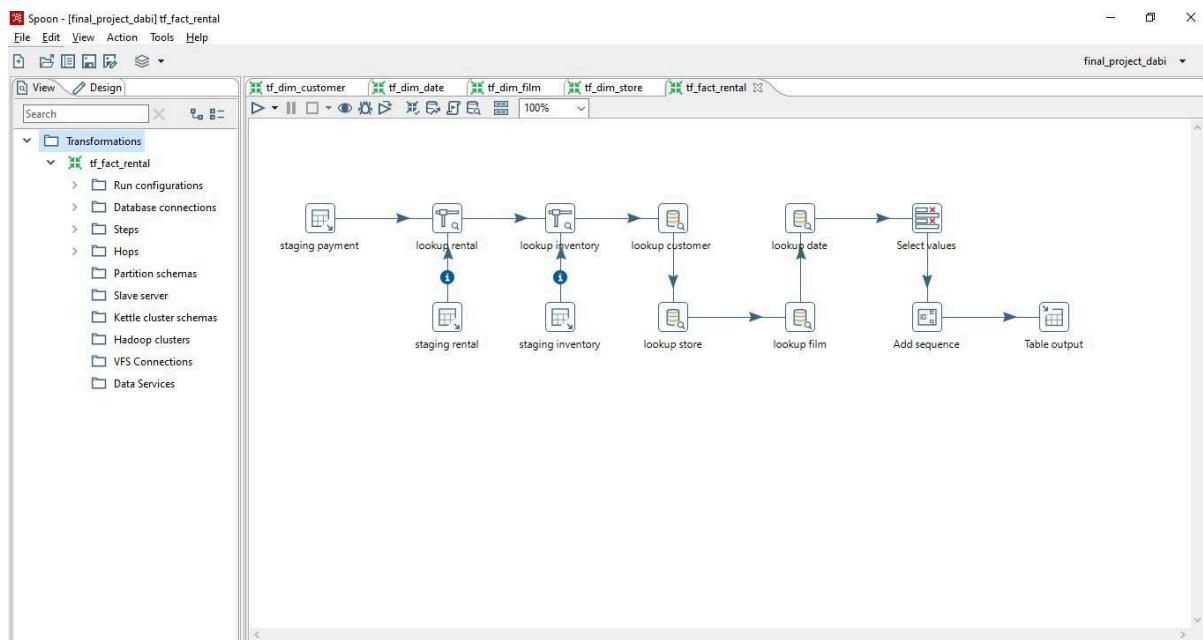
Gambar 2.22 Transformasi Dimensi Film

Transformasi Dimensi Film dimulai dengan memasukkan data staging film, kemudian dilakukan langkah lookup untuk menggabungkan dengan data lainnya (staging data language, film category, category, film actor, dan actor).



Gambar 2.23 Transformasi Dimensi Store

Transformasi Dimensi Store dimulai dengan memasukkan data staging store, kemudian dilakukan langkah lookup untuk menggabungkan dengan data staging staff.

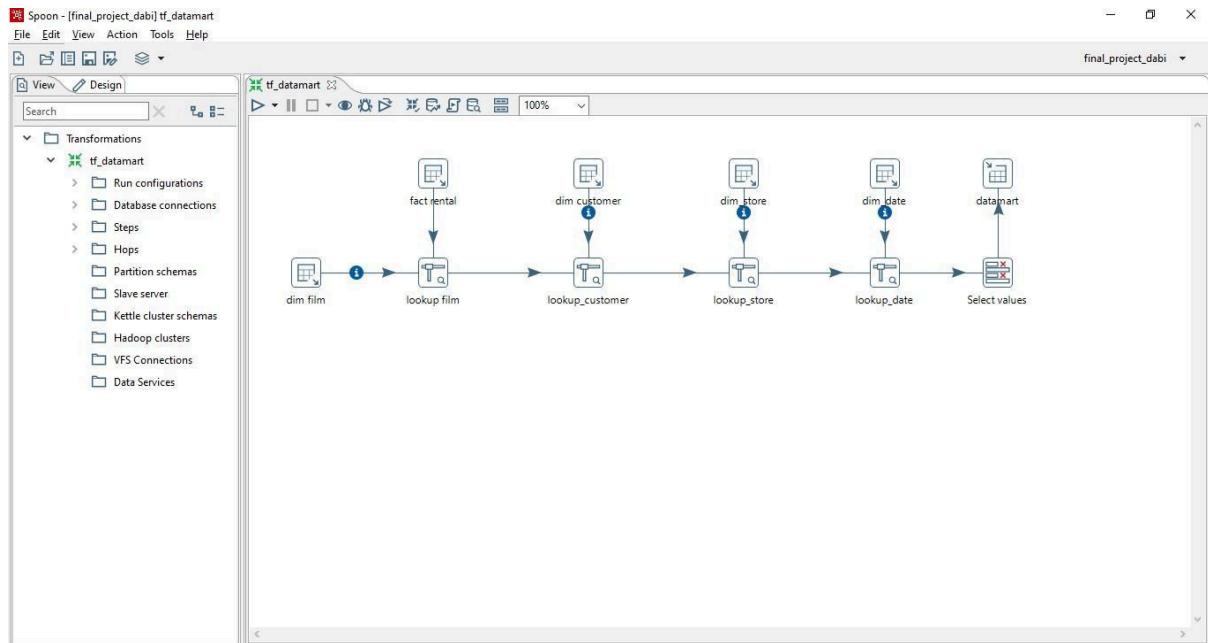


Gambar 2.24 Transformasi Fakta Rental

Transformasi Fakta Rental dimulai dengan memasukkan data staging payment, kemudian dilakukan langkah lookup untuk menggabungkan dengan data lainnya (staging data rental, inventory) kemudian dilakukan lookup dengan data dimensi sebelumnya (data customer, film, store, dan date). Selanjutnya akan menghasilkan tabel fakta yang akan digunakan sebagai Datamart.

#### 2.2.4 Pembuatan Data Mart

Data Mart merupakan sub kumpulan dari Data Warehouse yang dirancang khusus untuk melayani kebutuhan analisis atau pelaporan dari bisnis tertentu, seperti penjualan, pemasaran, keuangan, atau sumber daya manusia. Data mart menyediakan data yang lebih fokus dan relevan untuk pengguna tertentu, sehingga mempercepat proses pengambilan keputusan dan meningkatkan kinerja analisis.



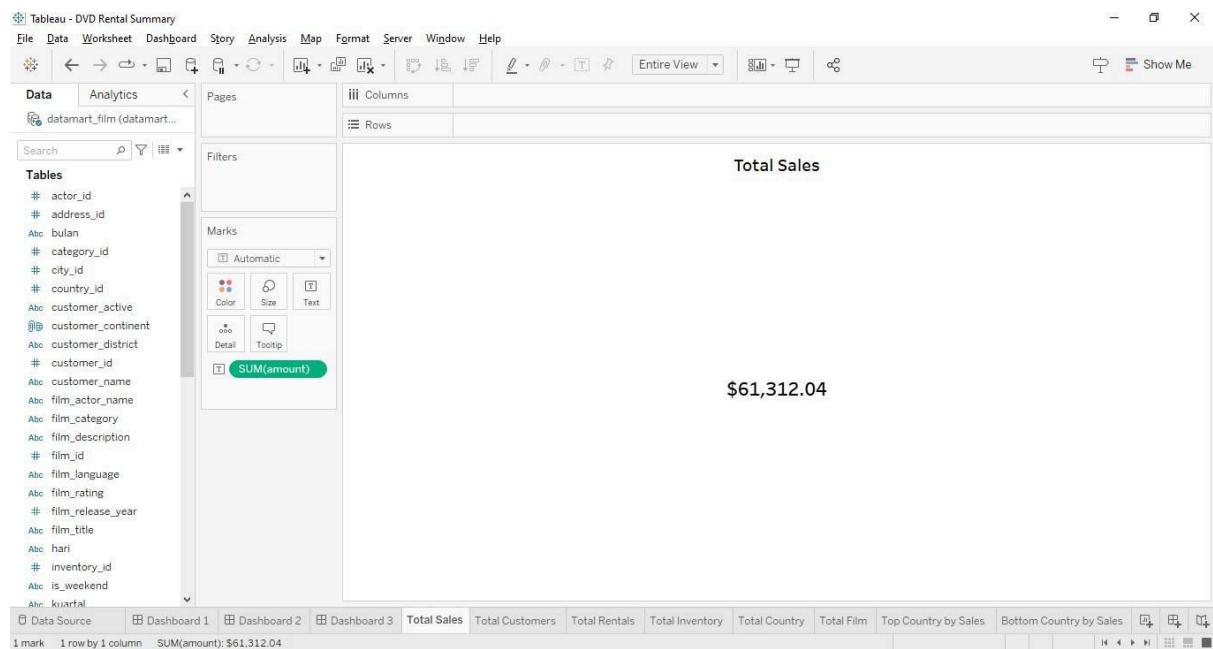
Gambar 2.25 Transformasi data mart

Pada pembuatan data mart kelompok kami, kami memusatkan pada dimensi film sebagai fokus utamanya. Kami memfokuskan kinerja film untuk meningkatkan pendapatan, memahami preferensi pelanggan, serta mengoptimalkan stok film di rental. Film sendiri adalah produk inti dari bisnis DVD rental. Dengan mengetahui performa setiap film seperti kategori, rating, durasi, dll langsung berdampak pada pendapatan. Seperti yang terlihat pada gambar diatas kami menggabungkan dimensi film dengan fact rental, dimensi customer, dimensi store, dimensi date dan akhirnya terbentuk sebuah data mart yang terstruktur.

## 2.2.5 Pembuatan Dashboard

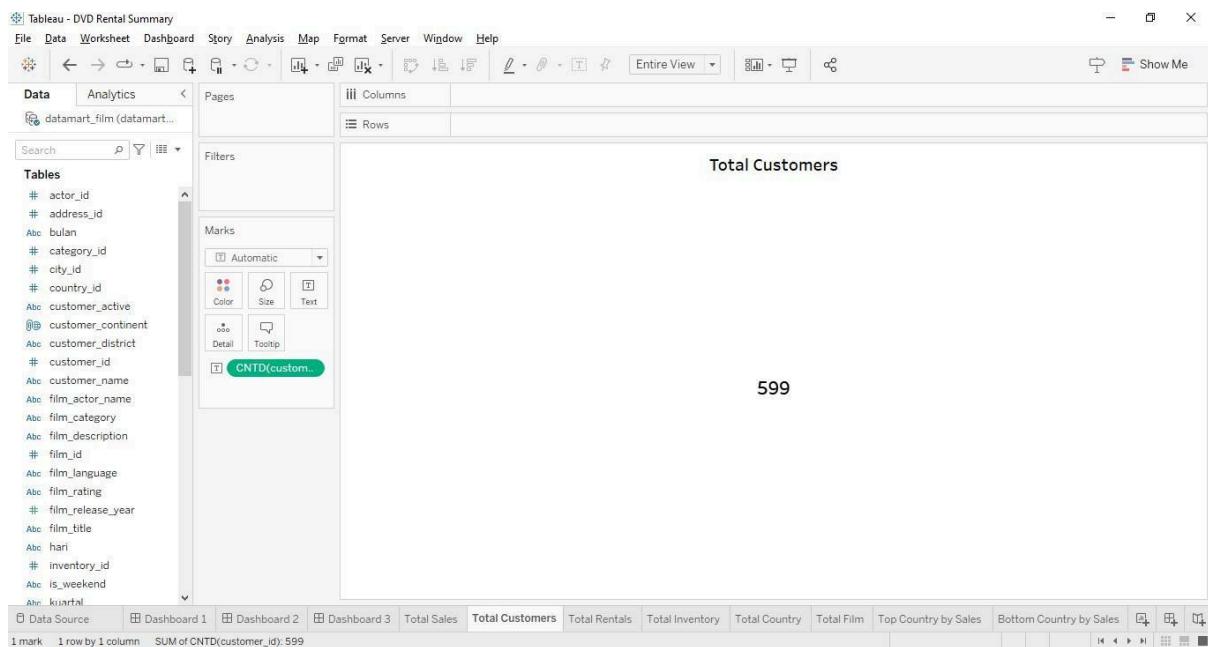
Dalam pembuatan dashboard, kelompok kami memutuskan untuk menggunakan Tableau sebagai *tools*-nya. Fokus utama dari pembuatan dashboard kami adalah memberikan gambaran komprehensif tentang operasional DVD Rental. beberapa metrik kunci yang kami ingin tampilkan antara lain total revenue, total rentals, total inventory, total customers, total film, dan total country. Kami ingin mengidentifikasi tren dan pola penting, memantau kinerja bisnis secara keseluruhan, dan tentunya memahami perilaku pelanggan dan kontribusi mereka. Informasi yang akan kami tunjukan di dashboard diharapkan dapat membuat keputusan strategis yang lebih baik dan meningkatkan pendapatan bisnis DVD Rental.

Langkah pertama yang kami lakukan adalah menghubungkan koneksi data mart ke dalam Tableau. Setelah dihubungkan, kami memulai membuat chart-chart yang nantinya kami akan ditampilkan di *dashboard*. Selain membuat chart kami juga membuat beberapa tabel yang berisi angka total dari kinerja bisnis.



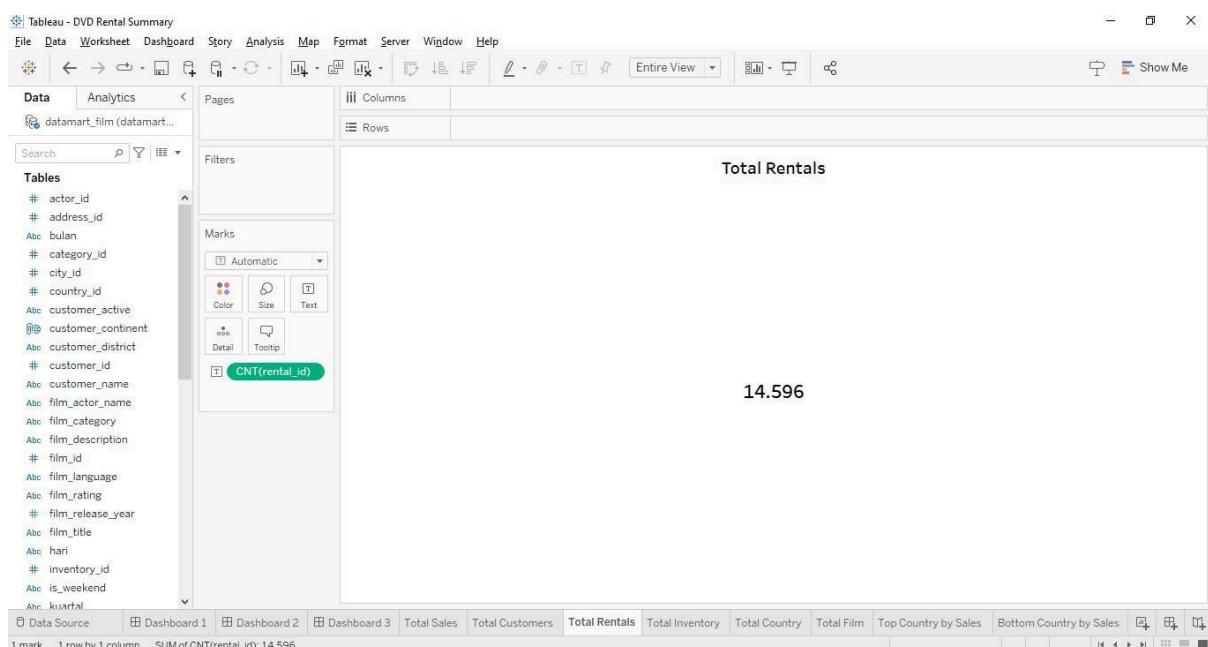
Gambar 2.26 Total Sales

tabel pertama yang kami buat adalah total sales yang berisi jumlah pendapatan yang didapat dari penyewaan DVD.



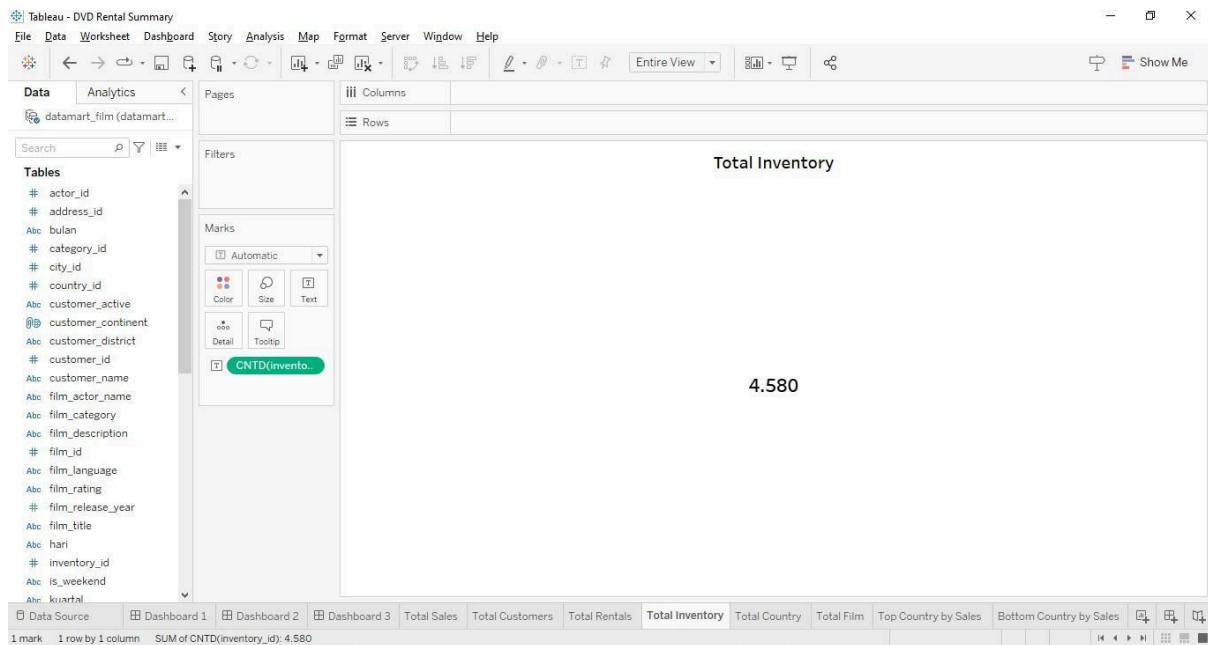
Gambar 2.27 Total Customer

Tabel yang kedua adalah total customer yang berisi jumlah total dari keseluruhan pelanggan yang pernah menyewa DVD.



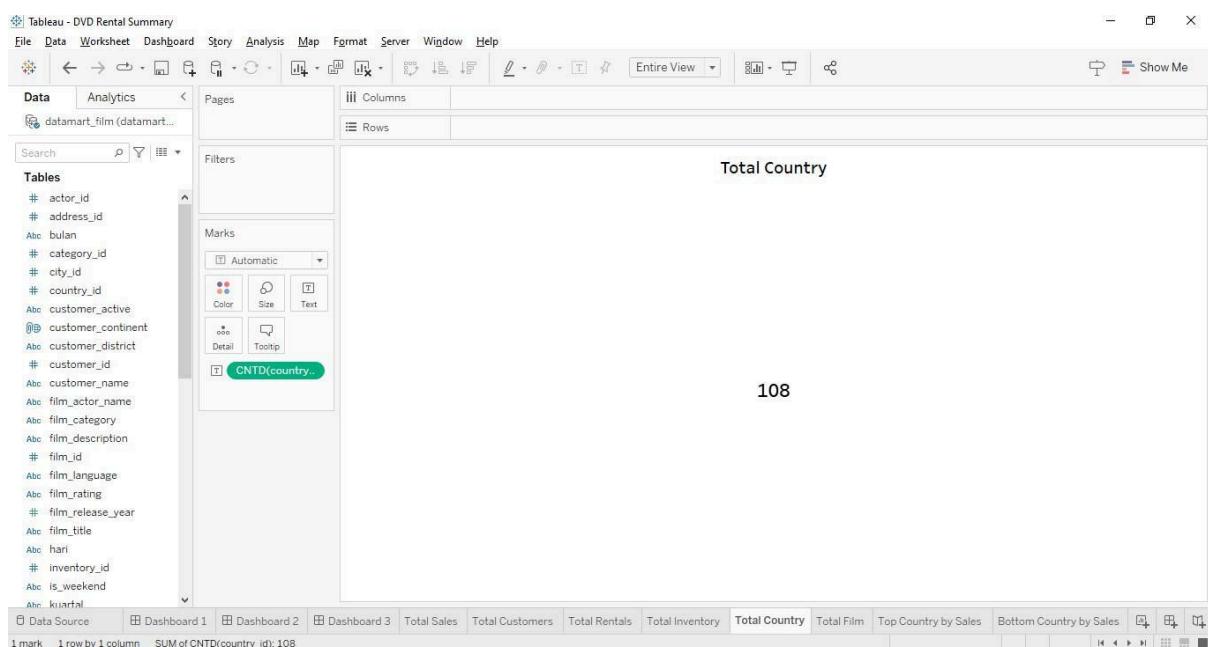
Gambar 2.28 Total Rentals

Tabel yang kedua kami menghitung total dari penyewaan yang sudah terlaksana selama bisnis berlangsung yaitu sebesar 14.596 kali sewa



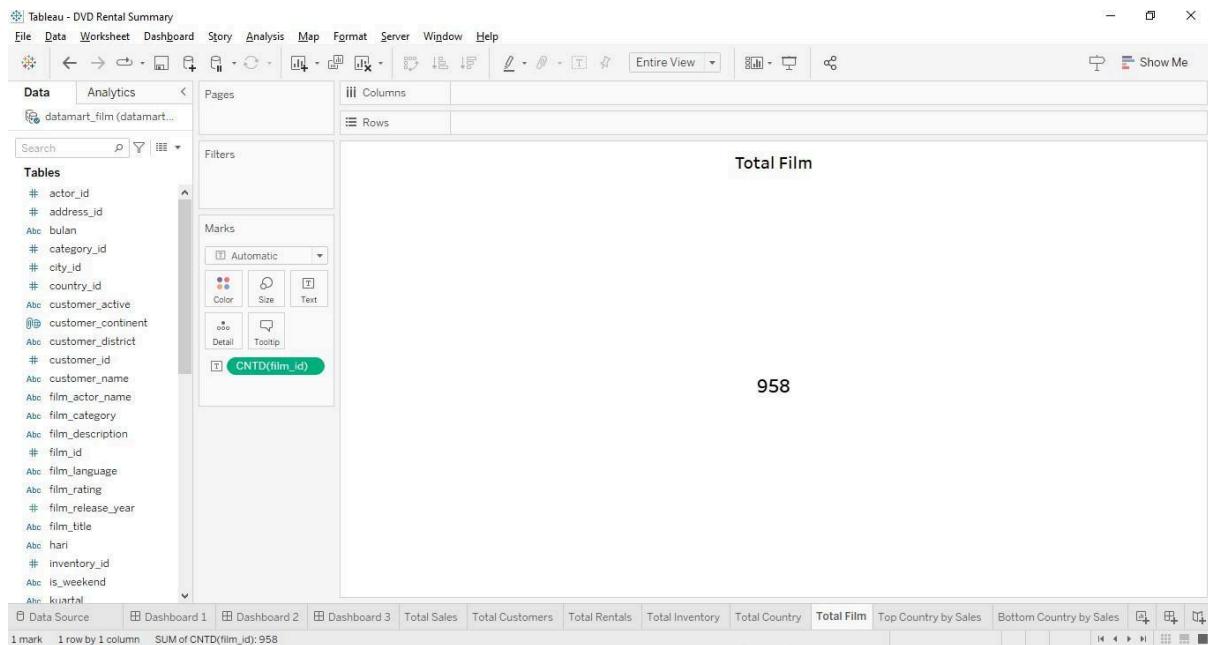
Gambar 2.29 Total Inventory

Tabel yang ketiga kami ingin menunjukkan jumlah dari persediaan yang ada yaitu sebesar 4.580.



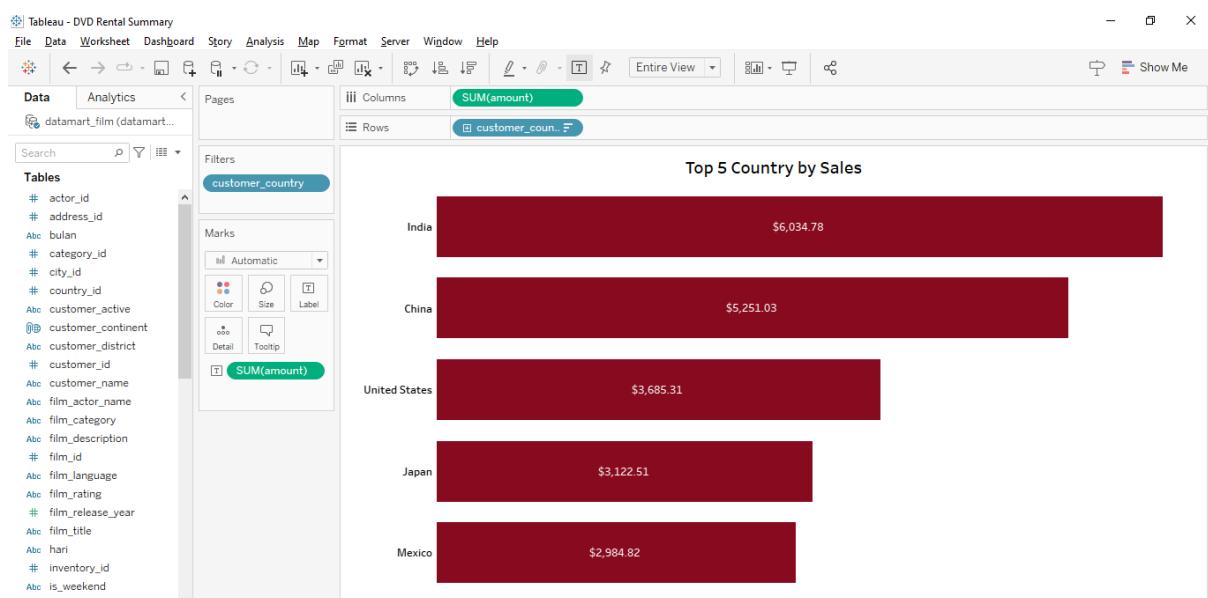
Gambar 2.30 Total Country

Tabel yang keempat, kami ingin mengetahui jumlah dari negara yang sudah pernah meminjam DVD. Sebanyak 108 negara sudah pernah meminjam di DVD Rental.



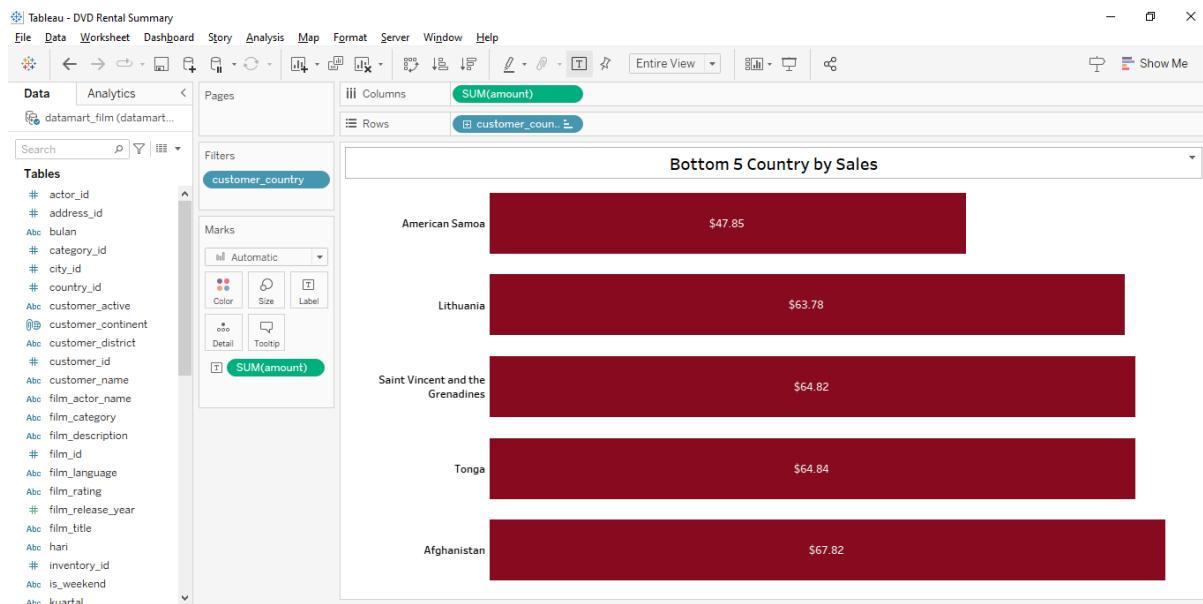
Gambar 2.31 Total Film

Tabel yang terakhir kami ingin melihat seberapa banyak film yang tersedia di penyewaan DVD. Sebanyak 108 film tersedia di DVD Rental.



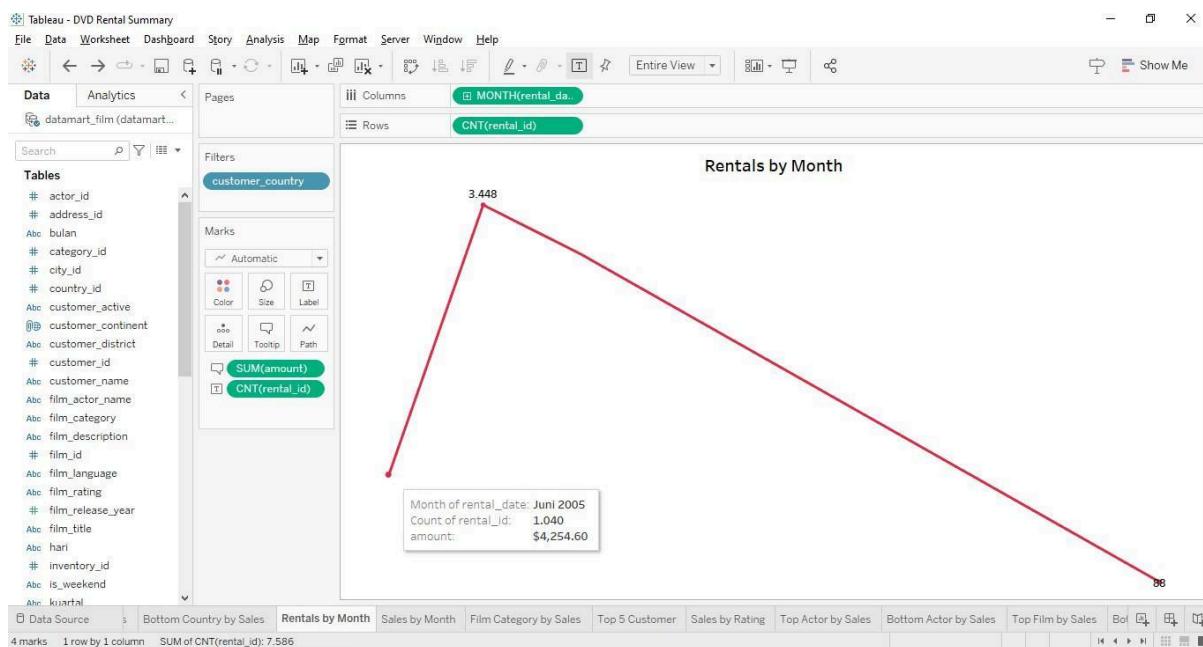
Gambar 2.32 Top 5 country by sales

Selanjutnya kami membuat *bar chart* yang berisi negara 5 teratas yang memiliki jumlah pendapatan sewa terbanyak. Kami memasukkan parameter jumlah ke kolom dan memasukan customer country ke row. Lalu kami menambahkan filter agar yang terlihat hanya top 10 negara. Urutkan *bar chart* dari yang tertinggi ke yang terendah dan beri sentuhan warna sesuai tema warna yang diinginkan.



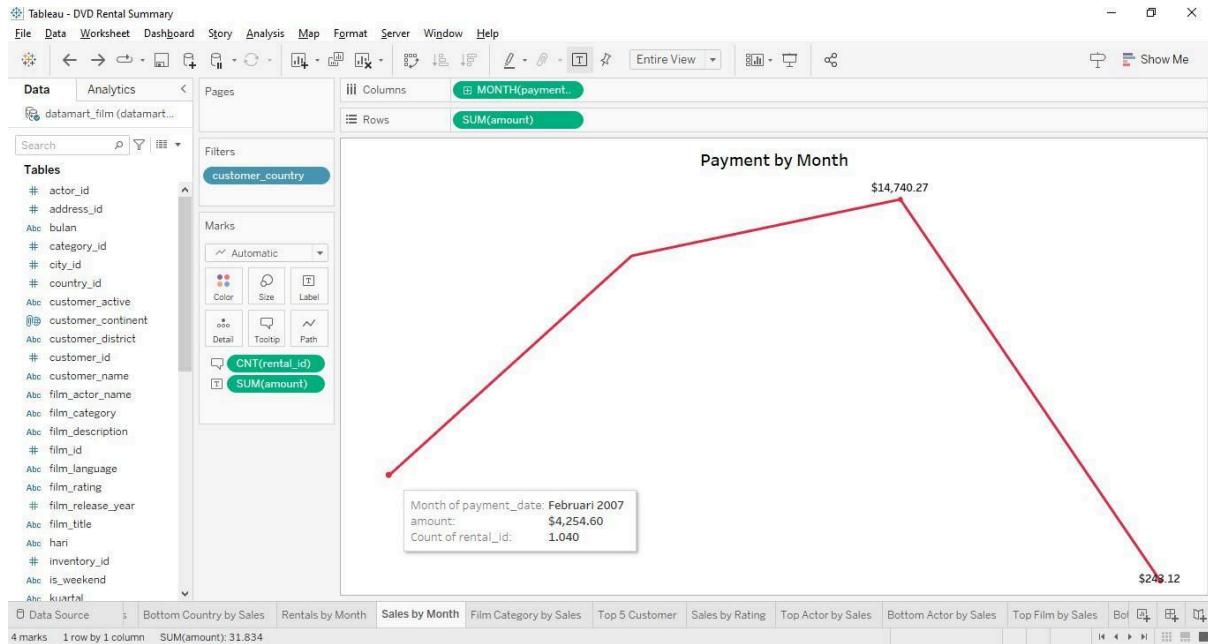
Gambar 2.33 Bottom 5 country by sales

Jika tadi kami membuat 5 negara teratas dari jumlah pendapatan, kali ini kami juga ingin menampilkan 5 negara terbawah dari jumlah pendapatan sewanya.



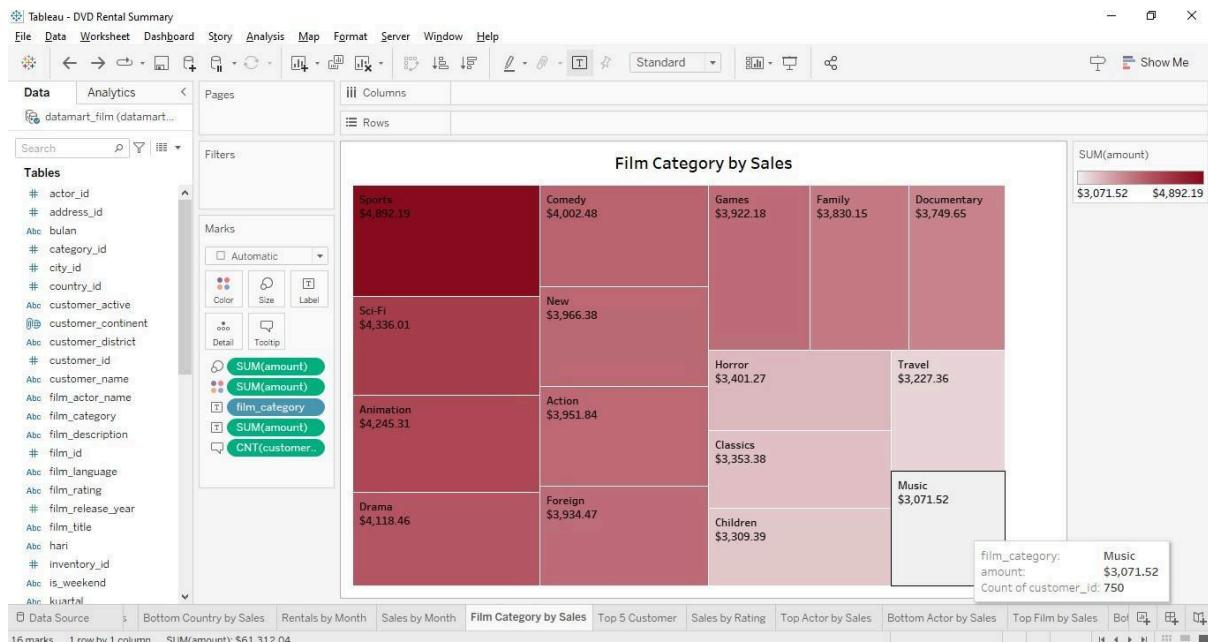
Gambar 2.34 Rentals by mouth

Selanjutnya kami membuat *line chart* yang berisi jumlah sewa berdasarkan bulan. Disini kami hanya menampilkan angka tertinggi dan terendahnya.



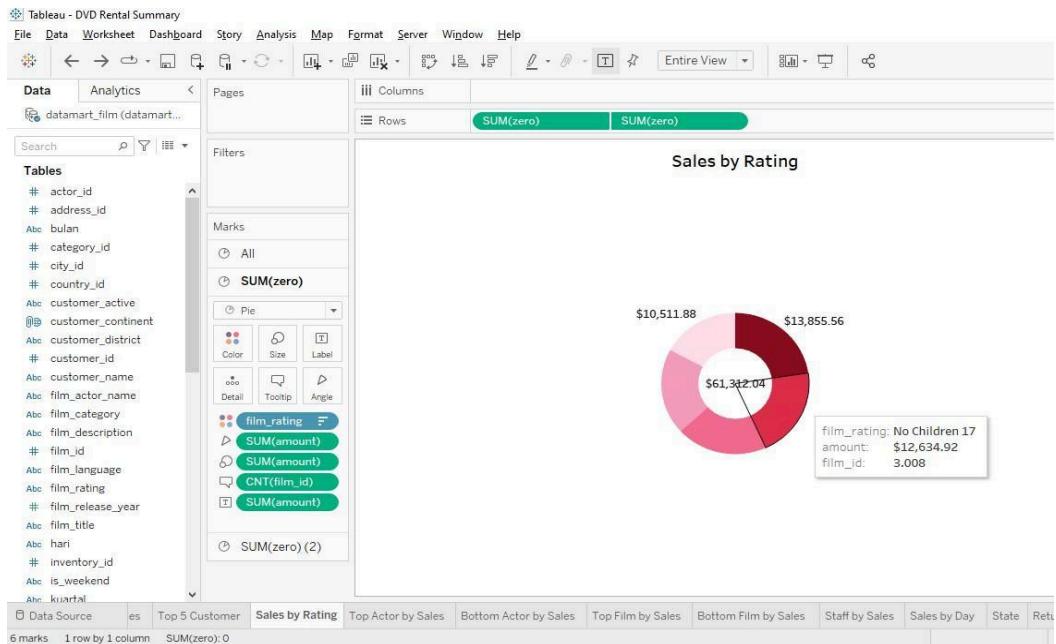
Gambar 2.35 Payment by month

Jika *line chart* yang sebelumnya adalah mengenai jumlah sewa berdasarkan bulan, untuk *line chart* diatas adalah pembayaran tertinggi dan terendah berdasarkan bulan.



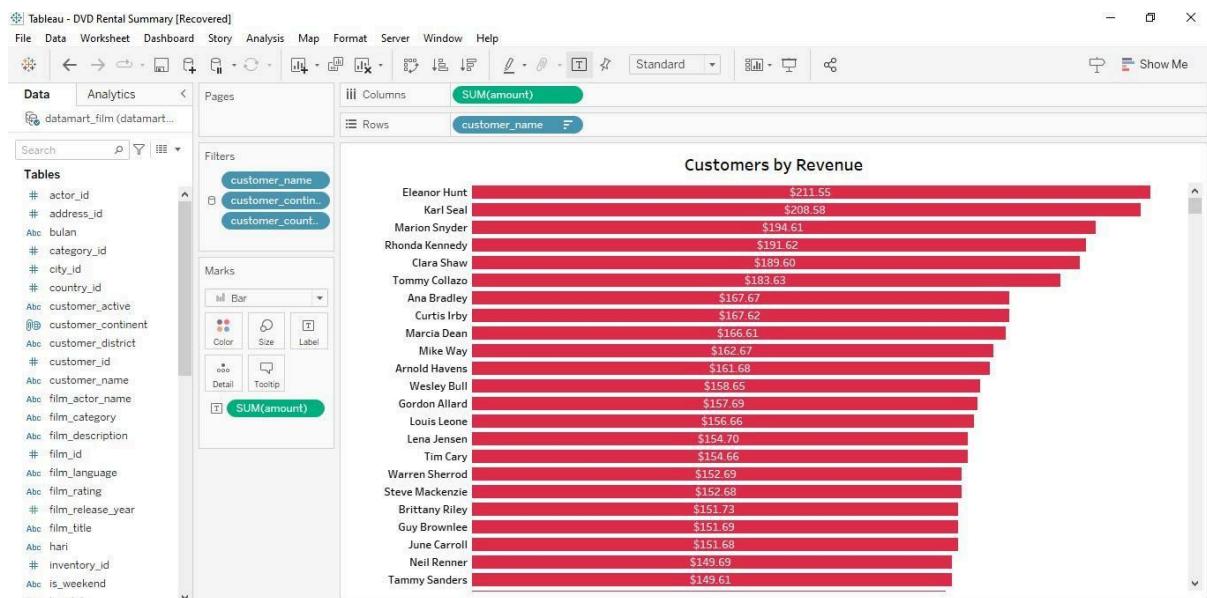
Gambar 2.36 Film category by sales

Kami juga membuat area chart untuk kategori film yang paling banyak terjual. Bisa terlihat pada gambar bahwa yang areanya paling luas dan yang warnanya paling gelap adalah kategori film yang paling banyak terjual. Untuk kategori film yang paling banyak terjual adalah kategori olahraga dan yang paling sedikit terjual adalah kategori musik.



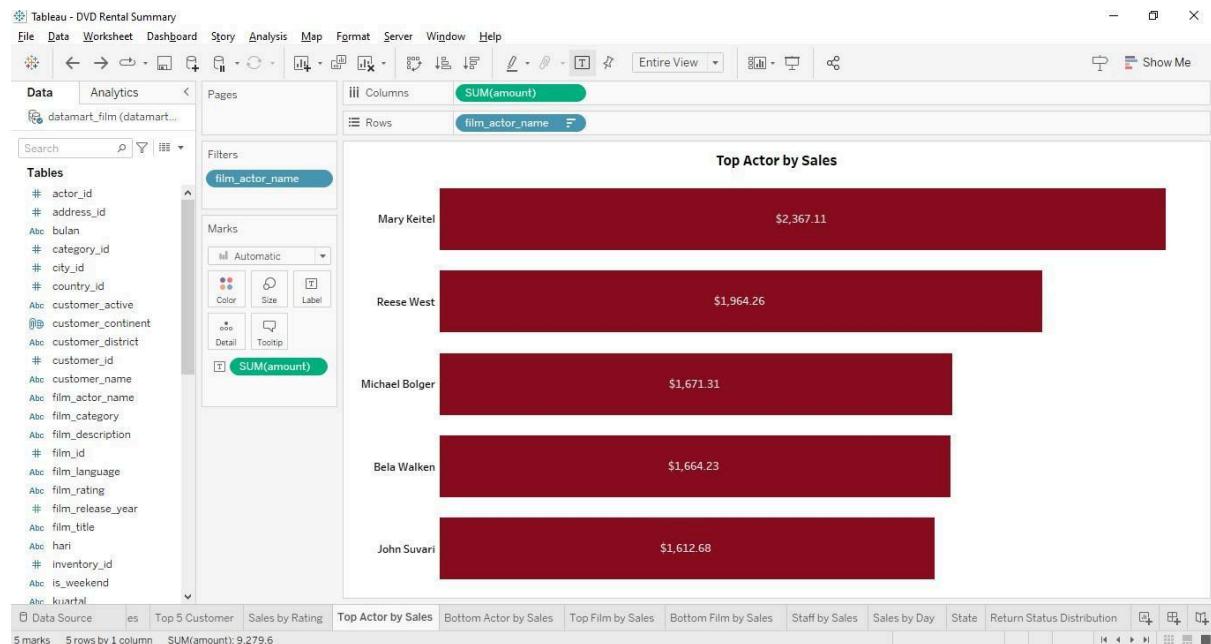
Gambar 2.37 Sales by rating

Kami juga membuat donat chart mengenai penjualan berdasarkan rating film. Ada 5 rating film, yaitu general audience, restricted, parental guidance suggested, no children 17, parents strongly cautioned. Untuk rating film yang paling banyak terjual adalah parents strongly cautioned sebanyak 13 ribu dolar dan yang paling sedikit terjual adalah general audience sebesar 10 ribu dolar.



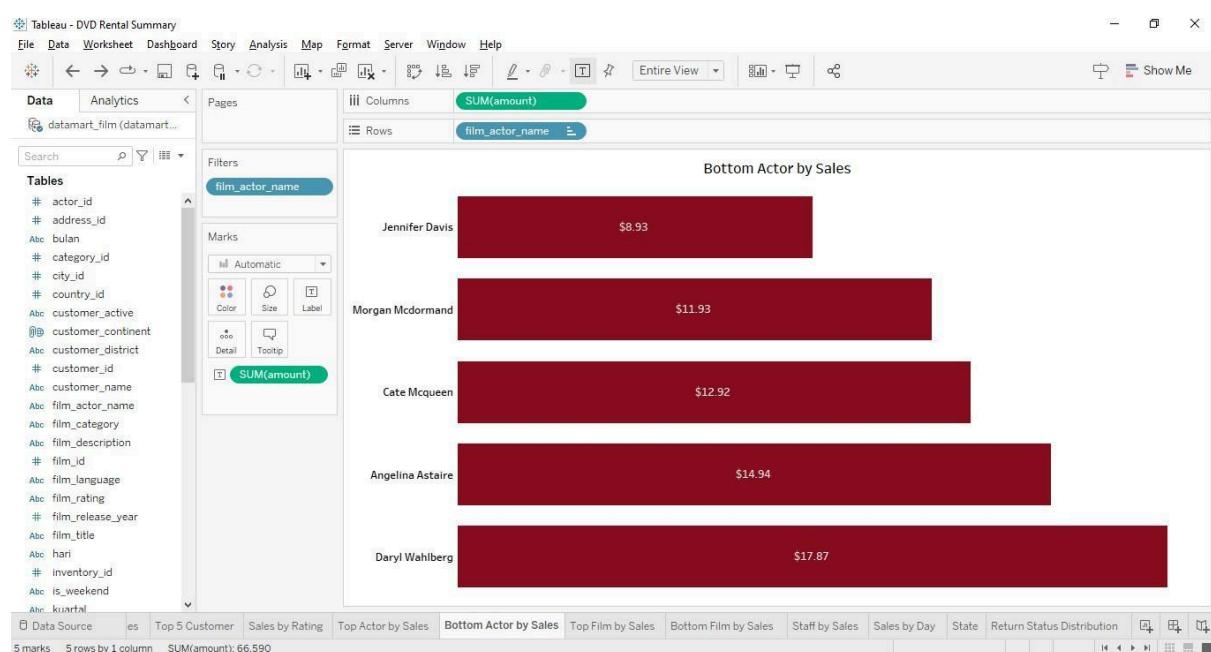
Gambar 2.38 Customers by revenue

Disini kami membuat bar chart berdasarkan pelanggan yang paling banyak menghabiskan uang untuk menyewa DVD. Eleanor hunt berada di peringkat atas karena sudah menghabiskan sebanyak 211 dolar untuk menyewa DVD.



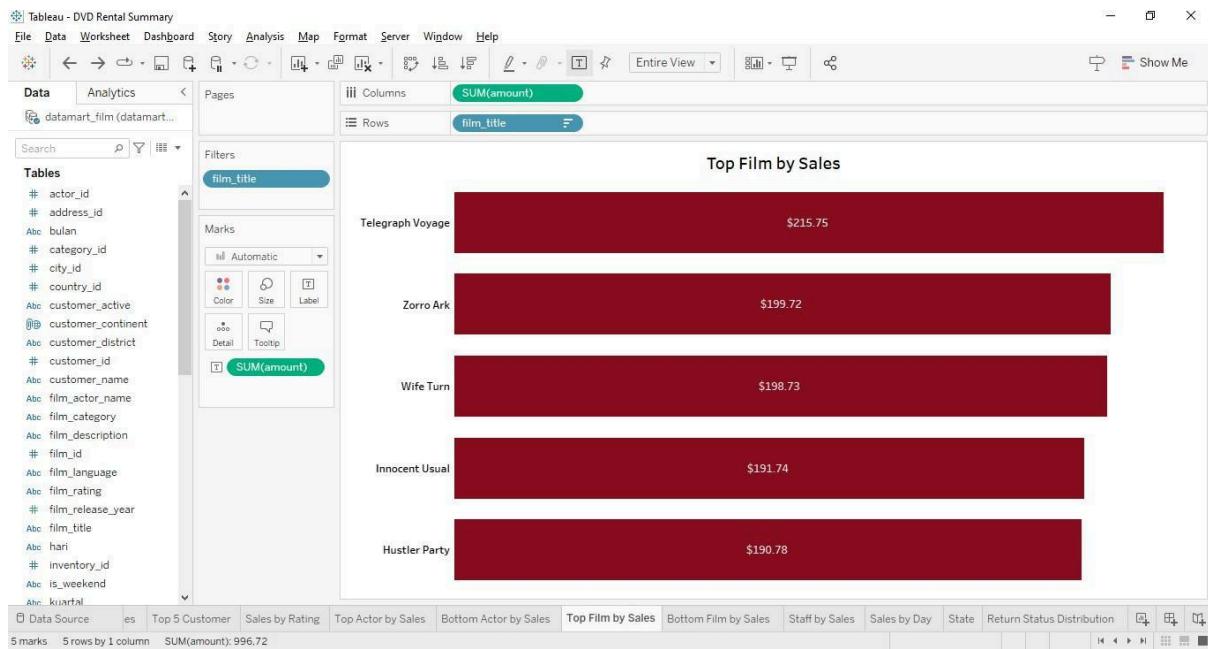
Gambar 2.39 Top actor by sales

Di bar chart ini kami ingin mengetahui aktor yang paling laku penjualannya. Dalam membentuk bar chart tersebut, kami menggunakan tabel film\_actor\_name. Mary Keitel aktor yang menduduki peringkat teratas dengan pendapatan sebesar 2 dollar.

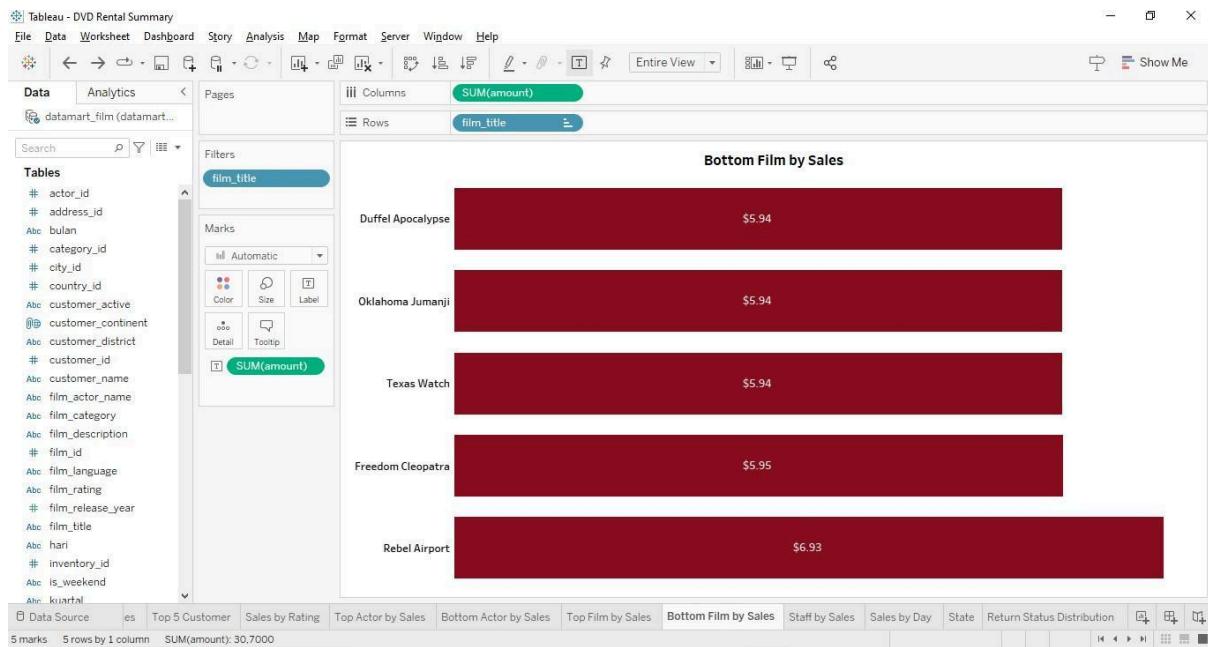


Gambar 2.40 Bottom actor by sales

Untuk *bar chart* di atas berpasangan dengan *bar chart* sebelumnya, dimana kami ingin mengetahui mengenai aktor namun disini kami menunjukkan dengan pendapatan terendah. Aktor dengan pendapatan penyewaan terendah adalah Jennifer Davis dengan pendapatan hanya sebesar 8 dolar.

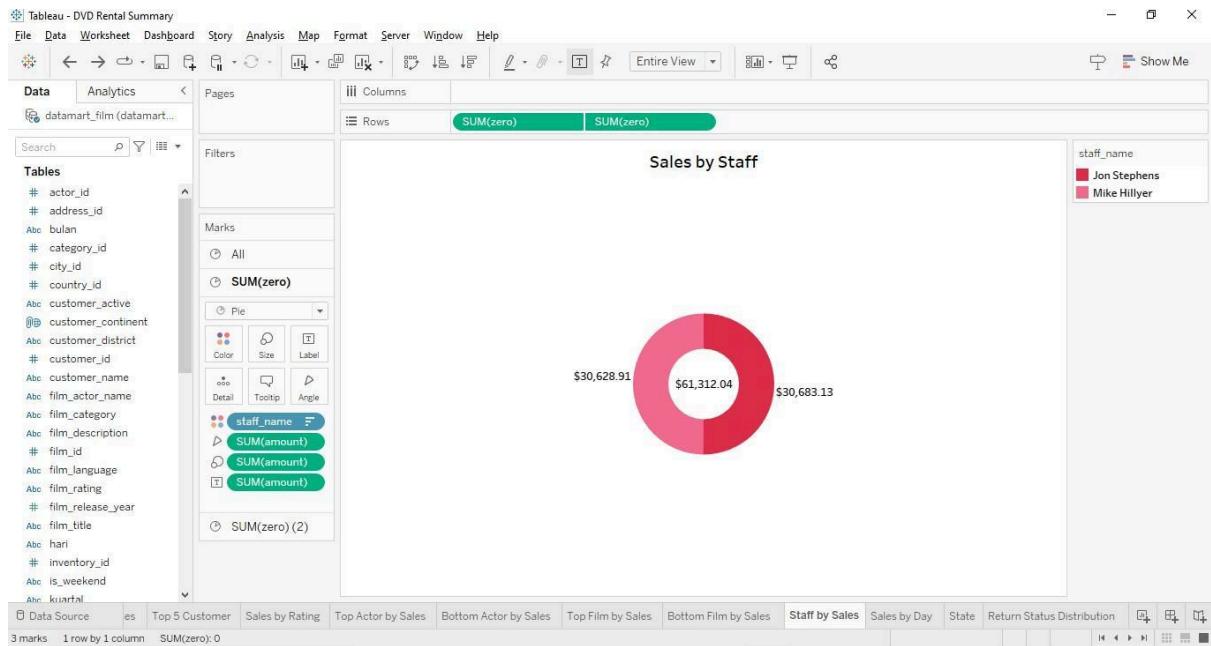


Gambar 2.41 Top film by sales



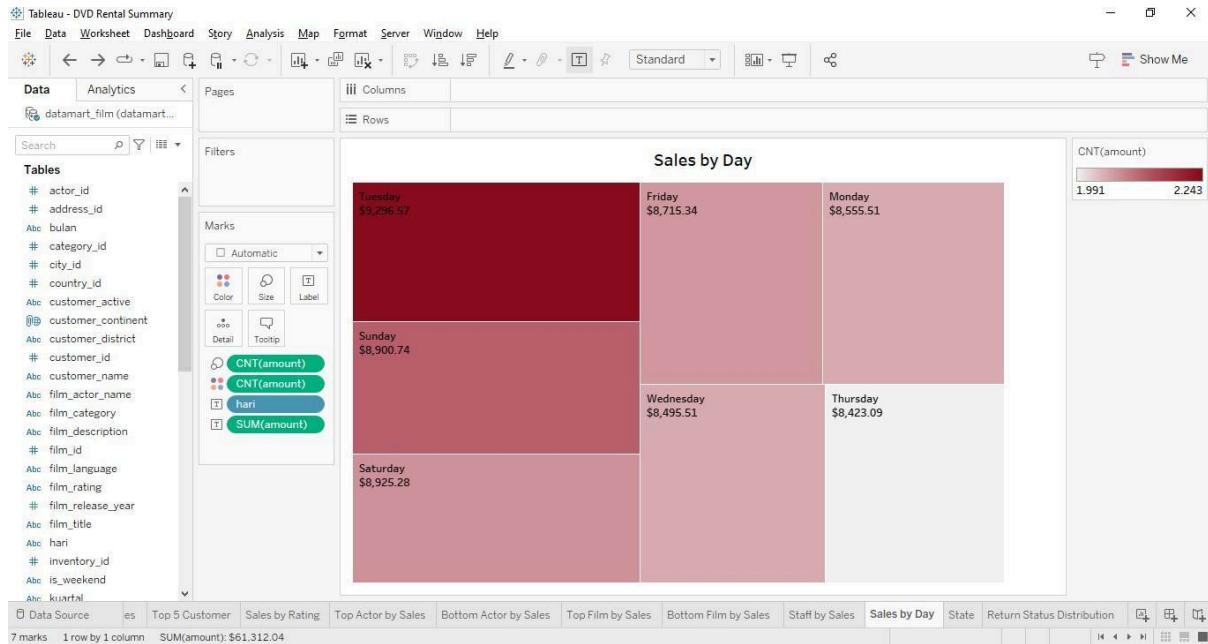
Gambar 2.42 Bottom film by sales

Untuk 2 *bar chart* di atas masih serupa dengan *bar chart* yang sebelumnya, hanya saja *bar chart* ini menjelaskan mengenai film yang paling laku penjualannya. Untuk film yang paling laku penjualannya adalah *Telegraph Voyage* dengan penjualan sebesar 215 dolar. Sedangkan untuk film yang paling sedikit penjualannya adalah *Duffel Apocalypse* yang hanya sebesar 5 dolar.



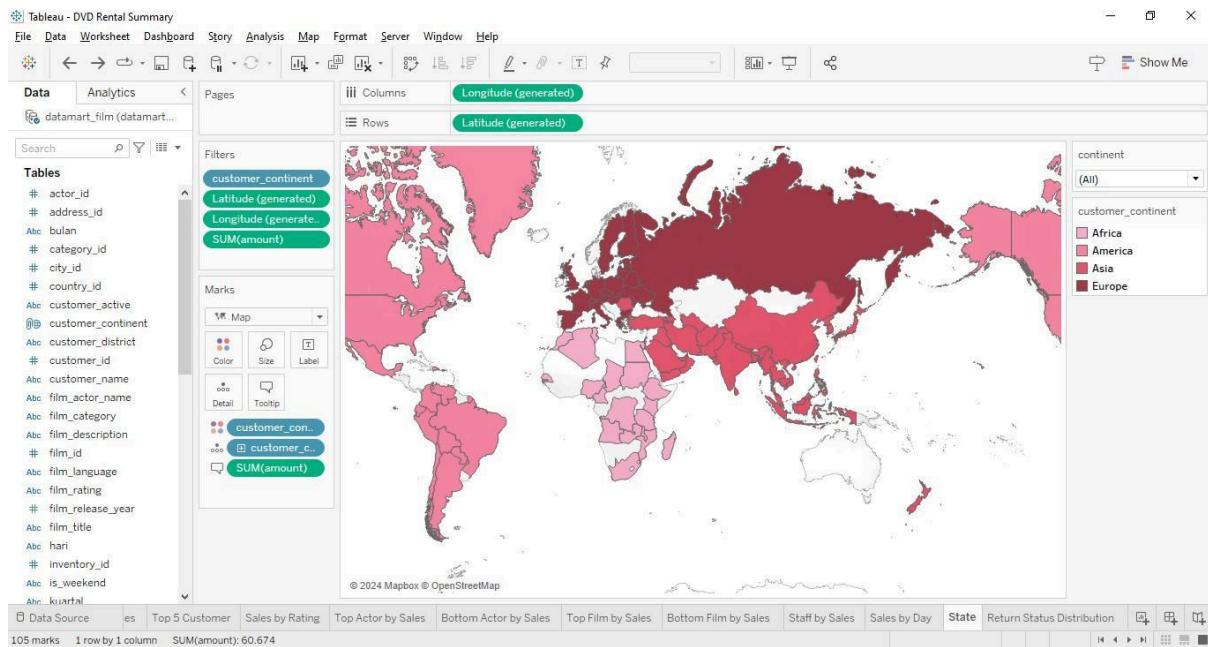
Gambar 2.43 Sales by staff

Untuk donat chart yang kedua, kami ingin mengetahui staff mana yang memiliki kinerja paling baik dengan mengukur dari segi pendapatan yang diterima melalui penyewaan DVD. Staff Jon Stephens unggul tipis dari Mike Hillyer dengan perbedaan sebesar 55 dolar.



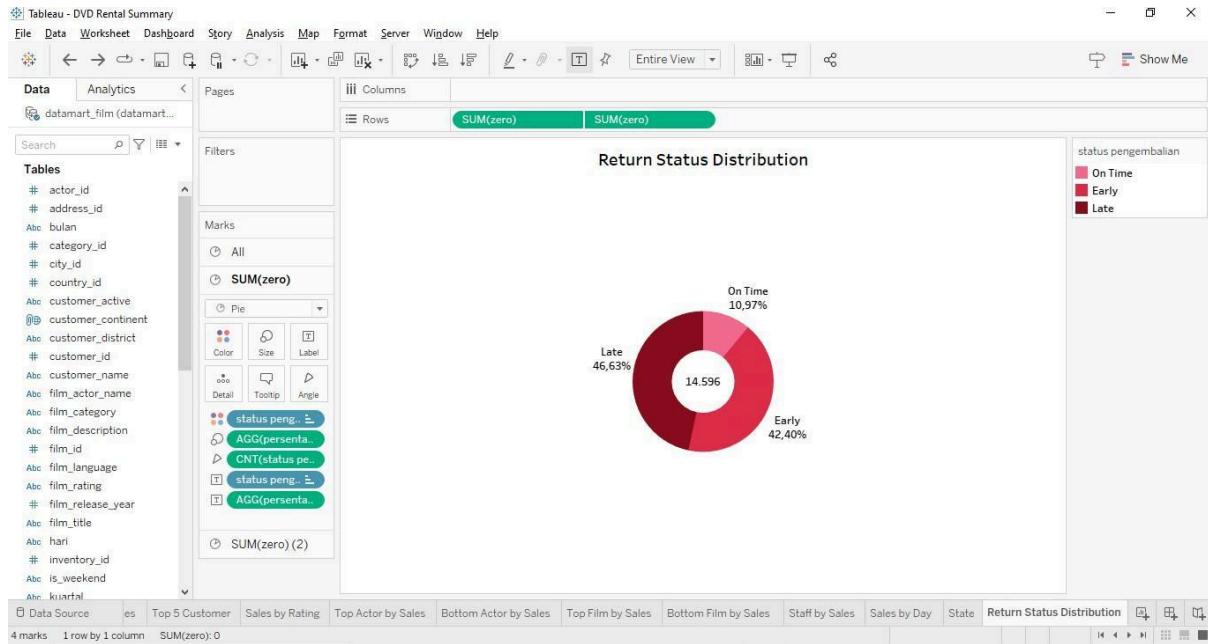
Gambar 2.44 Sales by day

Kami juga melihat besarnya penjualan berdasarkan hari menggunakan area chart. Hari Kamis menjadi hari terbanyak penyewaan DVD sebesar 9 dolar dan hari Selasa menjadi hari paling sedikit adanya penyewaan DVD yang hanya sebesar 8 dolar.



Gambar 2.45 State

Selanjutnya kami membuat peta chart untuk melihat total pelanggan penyewaan DVD di setiap negaranya. Negara yang memiliki paling banyak customer adalah India. Chart peta ini juga berguna untuk filter dari dashboard 1 ke dashboard 2.



Gambar 2.46 Return status by distribution

Dalam penyewaan DVD, data pengembalian DVD yang disewa perlu diperhatikan lebih. Ada 3 kategori dalam pengembalian sewa DVD yaitu, on time, early, dan late. Namun sayangnya, angka pengembalian terlambat (late) yang justru paling tinggi sebesar 46,63%. Hal ini perlu difokuskan dan menjadi evaluasi bagi pengembangan bisnis DVD rental.

## 2.3 Tools dan Teknologi

### 1. PostgreSQL



Gambar 2.47 PostgreSQL

PostgreSQL merupakan sistem manajemen basis data relasional (RDBMS) bersifat open-source yang kuat dan canggih. PostgreSQL sering digunakan untuk menyimpan, mengelola, dan mengambil data dalam berbagai jenis aplikasi, baik skala kecil maupun besar (Munawaroh, 2005). PostgreSQL menjadi solusi database open-source yang fleksibel,

canggih, dan kuat, cocok untuk berbagai kebutuhan mulai dari proyek kecil hingga aplikasi enterprise. Tampilan User Interface (UI) yang nyaman dan mudah digunakan untuk pemula maupun profesional sekaligus fitur yang lengkap untuk administrasi dan pemantauan database menjadikan alasan kami untuk memilih tools ini untuk mendukung pelaksanaan project kelompok kami.

## 2. DBeaver



Gambar 2.48 DBeaver

DBeaver adalah perangkat lunak *database management tool* yang mendukung berbagai jenis basis data, termasuk PostgreSQL, MySQL, Oracle, SQLite, SQL Server, dan banyak lainnya. DBeaver menyediakan antarmuka grafis (GUI) yang memudahkan pengguna untuk mengelola, menavigasi, dan melakukan operasi pada database. Dengan fitur-fitur canggih, antarmuka yang intuitif, fleksibilitas tinggi, dan sistem open source DBeaver menjadi pilihan ideal untuk pengembang, administrator database, analis data, dan mendukung pelaksanaan project kelompok kami.

## 3. Pentaho



Gambar 2.49 Pentaho

Pentaho adalah perangkat lunak *Business Intelligence* (BI) dan *Data Integration* (ETL) yang digunakan untuk menganalisis, mengolah, dan memvisualisasikan data dari berbagai sumber. Dengan kombinasi fitur open-source dan enterprise, Pentaho memberikan fleksibilitas tinggi untuk mengolah, menganalisis, dan memvisualisasikan data dalam mendukung pengambilan keputusan strategis (Hasan dan Sudaryana, 2022).

#### 4. Tableau



Gambar 2.50 Tableau

Tableau adalah perangkat lunak *Business Intelligence* (BI) dan *data visualization* yang digunakan untuk menganalisis, memvisualisasikan, dan membagikan data dalam bentuk yang mudah dipahami (Puteri dan Suci Rahma, 2022). Dengan dukungan yang luas untuk berbagai sumber data, antarmuka yang ramah pengguna, dan kemampuan kolaborasi menjadikan alat pendukung untuk kemudahan project kelompok kami.

## BAB 3

### IMPLEMENTASI DAN HASIL

#### 3.1 Input Database

Dari yang sudah dijelaskan pada BAB 2 mengenai proses input database ke RDBMS, sejauh ini belum ada kendala dan error yang ditemukan. Kami juga menggunakan database DVD Rental yang dimana sudah tersedia dalam bentuk file, berbeda dengan database Chinook yang harus menjalankan query terlebih dahulu di Dbeaver untuk membentuk databasenya. Saat kami input database DVD ke postgreSQL, juga berhasil dan tidak failed.

Tabel-tabel yang ada di database DVD Rental antara lain tabel actor, address, category, city, country, customer, film, film\_actor, film\_category, inventory, language, payment, rental, staff, store. Kami menggunakan semua tabel tersebut untuk di visualisasi agar analisis menjadi lebih efektif dan relevan. Alasan kelompok kami memilih database DVD rental karena menurut kamu dari segi struktur tabelnya lebih mudah untuk kami analisis. Kami sudah mengetahui ingin melakukan apa saat awal kami melihat databasenya.

#### 3.2 Pembuatan Staging Area

Pada BAB 2 kami juga sudah menjelaskan secara rinci mengenai proses staging area. Kami menjelaskan step-step dari Dbeaver hingga ke Pentaho dalam pembuatan database staging. Namun dalam proses staging tidak semulus yang kami bayangkan. Kelompok kami mengalami beberapa kendala dan error saat proses staging di Pentaho.



Gambar 3.1 Error last\_updated

Error pertama yang kami hadapi adalah error saat mau membuat kerangka struktur tabel kosong di database staging. Error tersebut mengatakan bahwa fungsi last\_updated tidak ditemukan.

```

-- public.actor definition
-- Drop table
-- DROP TABLE public.actor;

CREATE TABLE public.actor (
    actor_id serial NOT NULL,
    first_name varchar(45) NOT NULL,
    last_name varchar(45) NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT actor_pkey PRIMARY KEY (actor_id)
);
CREATE INDEX idx_actor_last_name ON public.actor USING btree (last_name);

-- Table Triggers
-- --create trigger last_updated before
-- --update
-- -- on
-- -- public.actor for each row execute function last_updated();

-- public.category definition
-- Drop table
-- DROP TABLE public.category;

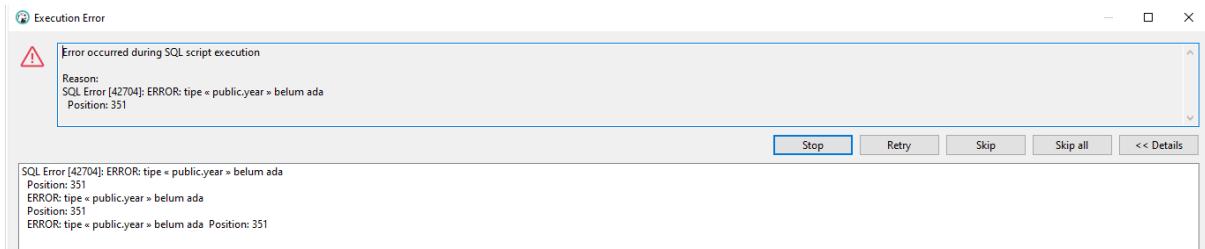
CREATE TABLE public.category (
    category_id serial NOT NULL,
    "name" varchar(25) NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT category_pkey PRIMARY KEY (category_id)
);
CREATE INDEX idx_category_name ON public.category USING btree ("name");

-- Table Triggers
-- --create trigger last_updated before
-- --update
-- -- on
-- -- public.category for each row execute function last_updated();

```

Gambar 3.2 Menghapus query table triggers

Untuk mengatasi *error* fungsi *last\_update* tidak ada, query bagian table triggers harus dihapus. Tidak hanya satu query, namun semua query table triggers harus dihapus sampai tidak tersisa lagi di *script*.



Gambar 3.3 Error public year

Jika di *running* setelah table triggers dihapus, akan muncul error baru yang bertuliskan *public.year* belum ada.

```

-- Drop table
-- DROP TABLE public.film;

CREATE TABLE public.film (
    film_id serial NOT NULL,
    title varchar(255) NOT NULL,
    description text NULL,
    release_year public.year NULL,
    language_id int NOT NULL,
    rental_duration int2 DEFAULT 3 NOT NULL,
    rental_rate numeric(4, 2) DEFAULT 4.99 NOT NULL,
    length int2 NULL,
    replacement_cost numeric(5, 2) DEFAULT 19.99 NOT NULL,
    rating public.mpaa_rating DEFAULT 'G'::mpaa_rating NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    special_features text NULL,
    full_text_vector vector NULL,
    CONSTRAINT film_pkey PRIMARY KEY (film_id),
    CONSTRAINT film_language_id_fkey FOREIGN KEY (language_id) REFERENCES public."language"(language_id) ON DELETE RESTRICT ON UPDATE CASCADE
);
CREATE INDEX film_fulltext_idx ON public.film USING gist (fulltext);
CREATE INDEX idx_fk_language_id ON public.film USING btree (language_id);
CREATE INDEX idx_title ON public.film USING btree (title);

```

Gambar 3.4 Mengganti tipe public year

Solusi mengatasinya adalah dengan mengganti tipe *public.year* pada kolom *film* dan ganti menjadi *integer*

```

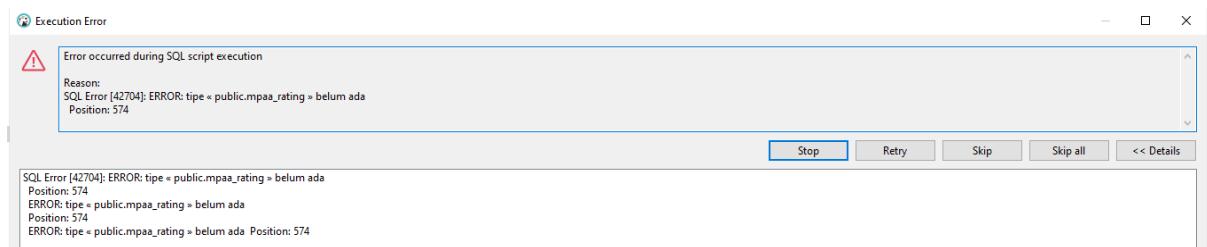
-- Drop table
-- DROP TABLE public.film;

CREATE TABLE public.film (
    film_id serial4 NOT NULL,
    title varchar(255) NOT NULL,
    description text NULL,
    release_year integer(4) NOT NULL,
    language_id int2 NOT NULL,
    rental_duration int2 DEFAULT 3 NOT NULL,
    rental_rate numeric(4, 2) DEFAULT 4.99 NOT NULL,
    length int2 NULL,
    replacement_cost numeric(5, 2) DEFAULT 19.99 NOT NULL,
    rating varchar(2) NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    special_features _text NULL,
    fulltext tsvector NOT NULL,
    CONSTRAINT film_pk PRIMARY KEY (film_id),
    CONSTRAINT film_language_id_fkey FOREIGN KEY (language_id) REFERENCES public."language"(language_id) ON DELETE RESTRICT ON UPDATE CASCADE
);
CREATE INDEX film_fulltext_idx ON public.film USING gist (fulltext);
CREATE INDEX idx_fk_language_id ON public.film USING btree (language_id);
CREATE INDEX idx_title ON public.film USING btree (title);

```

Gambar 3.5 Mengganti tipe public year

Setelah mengganti tipe public year menjadi integer, jika di *running* lagi maka akan muncul *error* baru.



Gambar 3.6 Error public mpaa rating

*Error* baru tersebut berbunyi public.mpaa\_rating belum ada.

```

-- Drop table
-- DROP TABLE public.film;

CREATE TABLE public.film (
    film_id serial4 NOT NULL,
    title varchar(255) NOT NULL,
    description text NULL,
    release_year integer NULL,
    language_id int2 NOT NULL,
    rental_duration int2 DEFAULT 3 NOT NULL,
    rental_rate numeric(4, 2) DEFAULT 4.99 NOT NULL,
    length int2 NULL,
    replacement_cost numeric(5, 2) DEFAULT 19.99 NOT NULL,
    rating varchar(2) NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    special_features _text NULL,
    fulltext tsvector NOT NULL,
    CONSTRAINT film_pk PRIMARY KEY (film_id),
    CONSTRAINT film_language_id_fkey FOREIGN KEY (language_id) REFERENCES public."language"(language_id) ON DELETE RESTRICT ON UPDATE CASCADE
);
CREATE INDEX film_fulltext_idx ON public.film USING gist (fulltext);
CREATE INDEX idx_fk_language_id ON public.film USING btree (language_id);
CREATE INDEX idx_title ON public.film USING btree (title);

```

Gambar 3.7 mengganti tipe data public mpaa rating

Untuk mengatasi *error* adalah dengan mengganti tipe datanya seperti pada gambar di atas.

The screenshot shows the DBeaver Database Navigator interface. On the left, the 'Tables' section of the 'public' schema is visible, listing various tables like actor, address, category, city, country, customer, film, film\_actor, film\_category, inventory, language, payment, rental, staff, and store. In the center, a script editor window titled 'Script-19' contains the SQL code for creating the 'film' table. The code includes columns such as film\_id, title, description, release\_year, language\_id, rental\_duration, rental\_rate, replacement\_cost, rating, last\_update, special\_features, and fulltext. It also includes indexes and triggers. On the right, a 'Statistics 1' panel displays performance metrics for the query.

```

CREATE TABLE public.film (
    film_id serial4 NOT NULL,
    title character(55) NOT NULL,
    description text NOT NULL,
    release_year integer NOT NULL,
    language_id int4 NOT NULL,
    rental_duration int2 DEFAULT 3 NOT NULL,
    rental_rate numeric(4, 2) DEFAULT 4.99 NOT NULL,
    length int4 NOT NULL,
    replacement_cost numeric(5, 2) DEFAULT 19.99 NOT NULL,
    rating varchar DEFAULT 'G' NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    special_features _text NOT NULL,
    fulltext _text NOT NULL,
    CONSTRAINT film_pk PRIMARY KEY (film_id),
    CONSTRAINT film_language_id_fkey FOREIGN KEY (language_id) REFERENCES public."language"(language_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

CREATE INDEX film_fulltext_idx ON public.film USING gist (fulltext);
CREATE INDEX idx_fk_language_id ON public.film USING btree (language_id);
CREATE INDEX idx_title ON public.film USING btree (title);

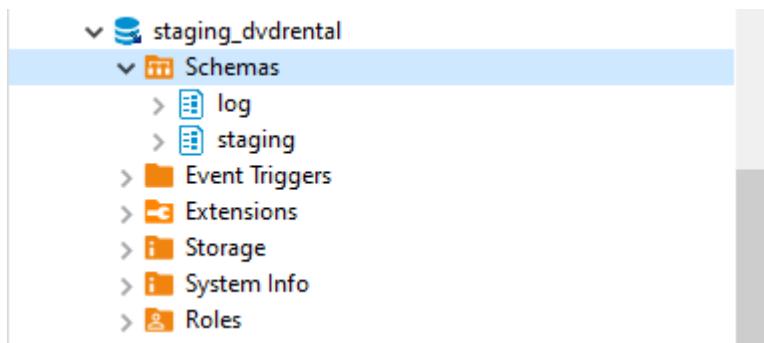
-- Table Triggers
-- --create trigger film_fulltext_trigger before
-- --insert
-- --or
-- --update
-- --on

```

Name	Value
Queries	32
Updated Rows	0
Execute time (ms)	1121
Fetch time (ms)	0
Total time (ms)	1121
Start time	2024-12-04 18:01:34.437
Finish time	2024-12-04 18:01:35.592

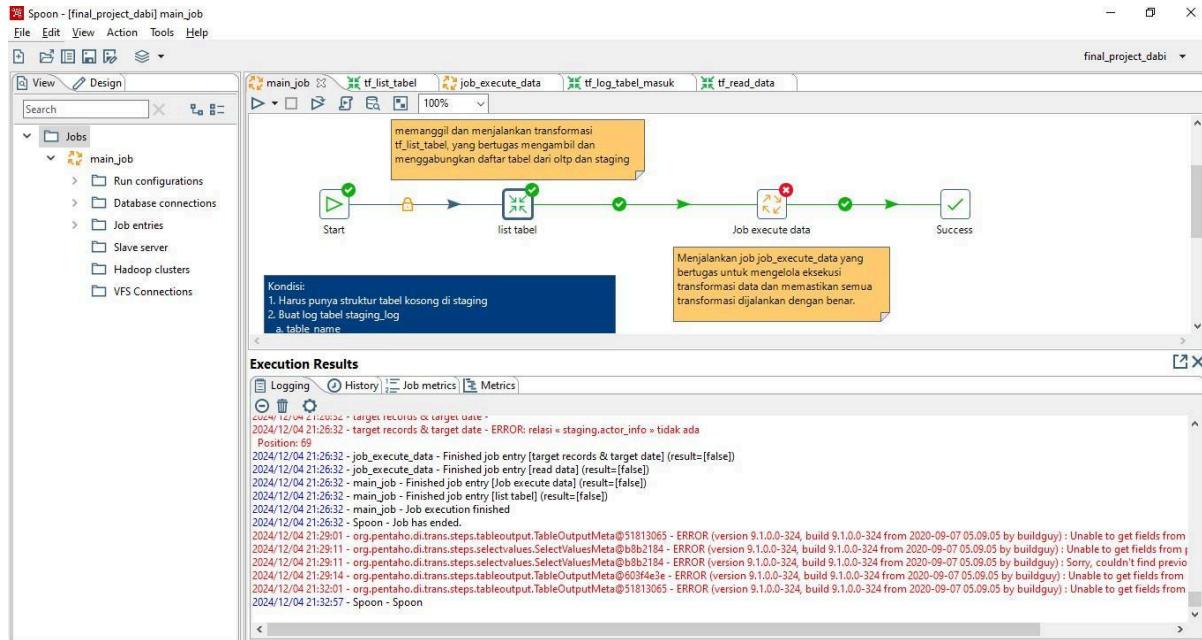
Gambar 3.8 Berhasil membuat kerangka tabel

Setelah mengatasi beberapa *error*, akhirnya kami berhasil membuat kerangka struktur tabel di database staging. Setelah itu kami me-*rename* schema oublin menjadi staging dan membuat schema baru dengan nama log.

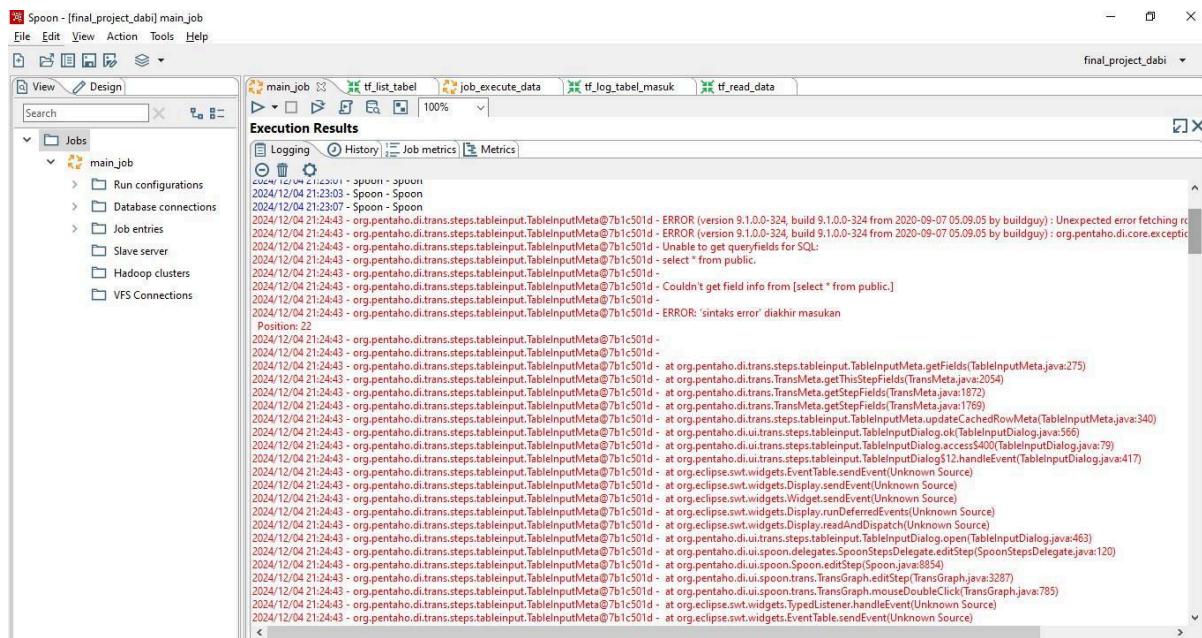


Gambar 3.9 membuat schema log

Selain error yang kami alami di Dbeaver, kami juga mengalami banyak error saat proses staging di Pentaho. Berikut adalah gambar dari beberapa error yang kami alami.



Gambar 3.10 Error staging pentaho



Gambar 3.11 Error staging pentaho

```

Spoon - [final_project_dabi] main.job
File Edit View Action Tools Help
main_job tf_list_table job_execute_data tf_log_table_masuk tf_read_data
Execution Results
Logging History Job metrics Metrics
2024/12/04 21:26:29 - Select value:0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2024/12/04 21:26:29 - staging:0 - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05:09:05 by buildguy) : Unexpected error
2024/12/04 21:26:29 - staging:0 - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05:09:05 by buildguy) : org.pentaho.di.core.exception.KettleDatabaseException:
2024/12/04 21:26:29 - staging:0 - Couldn't execute SQL: TRUNCATE TABLE staging.actor
2024/12/04 21:26:29 - staging:0 -
2024/12/04 21:26:29 - staging:0 - ERROR: tidak dapat memotong referensi tabel dalam constraint foreign key
Detail: Tabel « film_actor » mengacu pada « actor ».
Hin: Pemotongan tabel « film_actor » pada waktu yang sama, atau gunakan TRUNCATE ... CASCADE.
2024/12/04 21:26:29 - staging:0 - at org.pentaho.di.core.database.Database.execStatement(Database.java:1568)
2024/12/04 21:26:29 - staging:0 - at org.pentaho.di.core.database.Database.execStatement(Database.java:1518)
2024/12/04 21:26:29 - staging:0 - at org.pentaho.di.core.database.Database.truncateTable(Database.java:3316)
2024/12/04 21:26:29 - staging:0 - at org.pentaho.di.trans.steps.tableoutput.TableOutput.truncateTable(TableOutput.java:567)
2024/12/04 21:26:29 - staging:0 - at org.pentaho.di.trans.steps.tableoutput.TableOutput.processRow(TableOutput.java:86)
2024/12/04 21:26:29 - staging:0 - at org.pentaho.di.trans.step.RunThread.run(RunThread.java:62)
2024/12/04 21:26:29 - staging:0 - at java.lang.Thread.run(Unknown Source)
2024/12/04 21:26:29 - staging:0 - Caused by: org.postgresql.util.PSQLException: ERROR: tidak dapat memotong referensi tabel dalam constraint foreign key
Detail: Tabel « film_actor » mengacu pada « actor ».
Hin: Pemotongan tabel « film_actor » pada waktu yang sama, atau gunakan TRUNCATE ... CASCADE.
2024/12/04 21:26:29 - staging:0 - at org.pentaho.di.core.database.QueryExecutorImpl.executeError(QueryExecutorImpl.java:2440)
2024/12/04 21:26:29 - staging:0 - at org.postgresql.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:2183)
2024/12/04 21:26:29 - staging:0 - at org.postgresql.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:308)
2024/12/04 21:26:29 - staging:0 - at org.postgresql.jdbc.PgStatement.executeUpdateInternal(PgStatement.java:441)
2024/12/04 21:26:29 - staging:0 - at org.postgresql.jdbc.PgStatement.execute(PgStatement.java:365)
2024/12/04 21:26:29 - staging:0 - at org.postgresql.jdbc.PgStatement.executeWithFlags(PgStatement.java:307)
2024/12/04 21:26:29 - staging:0 - at org.postgresql.jdbc.PgStatement.executeCachedSql(PgStatement.java:293)
2024/12/04 21:26:29 - staging:0 - at org.postgresql.jdbc.PgStatement.executeWithFlags(PgStatement.java:270)
2024/12/04 21:26:29 - staging:0 - at org.postgresql.jdbc.PgStatement.execute(PgStatement.java:266)
2024/12/04 21:26:29 - staging:0 - at org.pentaho.di.core.database.Database.execStatement(Database.java:1543)
2024/12/04 21:26:29 - staging:0 - . . more
2024/12/04 21:26:29 - staging:0 - Finished processing (I=0, O=0, R=1, W=0, U=0, E=1)
2024/12/04 21:26:29 - tf_read_data - Transformation detected one or more steps with errors.
2024/12/04 21:26:29 - tf_read_data - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05:09:05 by buildguy) : Errors detected!

```

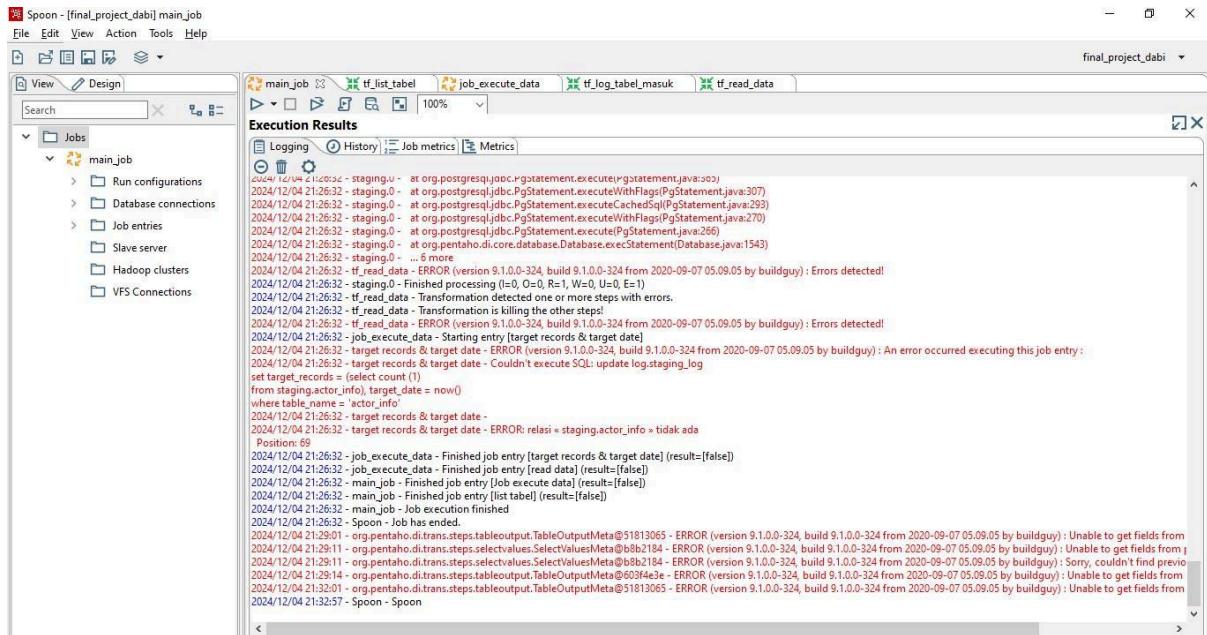
Gambar 3.12 Error staging pentaho

```

Spoon - [final_project_dabi] main.job
File Edit View Action Tools Help
main_job tf_list_table job_execute_data tf_log_table_masuk tf_read_data
Execution Results
Logging History Job metrics Metrics
2024/12/04 21:26:32 - output:0 - rimnea processing (I=200, O=0, R=0, W=200, U=0, E=0)
2024/12/04 21:26:32 - staging:0 - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05:09:05 by buildguy) : Unexpected error
2024/12/04 21:26:32 - staging:0 - Couldnt't execute SQL: TRUNCATE TABLE staging.actor_info
2024/12/04 21:26:32 - staging:0 - ERROR: relasi « staging.actor_info » tidak ada
2024/12/04 21:26:32 - staging:0 - at org.pentaho.di.core.database.Database.execStatement(Database.java:1568)
2024/12/04 21:26:32 - staging:0 - at org.pentaho.di.core.database.Database.execStatement(Database.java:1518)
2024/12/04 21:26:32 - staging:0 - at org.pentaho.di.core.database.Database.truncateTable(Database.java:3316)
2024/12/04 21:26:32 - staging:0 - at org.pentaho.di.trans.steps.tableoutput.TableOutput.truncateTable(TableOutput.java:567)
2024/12/04 21:26:32 - staging:0 - at org.pentaho.di.trans.steps.tableoutput.TableOutput.processRow(TableOutput.java:86)
2024/12/04 21:26:32 - staging:0 - at org.pentaho.di.trans.step.RunThread.run(RunThread.java:62)
2024/12/04 21:26:32 - staging:0 - at java.lang.Thread.run(Unknown Source)
2024/12/04 21:26:32 - staging:0 - Caused by: org.postgresql.util.PSQLException: ERROR: relasi « staging.actor_info » tidak ada
2024/12/04 21:26:32 - staging:0 - at org.pentaho.di.core.database.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2440)
2024/12/04 21:26:32 - staging:0 - at org.postgresql.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:2183)
2024/12/04 21:26:32 - staging:0 - at org.postgresql.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:308)
2024/12/04 21:26:32 - staging:0 - at org.postgresql.jdbc.PgStatement.executeUpdateInternal(PgStatement.java:441)
2024/12/04 21:26:32 - staging:0 - at org.postgresql.jdbc.PgStatement.execute(PgStatement.java:365)
2024/12/04 21:26:32 - staging:0 - at org.postgresql.jdbc.PgStatement.executeWithFlags(PgStatement.java:307)
2024/12/04 21:26:32 - staging:0 - at org.postgresql.jdbc.PgStatement.executeCachedSql(PgStatement.java:293)
2024/12/04 21:26:32 - staging:0 - at org.postgresql.jdbc.PgStatement.executeWithFlags(PgStatement.java:270)
2024/12/04 21:26:32 - staging:0 - at org.postgresql.jdbc.PgStatement.execute(PgStatement.java:266)
2024/12/04 21:26:32 - staging:0 - at org.pentaho.di.core.database.Database.execStatement(Database.java:1543)
2024/12/04 21:26:32 - staging:0 - . . more
2024/12/04 21:26:32 - tf_read_data - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05:09:05 by buildguy) : Errors detected!
2024/12/04 21:26:32 - staging:0 - Finished processing (I=0, O=0, R=1, W=0, U=0, E=1)
2024/12/04 21:26:32 - tf_read_data - Transformation detected one or more steps with errors.
2024/12/04 21:26:32 - tf_read_data - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05:09:05 by buildguy) : Errors detected!
2024/12/04 21:26:32 - job_execute_data - Starting entry [target records & target date]
2024/12/04 21:26:32 - target records & target date - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05:09:05 by buildguy) : An error occurred executing this job entry:
2024/12/04 21:26:32 - target records & target date - Couldn't execute SQL: update log.staging_log

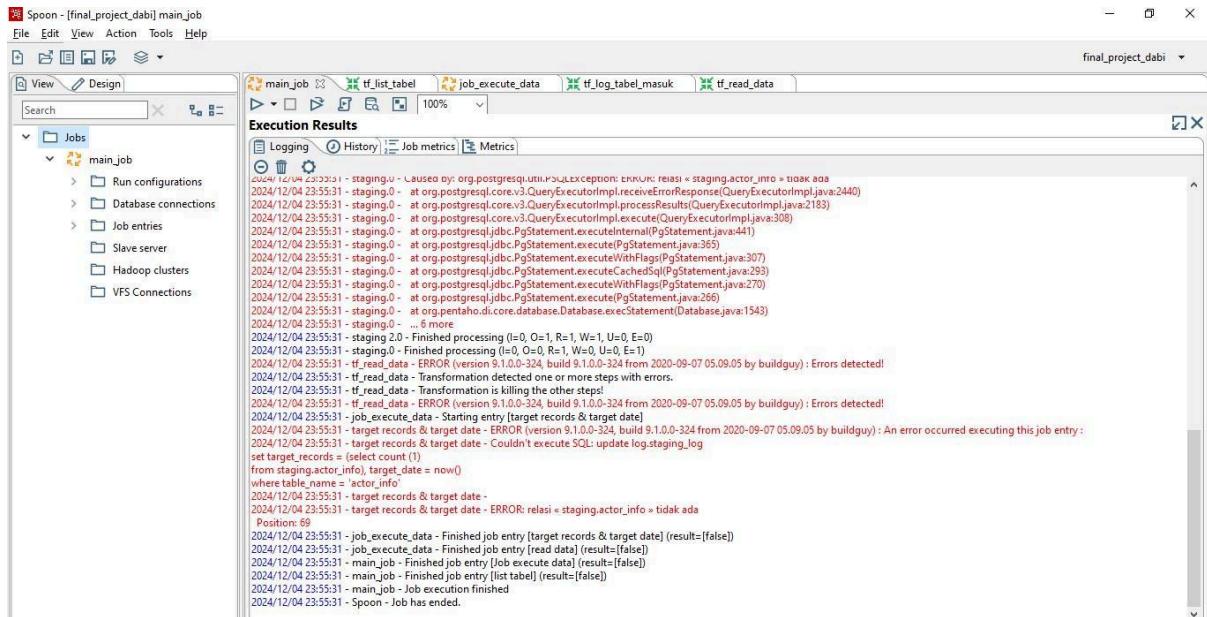
```

Gambar 3.13 Error staging pentaho



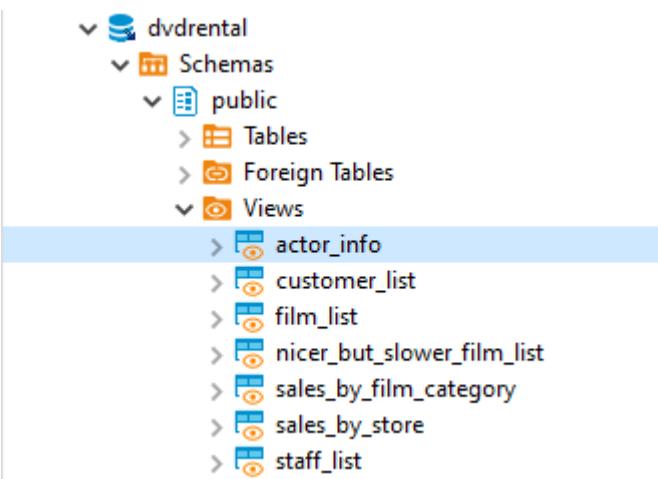
Gambar 3.14 Error staging pentaho

Untuk mengatasi error seperti pada gambar di atas, kami menghapus foreign key tabel yang ada dalam staging.



Gambar 3.15 Error berkurang saat foreign key dihapus

Error berkurang saat kami menghapus foreign key tabel. Untuk mengatasi error yang masih ada kami menghapus tabel *views* yang terdapat di database DVD Rental. Lalu uncekis *district* dan *phone* di staging kemudian *save*. Kami juga menghapus index yang ada *fulltext* di staging, mengubah tipe data *fulltext* dan *special\_features* di staging.



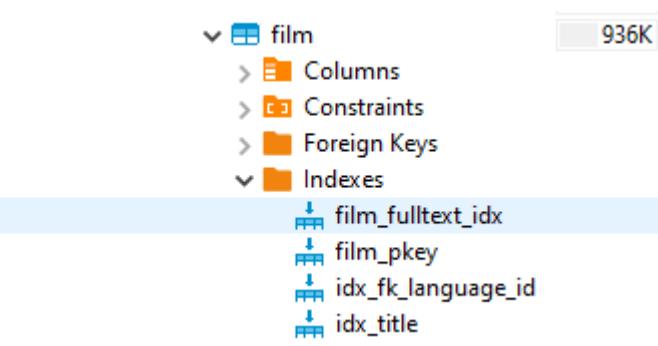
Gambar 3.16 Menghapus tabel views

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
address_id	1	serial4			[v]	nextval('addre...')	
address	2	varchar(50)		default	[v]		
address2	3	varchar(50)		default	[ ]		
district	4	varchar(20)		default	[v]		
city_id	5	int2			[v]		
postal_code	6	varchar(10)		default	[ ]		
phone	7	varchar(20)		default	[v]		
last_update	8	timestamp			[v]	now()	

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
address_id	1	serial4			[v]	nextval('staging...')	
address	2	varchar(50)		default	[v]		
address2	3	varchar(50)		default	[ ]		
district	4	varchar(20)		default	[ ]		
city_id	5	int2			[v]		
postal_code	6	varchar(10)		default	[ ]		
phone	7	varchar(20)		default	[ ]		
last_update	8	timestamp			[v]	now()	

Gambar 3.17 Uncekis district dan phone



Gambar 3.18 Menghapus Indexes

```

-- mengubah tipe data
ALTER TABLE film
ALTER COLUMN special_features TYPE varchar USING array_to_string(special_features, ', ');

-- ALTER TABLE film
ALTER COLUMN fulltext TYPE varchar(255) USING fulltext::text;

```

**Statistics 1**

Name	Value
Updated Rows	0
Query	ALTER TABLE film ALTER COLUMN special_features TYPE varchar USING array_to_string(special_features, ', '); ALTER TABLE film ALTER COLUMN fulltext TYPE varchar(255) USING fulltext::text
Start time	Thu Dec 05 09:20:28 WIB 2024
Finish time	Thu Dec 05 09:20:28 WIB 2024

Gambar 3.19 Merunning query

**address**

Grid	1 address_id	abc address	abc address2	abc district	123 city_id	abc postal_code	abc phone	last_update
1	1	47 MySakila Drive	[NULL]	Alberta	300			2006-02-15 09:45:30.000
2	2	28 MySQL Boulevard	[NULL]	QLD	576			2006-02-15 09:45:30.000
3	3	23 Workhaven Lane	[NULL]	Alberta	300		1403335568	2006-02-15 09:45:30.000
4	4	1411 Lillydale Drive	[NULL]	QLD	576		6172235589	2006-02-15 09:45:30.000
5	5	1913 Hanoi Way		Nagasaki	463	35200	28303384290	2006-02-15 09:45:30.000
6	6	1121 Loja Avenue		California	449	17886	838635286649	2006-02-15 09:45:30.000
7	7	692 Joliet Street		Attika	38	83579	448477190408	2006-02-15 09:45:30.000
8	8	1566 Ingle Manor		Mandalay	349	53561	705814003527	2006-02-15 09:45:30.000
9	9	53 Idfu Parkway		Nantou	361	42399	10655648674	2006-02-15 09:45:30.000
10	10	1795 Santiago de Compostela Way		Texas	295	18743	860452626434	2006-02-15 09:45:30.000
11	11	900 Santiago de Compostela Parkway		Central Serbia	280	93896	716571220373	2006-02-15 09:45:30.000
12	12	478 Joliet Way		Hamilton	200	77948	657282285970	2006-02-15 09:45:30.000
13	13	613 Korolev Drive		Masqat	329	45844	380657522649	2006-02-15 09:45:30.000
14	14	1531 Sal Drive		Esfahan	162	53628	648856936185	2006-02-15 09:45:30.000
15	15	1542 Tarlac Parkway		Kanagawa	440	1027	635297277345	2006-02-15 09:45:30.000
16	16	808 Bhopal Manor		Haryana	582	10672	465887807014	2006-02-15 09:45:30.000
17	17	270 Amroha Parkway		Osmaniye	384	29610	695479675738	2006-02-15 09:45:30.000
18	18	770 Bydgoszcz Avenue		California	120	16266	51733814235	2006-02-15 09:45:30.000
19	19	419 Iligan Lane		Madhya Pradesh	76	72878	990911107354	2006-02-15 09:45:30.000
20	20	360 Toulouse Parkway		England	495	54308	949312333307	2006-02-15 09:45:30.000
21	21	270 Toulon Boulevard		Kalmykia	156	81766	407752414682	2006-02-15 09:45:30.000
22	22	320 Brest Avenue		Kaduna	252	43331	747791594069	2006-02-15 09:45:30.000
23	23	1417 Lancaster Avenue		Northern Cape	267	72192	272572357893	2006-02-15 09:45:30.000
24	24	1688 Okara Way		Nothwest Border Prov	327	21954	144453869132	2006-02-15 09:45:30.000

Gambar 3.20 Terdapat null di kolom address

Setelah kami perhatikan lagi, ternyata di dalam tabel address terdapat kolom address yang tidak memiliki value/null.

```


--cek tabel address yang ternyata ada kolom address2 yang isinya null/kosong
SELECT COUNT(address) FROM address;
SELECT COUNT(address2) FROM address;

--Mengisi Nilai NULL Sementara dengan N/A
SELECT
    COALESCE(address, 'N/A') AS kolom1,
    COALESCE(address2, 'N/A') AS kolom2
FROM address;

-- menggunakan kombinasi COALESCE dan NULLIF untuk mendeteksi string kosong sebagai NULL
SELECT
    COALESCE(NULLIF(address, ''), 'N/A') AS kolom1,
    COALESCE(NULLIF(address2, ''), 'N/A') AS kolom2
FROM address;

-- 1. COALESCE(NULLIF(address, ''), 'N/A')
--> Mengubah nilai kosong ('') menjadi NULL, lalu mengganti nilai NULL menjadi N/A.
-- 2. CASE WHEN ... THEN 1 ELSE 0 END
--> Memeriksa apakah hasil transformasi sama dengan N/A. Jika iya, hitung sebagai 1; jika tidak, hitung sebagai 0.
-- 3. SUM...
--> Menjumlahkan seluruh 1 untuk mendapatkan jumlah N/A di masing-masing kolom.

SELECT
    SUM(CASE WHEN COALESCE(NULLIF(address, ''), 'N/A') = 'N/A' THEN 1 ELSE 0 END) AS jumlah_kolom1,
    SUM(CASE WHEN COALESCE(NULLIF(address2, ''), 'N/A') = 'N/A' THEN 1 ELSE 0 END) AS jumlah_kolom2
FROM address;


```

Results 1 ×

	123 jumlah_kolom1	123 jumlah_kolom2
1	0	603

Gambar 3.21 mengatasi null

Lalu kami mencari *insight* dan memastikan nilai null/kosong di kolom address. Setelah semua error yang ada di proses staging kami pecahkan. Proses staging berjalan dengan lancar, dan kami berhasil membuat staging dari OLTP DVD rental.

### 3.3 Desain Data Warehouse

Dalam menyusun data warehouse, kami memikirkan sekiranya tabel apa saja yang bisa digabungkan. Kami memperhatikan *primary key* dari setiap tabel dan menghubungkannya satu sama lain. Kami mengubah beberapa nama kolom dan juga menyatukan beberapa kolom. Berikut struktur dari tiap dimensi dan fakta data warehouse yang kami buat.

#### 1. dim\_store

- **sk\_store**
- **store\_id**
- manajer\_staff\_id → staff\_id join to table staff → staff\_id

*target column :*

first\_name + last\_name → staff\_name

- **address\_id**
- scd type 1

#### 2. dim\_customer

- **sk\_customer**
- **customer\_id**
- **store\_id**
- first\_name + last\_name → customer\_name

- address\_id → address\_id join to table address
 

*target column :*

  - address → customer\_address
  - district → customer\_district
  - city\_id → city\_id join to table city

*target column :*

  - city → customer\_city
  - country\_id → country\_id join to table country

*target column :*

  - country → customer\_country
  - postal\_code → customer\_postalcode
  - phone → customer\_phone
- active
  - 1 → Active
  - 0 → Not Active
- version
- date\_from
- date\_to
- scd type 2

### 3. dim\_film

- sk\_film
- film\_id
- title → film\_title
- description → film\_description
- release\_year → film\_release\_year
- language\_id → join to table language
 

*target column :*

  - name → film\_language
- rental\_duration → film\_rental\_duration
- rental\_rate → film\_rental\_rate
- length → film\_duration
- replacement\_cost → film\_replacement\_cost
- rating → film\_rating
 

*standarisasi rating :*

PG-13	: Parents Strongly
Cautioned G	: General Audience
R	: Restricted
NC-17	: No Children 17
PG	: Parental Guidance Suggested
- special\_features → film\_special\_features
- film\_id → join to table film\_category
 

*target column :*

  - category\_id → join to table category

- target column :*
- name → film\_category
  - film\_id → join to table film\_actor
- target column :*
- actor\_id → join to table actor
- target column :*
- first\_name + last\_name = actor\_name
- scd type 1

#### 4. dim\_date

- sk\_date
- date
- year
- quarter
- month
- week
- day
- scd type 1

#### 5. fact\_rental

- trx\_id
- sk\_customer
- sk\_store
- sk\_film
- sk\_date
- payment\_id
- customer\_id
- staff\_id
- rental\_id → rental\_id join to table rental

*target column :*

inventory\_id → inventory\_id join to table inventory

- target column :*
- film\_id
  - store\_i
  - d
  - rental\_date
  - return\_date
  - amount
  - payment\_date
  - scd type 1

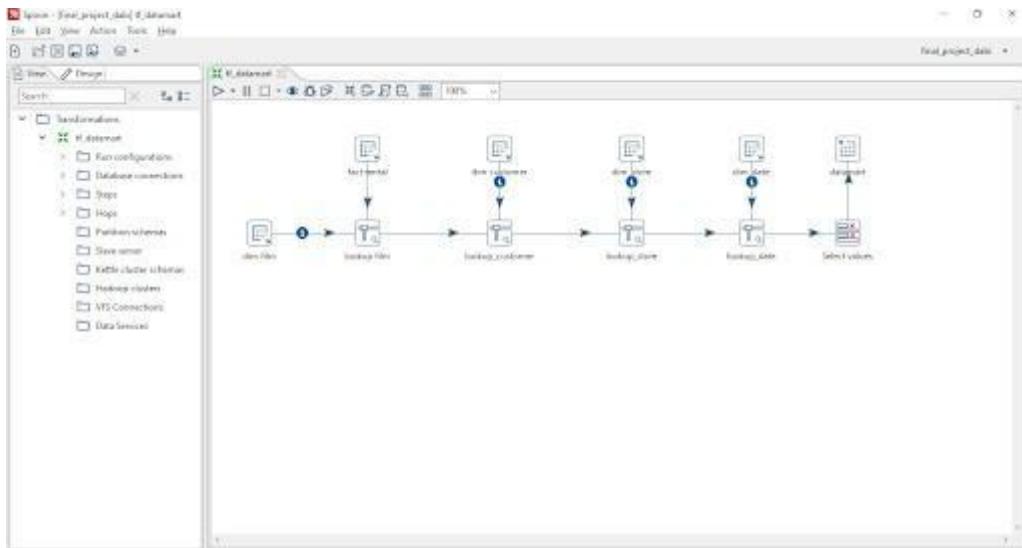
Desain data warehouse yang kami buat dirancang untuk mendukung analisis bisnis pada sistem rental dengan menggunakan model dimensional, yang terdiri dari tabel fakta dan tabel dimensi. Tabel dimensi menyimpan informasi deskriptif, seperti data pelanggan, toko,

film, dan tanggal, sedangkan tabel fakta mencatat data transaksi penyewaan. Tabel dim\_customer menyimpan informasi pelanggan seperti customer\_name, customer\_address, customer\_city, customer\_country, customer\_postalcode, customer\_phone, dan status pelanggan (active), yang menunjukkan apakah pelanggan aktif atau tidak. Riwayat perubahan data pelanggan dikelola menggunakan Slowly Changing Dimension (SCD) tipe 2, dengan menyimpan versi data historis dan periode waktu yang relevan.

Tabel dim\_store mencatat data toko, termasuk store\_id dan nama manajer toko (staff\_name), yang diperoleh dari tabel staff. Perubahan data pada tabel ini dikelola dengan SCD tipe 1, di mana data diperbarui langsung tanpa menyimpan riwayat. Tabel dim\_film berisi detail film seperti film\_title, film\_description, film\_release\_year, film\_language, film\_category, film\_duration, film\_rating, dan film\_special\_features. Tabel ini juga menyimpan informasi aktor melalui relasi dengan tabel actor, di mana actor\_name dihasilkan dari gabungan kolom first\_name dan last\_name. Standarisasi rating film dilakukan dengan format seperti PG-13 untuk Parents Strongly Cautioned dan R untuk Restricted. Sementara itu, tabel dim\_date menyimpan data kalender seperti year, quarter, month, week, dan day, untuk mendukung analisis berbasis waktu.

Tabel fakta, fact\_rental, mencatat semua transaksi penyewaan, seperti trx\_id, rental\_date, return\_date, amount, dan informasi terkait pelanggan, film, dan toko. Relasi antar tabel menggunakan foreign key, misalnya sk\_customer, sk\_store, sk\_film, dan sk\_date, yang terhubung ke tabel dim\_customer, dim\_store, dim\_film, dan dim\_date. Desain ini mendukung pertumbuhan data dan fleksibilitas analisis, memungkinkan analisis lintas dimensi, seperti pola penyewaan berdasarkan waktu, toko, atau kategori film.

### 3.4 Pembuatan Data Mart



Gambar 3.22 Transformasi data mart

Desain data mart di atas bertujuan untuk menganalisis transaksi penyewaan pada sistem rental dengan menggabungkan data dari beberapa tabel dimensi dan tabel fakta. Proses transformasi dan integrasi data dimulai dengan mengambil data dari tabel dimensi, seperti dim\_film, dim\_customer, dim\_store, dan dim\_date. Selanjutnya, proses lookup dilakukan pada setiap tabel dimensi untuk mencocokkan data menggunakan kunci, seperti sk\_film, sk\_customer, sk\_store, dan sk\_waktu , agar atribut tambahan dapat diambil dan dilengkapi.

Sebagai contoh, dari dim\_film, atribut seperti film\_title, film\_category, dan durasi film diambil; dari dim\_customer, atribut seperti customer\_name, alamat, dan status aktif pelanggan diperoleh; sedangkan dari dim\_store diambil data toko, seperti store\_id dan manajer toko. Untuk data tanggal, atribut seperti tahun, bulan, dan hari diperoleh dari tabel dim\_date. Setelah semua proses lookup selesai, kolom-kolom yang relevan dipilih melalui proses select values untuk membentuk datamart yang sudah terintegrasi.

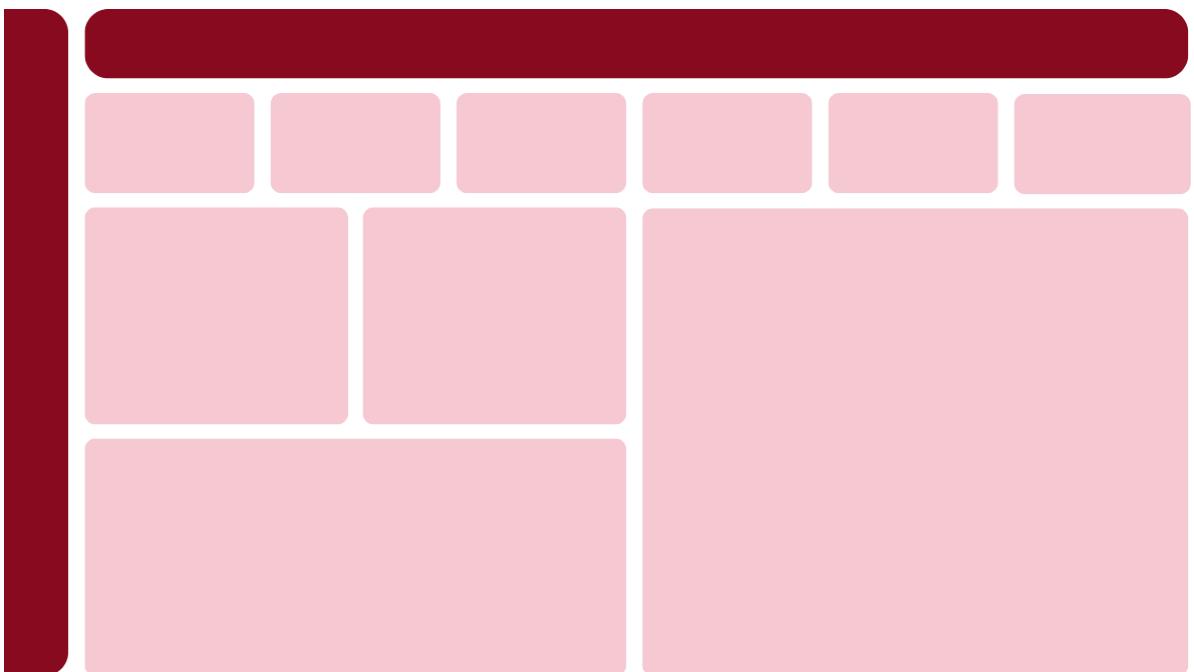
Output akhir dari proses ini adalah data yang terstruktur dan siap untuk dianalisis lebih lanjut. Data tersebut mencakup informasi transaksi penyewaan, seperti tanggal penyewaan, tanggal pengembalian, jumlah pembayaran, detail film yang disewa, data pelanggan, informasi toko, serta waktu transaksi. Dengan adanya datamart ini, berbagai analisis dapat dilakukan, seperti melihat tren penyewaan berdasarkan periode waktu, menganalisis performa toko, mempelajari pola penyewaan film berdasarkan kategori atau rating, serta memahami profil pelanggan berdasarkan aktivitas penyewaan.

### 3.5 Visualisasi dan Analisis Dashboard

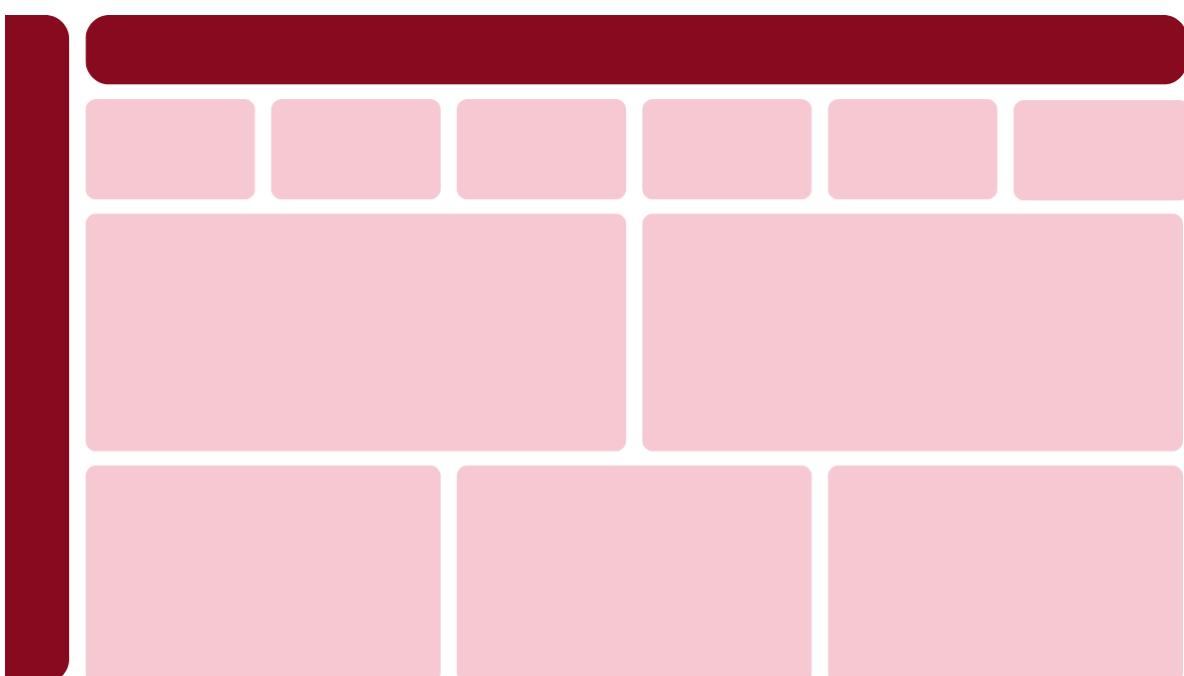


Gambar 3.23 Color Palette Visualisasi

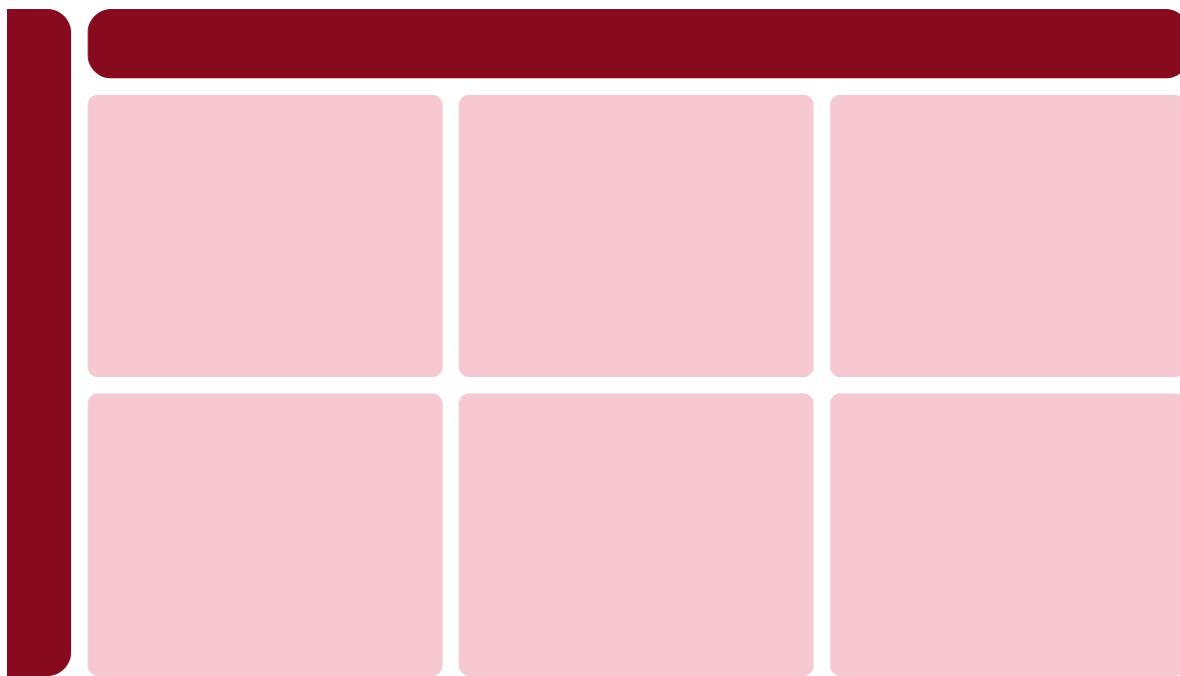
Sebelum membuat *dashboard*, kami menentukan dahulu *color palette* untuk background dan chart yang nantinya akan kami buat. Disini kami menggunakan warna-warna pastel dengan dominan warna pink. Pada mulanya kami menggunakan gradasi warna pink untuk *background*-nya, namun kami rubah agar tidak terjadi kesalahan persepsi saat *user/client* melihat *dashboard* tersebut. Karena permainan warna pada visualisasi cukup krusial. Ditakutkannya apabila warnanya dari gelap ke terang, mengartikan bahwa *value*-nya dari besar ke kecil.



Gambar 3.24 Desain Dashboard 1

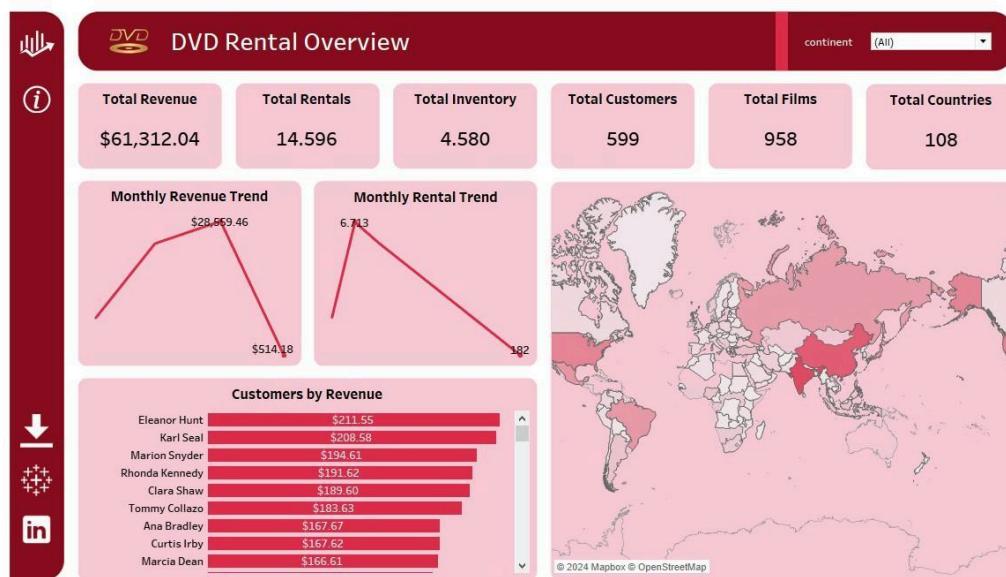


Gambar 3.25 Desain Dashboard 2



Gambar 3.26 Desain Dashboard 3

Setelah menentukan *color palette*, kami membuat template untuk *dashboard* menggunakan Canva. Kami membuat sekitar 21 *background chart* berwarna merah muda. Pada bagian atas kami khususkan untuk meletakkan judul, logo, dan filter. Lalu pada bagian kiri kami menempatkan background berwarna merah gelap untuk meletakkan button yang mengarahkan ke *dashboard* lain dan juga logo *download*, Tableau, dan Linkedin.

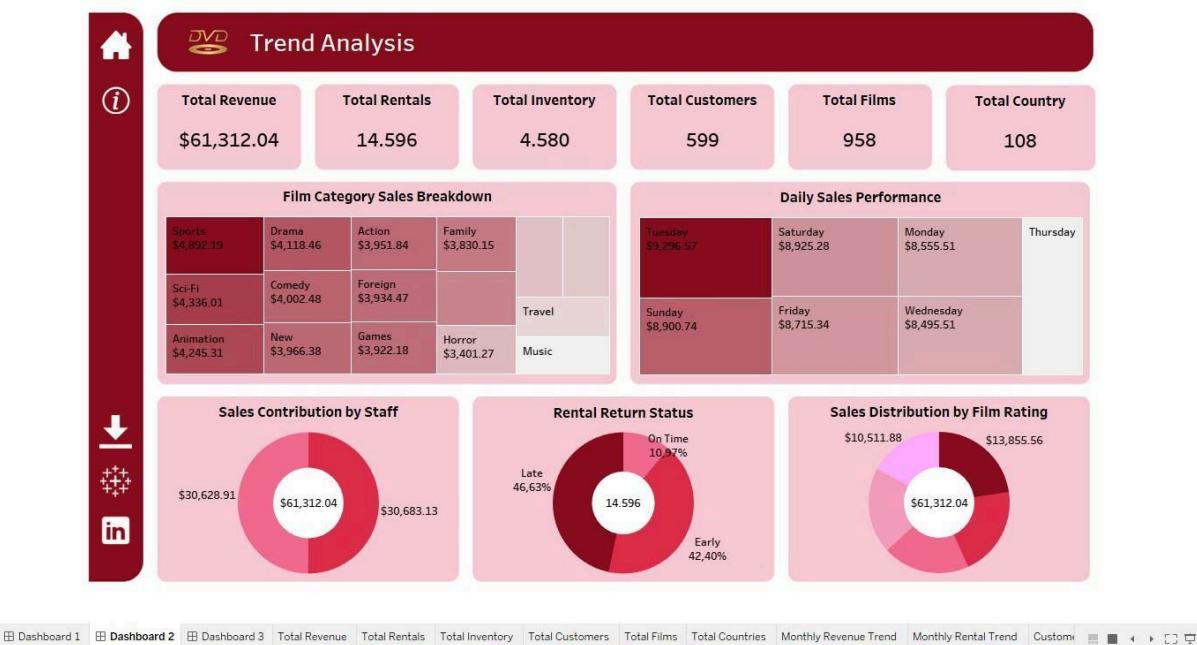


Gambar 3.27 Dashboard 1

Gambar di atas merupakan *dashboard* utama yang kami tampilkan. Judul dari dashboard utama kami adalah DVD Rental Overview. Kami menampilkan sebanyak 10 chart yang terdiri dari 6 *chart total*, 2 *line chart*, 1 *bar chart*, dan 1 terakhir berupa *map chart*. Kami menampilkan beberapa *chart total*, diantaranya total pendapatan yang diterima dari penyewaan DVD sebanyak 61 dolar, total penyewaan sebanyak 14,596, total persediaan sebanyak 4.580, total pelanggan yang sudah menyewa sebanyak 599, dan terakhir total negara penyewa sebanyak 108 negara.

Dibawahnya kami membuat line chart yang menjelaskan mengenai pendapatan dan penyewaan terbanyak terjadi di bulan apa. Diketahui pendapatan terbanyak terjadi pada bulan April 2007, sedangkan pendapatan terendah terjadi pada bulan May 2007. Untuk penyewaan terbanyak terjadi pada bulan Juli 2015 sebanyak 6.713 DVD tersewa, dan penyewaan terendah terjadi pada bulan Februari 2006 sebanyak 182 tersewa. Lalu pada bagian kanan terdapat *map chart* yang menunjukan persebaran dari negara yang sudah menyewa DVD. Negara-negara di benua Asia paling banyak menyewa DVD. *Chart* terakhir di dashboard 1 adalah *bar chart* yang menunjukan peringkat teratas dari *customer* berdasarkan uang yang sudah mereka keluarkan untuk menyewa DVD. Peringkat teratas dipegang oleh Eleanor Hunt sebanyak 211 dolar.

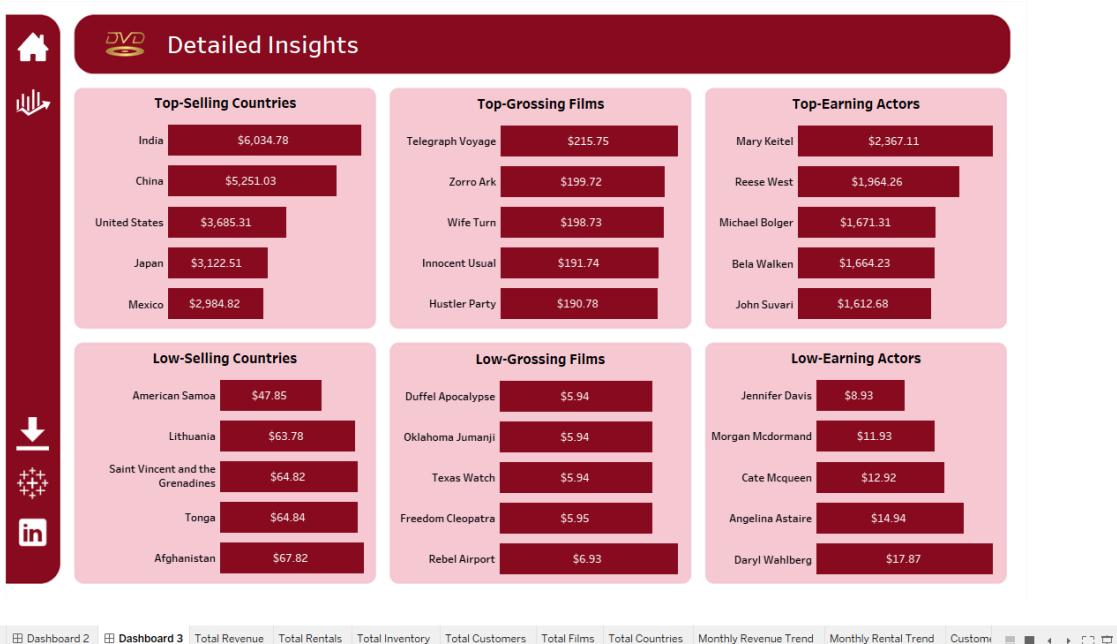
Tidak hanya *chart* yang terdapat pada *dashboard* 1, dapat terlihat juga kami menambahkan beberapa yang *button* dan parameter. Di bagian atas terdapat logo DVD yang diletakkan di sebelah judul. Lalu kami memberi judul *dashboard* yang pertama dengan “DVD Rental Overview” karena di *dashboard* pertama hanya menampilkan visualisasi umum. Di bagian kanan atas terdapat parameter *continent* yang dapat di *adjust* berdasarkan benua. Di bagian kiri dashboard terdapat *button* tanda panah untuk mendownload *dashboard*, *button* Tableau untuk mengarahkan ke profil tableau public, dan terakhir button Linkedin untuk mengarahkan ke profil Linkedin.



Gambar 3.28 Dashboard 2

Jika kita menekan salah satu negara pada bagian map, maka tampilan *dashboard* langsung beralih ke *dashboard* kedua. Pada *dashboard* kedua ada beberapa *chart* yang sama dengan *dashboard* pertama. Terdapat 5 *chart* yang baru pada *dashboard* kedua, 2 *area chart* dan 3 donat *chart*. Area *chart* yang pertama menjelaskan mengenai kategori film yang paling banyak disewa. Kategori film olahraga adalah yang paling banyak terjual sebesar 4,892 dolar dan musik yang paling sedikit terjual sebesar 3,071 dolar. Area *chart* yang kedua menampilkan hari yang paling banyak terjadi penyewaan DVD diukur berdasarkan pendapatan yang diperoleh. Hari Kamis menjadi hari terbanyak terjadi penyewaan DVD sebesar 9 dolar, sedangkan hari selasa paling sedikit hanya sebesar 8 dolar.

Pada bagian bawah *dashboard* terdapat 3 donat *chart*, donat *chart* yang pertama menampilkan kontribusi staff diukur dari segi hasil penjualan. Dari kedua staf tidak memiliki perbedaan yang signifikan. Jon Stephens memiliki penjualan yang lebih tinggi dibanding Mike Hillyer sebesar 30,683 dolar. Donat *chart* yang kedua memvisualisasikan mengenai ketepatan pengembalian DVD. Dari data yang dihasilkan, 46% *customer* terlambat mengembalikan DVD dan hanya 10% yang mengembalikan tepat waktu. Donat *chart* terakhir mengenai rating film yang paling banyak disewa. Film dengan rating *parents strongly cautioned* paling banyak disewa dengan pendapatan sebesar 13 dolar. Sedangkan film dengan rating *General Audience* paling sedikit disewa hanya sebesar 10 dolar.



Gambar 3.29 Dashboard 3

Jika dilihat pada bagian kiri dashboard 2 terdapat button pada bagian kiri yang jika ditekan akan mengerahkan ke dashboard 3. Di dashboard 3 kami menampilkan detail dari insight DVD rental. terdapat 6 bar chart yang saling berpasangan. Bar chart pertama adalah negara yang paling banyak menyewa DVD yaitu India dan bar chart di bawahnya mengenai negara dengan pendapatan sewa paling sedikit yaitu American Samoa. Bar chart yang ditengah menunjukkan Film yang paling banyak disewa dan diukur berdasarkan total penjualan. Film yang paling banyak disewa adalah Telegraph Voyage sebesar 215 dolar, dan film yang paling sedikit disewa adalah Duffel Apocalypse yang hanya sebesar 5 dolar.

Pasangan bar chart yang terakhir adalah aktor pemain film yang paling laku disewa dan aktor dengan pendapatan sewa paling sedikit. Aktor dengan pendapatan sewa paling banyak adalah Mary Keitel dengan pendapatan sebesar 2,367.11 dolar dan Jennifer Davis adalah aktor dengan pendapatan sewa paling sedikit hanya sebesar 8.93 dolar. Di bagian Kiri atas dashboard terdapat 2 button, button berbentuk rumah akan mengarahkan ke dashboard pertama dan button berbentuk chart akan mengarahkan kembali ke dashboard 2.

Dari tiga gambar yang ditampilkan, dashboard ini dirancang untuk memberikan gambaran menyeluruh tentang bisnis penyewaan DVD. Bagian pertama menyajikan ikhtisar kinerja bisnis, seperti total pendapatan, total penyewaan, total inventaris, total pelanggan, total film, dan total negara yang dilayani. Informasi ini memberikan gambaran umum tentang skala dan volume bisnis penyewaan DVD. Bagian selanjutnya adalah analisis tren, yang

menyajikan tren penjualan berdasarkan kategori film, seperti olahraga, drama, action, dan lain-lain. Selain itu, juga ada tren penjualan harian, yang memungkinkan pemantauan pola penjualan dari hari ke hari. Informasi ini dapat membantu mengidentifikasi preferensi pelanggan dan tren pasar yang sedang berlangsung. Bagian terakhir menyajikan wawasan rinci, seperti negara-negara terlaris, film-film box office, dan pendapatan aktor-aktor top. Informasi ini memberikan pemahaman yang lebih komprehensif tentang dinamika pasar dan faktor-faktor yang mempengaruhi kinerja bisnis.

## **BAB 4**

### **PENUTUP**

#### **4.1 Kesimpulan**

Dari analisis yang telah kami lakukan telah memberikan manfaat signifikan dalam mendukung kinerja bisnis persewaan DVD. Dengan memanfaatkan teknologi seperti PostgreSQL, Pentaho, Dbeaver dan Tableau, kami berhasil membangun infrastruktur data yang mencakup staging area, data warehouse, data mart, dan dashboard interaktif. Melalui analisis dan visualisasi yang kamu lakukan berhasil mengidentifikasi pola pendapatan, preferensi pelanggan, dan kontribusi berbagai variabel terhadap pendapatan bisnis. Dari analisis ini juga memberikan solusi untuk mengurangi fluktuasi pendapatan, meningkatkan efisiensi layanan pelanggan, dan mengoptimalkan stok film. Selain itu, *dashboard* dirancang untuk memudahkan para pemangku kepentingan untuk memantau kinerja bisnis secara *real-time* dan membuat keputusan strategis. Meskipun kami menghadapi beberapa kendala teknis selama implementasi, analisis ini berhasil menunjukkan bahwa integrasi data dan analisis berbasis teknologi dapat mendukung stabilisasi dan peningkatan pendapatan pada bisnis rental DVD.

#### **4.2 Saran**

Untuk pengembangan lebih lanjut, kami merekomendasikan beberapa saran strategis. Dari sisi teknis, infrastruktur data dapat diperluas dengan integrasi sumber data tambahan, seperti data pemasaran atau media sosial, untuk mendapatkan wawasan yang lebih komprehensif terkait preferensi pelanggan. Selain itu, pembangunan fitur otomatisasi pada proses ETL (Extract, Transform, Load) akan meningkatkan efisiensi dan pemrosesan data dan mengurangi potensi kesalahan. Dari sisi analisis, penerapan model prediktif berbasis machine learning dapat digunakan untuk memproyeksikan tren pendapatan, perilaku pelanggan, atau kategori film yang akan populer di masa mendatang. Terakhir, dari sisi bisnis, strategi pemasaran berbasis data seperti personalisasi promosi atau program loyalitas pelanggan dapat diterapkan untuk meningkatkan retensi pelanggan dan mendorong penjualan.

## **DAFTAR PUSTAKA**

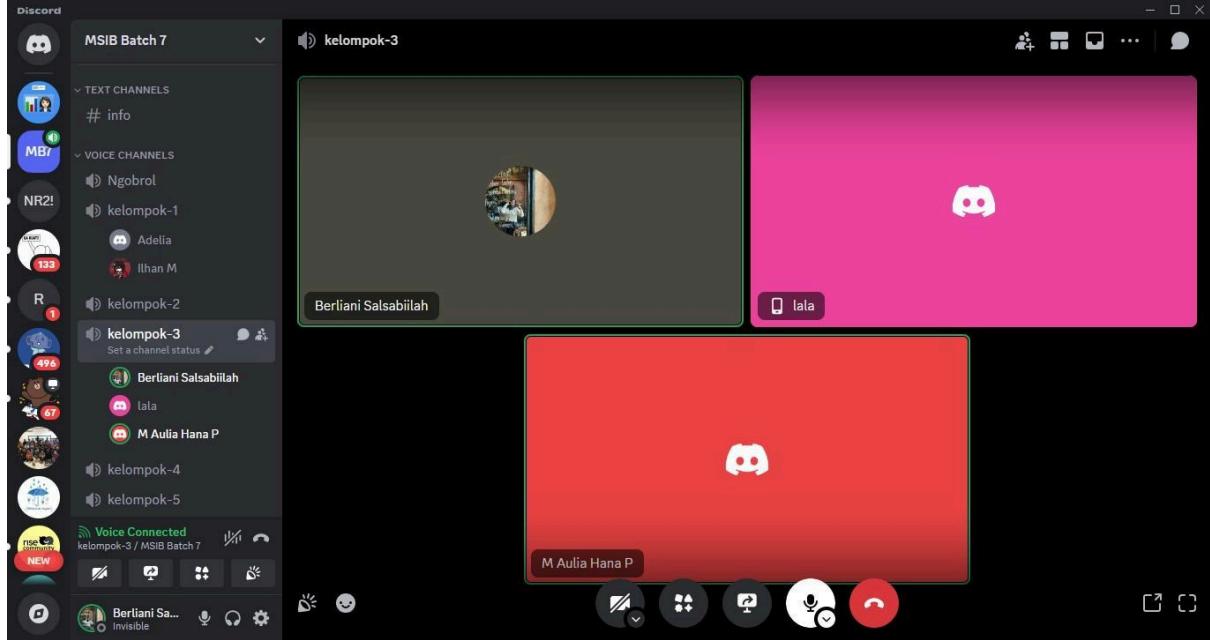
Hasan, F. N., & Sudaryana, I. K. (2022). Penerapan Business Intelligence & Online Analytical Processing untuk Data-Data Penelitian dan Luarannya pada Perguruan Tinggi Menggunakan Pentaho. Infotech: Journal of Technology Information, 8(2), 85-92.

Munawaroh, S. (2005). Mengeksplorasi Database PostgreSQL dengan PgAdmin III. Dinamik, 10(2).

Puteri, S. R. (2022). Analisis Visualisasi Data Kecamatan Kertapati menggunakan Tableau Public. *JUPITER: Jurnal Penelitian Ilmu dan Teknologi Komputer*, 14(2-b), 366-373.

## LAMPIRAN

### Lampiran 1



Kami rutin mengadakan diskusi di Discord setiap jam 20.00 malam dari senin - jumat. Tak jarang juga pada hari sabtu dan minggu kami tetap mengadakan discord agar bisa memanfaatkan waktu yang ada semaksimal mungkin. Saat kami dan kelompok-kelompok lain mengadakan Discord, Kak Ryan Firmansyah selaku mentor memberi arahan-arahan dan evaluasi yang harus kami lakukan selama *final project* berlangsung.