

***FINAL PROJECT***

**STUDI KASUS ANALISIS POPULARITAS FILM BERDASARKAN FREKUENSI  
PENYEWAAN DVD RENTAL**



**Mentor:**  
Lisna Sulistiawati

**Kelompok 5:**

Muhammad Rakha Naufal	22082010060
Adelia Putri Pratiwi	22082010034
Yessi Mardina Manik	210803015
Ayunda Meilin Stevany	22051214055

**Kampus Merdeka Batch 7**

**Data Analyst & Business Intelligence**

**Celerates School**

**2024**

## KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, kami dapat menyelesaikan tugas akhir ini dengan baik. Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan program Studi Independen Kampus Merdeka sebagai *Data Analyst & Business Intelligence* di Celerates (PT. Mitra Talenta Grup).

Kami ingin menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam proses penyelesaian *final project* ini. Terima kasih kepada trainer, Kak Putra Gema Nusa, atas bimbingan, arahan serta saran yang telah diberikan selama proses penggerjaan laporan ini. Kami juga berterima kasih kepada mentor, Kak Lisna Sulistiawati, yang telah memberikan dukungan dan motivasi.

Kami menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, kami sangat terbuka untuk menerima kritik dan saran yang membangun demi perbaikan. Kami berharap laporan *final project* ini dapat memberikan kontribusi yang bermanfaat, tidak hanya sebagai pembelajaran bagi kami, tetapi juga memberikan wawasan lebih dalam tentang penerapan analisis data. Akhir kata, semoga laporan *final project* ini dapat memenuhi ekspektasi dan dapat diapresiasi oleh semua pihak yang berkepentingan.

**Kelompok 5 DABI**

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>DAFTAR ISI.....</b>	<b>3</b>
<b>DAFTAR GAMBAR.....</b>	<b>4</b>
<b>ABSTRAK.....</b>	<b>7</b>
<b>BAB I.....</b>	<b>8</b>
1.1 Latar Belakang.....	8
1.2 Tujuan Project.....	8
1.3 Manfaat Project.....	9
1.3.1 Manfaat Teknis.....	9
1.3.2 Manfaat Bisnis.....	10
1.4 Lingkup Project.....	11
<b>BAB II.....</b>	<b>12</b>
<b>METODOLOGI PROJECT.....</b>	<b>12</b>
2.1 Deskripsi Database.....	12
2.2 Tahapan Project.....	18
2.3 Tools dan Teknologi.....	22
<b>BAB III.....</b>	<b>26</b>
<b>IMPLEMENTASI &amp; HASIL.....</b>	<b>26</b>
3.1 Input Database.....	26
3.2 Pembuatan Staging Area.....	27
3.3 Desain Data Warehouse.....	30
3.4 Pembuatan Data Mart.....	42
3.5 Visualisasi dan Analisis Dashboard.....	43
<b>BAB IV.....</b>	<b>52</b>
<b>PENUTUP.....</b>	<b>52</b>
4.1 Kesimpulan.....	52
4.2 Saran.....	52
<b>DAFTAR PUSTAKA.....</b>	<b>54</b>
<b>LAMPIRAN.....</b>	<b>55</b>
1. Proses Staging Area.....	55
2. Pembuatan Data Warehouse.....	64
3. Pembuatan Data Mart.....	83
4. Pembuatan Visualisasi Data melalui Tableau sudah didokumentasikan pada Bab 3.....	86

## DAFTAR GAMBAR

Gambar 2.2.1. Tabel City.....	11
Gambar 2.2.2. Tabel Country.....	12
Gambar 2.2.3. Tabel Customer.....	12
Gambar 2.2.4. Tabel Film.....	13
Gambar 2.2.5. Tabel Film Category.....	13
Gambar 2.2.6. Tabel Inventory.....	14
Gambar 2.2.7. Tabel Language.....	15
Gambar 2.2.8. Tabel Rental.....	15
Gambar 3.1.1 Hasil restore database.....	25
Gambar 3.2.1. Membuat schema log.....	26
Gambar 3.2.2. Menjalankan Transformasi tf_list_tabel.....	27
Gambar 3.2.3. Menjalankan main.job.....	27
Gambar 3.2.4. Membuka Dbeaver dan Memeriksa Tabel log_table_masuk.....	28
Gambar 3.2.5. Pengecekan pada Tabel staging_log.....	28
Gambar 3.2.6. Memeriksa Salah Satu Tabel di Staging Area.....	29
Gambar 3.3.1. ETL Customer.....	30
Gambar 3.3.2 Running Pentaho.....	30
Gambar 3.3.3 ETL Dimensi Film.....	31
Gambar 3.3.4 Running Pentaho.....	31
Gambar 3.3.5 ETL Dimensi Inventory.....	32
Gambar 3.3.6. Running Pentaho.....	33
Gambar 3.3.7. ETL Dimensi Payment.....	34
Gambar 3.3.8. Running Pentaho.....	34
Gambar 3.3.9. ETL Dimensi Rental Date.....	35
Gambar 3.3.10. Running Pentaho.....	35
Gambar 3.3.11. ETL Dimensi Staff.....	36
Gambar 3.3.12. Running Pentaho.....	37
Gambar 3.3.13 ETL Dimensi Store.....	38
Gambar 3.3.14. Running Pentaho.....	38
Gambar 3.3.15. ETL Fact Tabel.....	39
Gambar 3.3.16. Running Pentaho.....	40
Gambar 3.4.1. ETL Datamart.....	41
Gambar 3.4.2. Running Pentaho.....	41
Gambar 3.5.1.1. Tampilan awal Tableau.....	43
Gambar 3.5.2.1. Visualisasi Total Rental.....	43
Gambar 3.5.2.2. Visualisasi Total Movies Rented.....	44
Gambar 3.5.2.3. Visualisasi Most Popular Film.....	44
Gambar 3.5.2.4. Visualisasi Highest Grossing Film.....	45
Gambar 3.5.2.5. Visualisasi Top 10 Most Popular Movies.....	45
Gambar 3.5.2.6. Visualisasi Movie Rental Trends.....	46

Gambar 3.5.2.7. Visualisasi Rental Distribution by Store.....	46
Gambar 3.5.2.8. Visualisasi Movie Rental Distribution by Country.....	47
Gambar 3.5.3.1. Tampilan dashboard 1.....	48
Gambar 3.5.3.1. Tampilan dashboard 2.....	48
Gambar 1.1. Restore database.....	52
Gambar 1.2. Copy tabel di database.....	53
Gambar 1.3. Pembuatan Struktur Tabel.....	53
Gambar 1.4. Rename nama schema.....	54
Gambar 1.5. Menghapus View yang Tidak Diperlukan.....	54
Gambar 1.6. Membuat Schema Baru untuk Log.....	54
Gambar 1.7. Hapus foreign key Setiap Tabel Schema.....	55
Gambar 1.8. Hapus foreign key Setiap Tabel Schema.....	56
Gambar 1.9. Hasil hapus foreign key Setiap Tabel Schema.....	56
Gambar 1.10. Ubah Properti Kolom Menjadi Nullable di Tabel Address.....	56
Gambar 1.11. Hasil Mengubah Kolom di Tabel Address.....	57
Gambar 1.12. Hapus Indeks film_fulltext.idx pada Tabel Staging.....	58
Gambar 1.13. Mengubah Data Type pada Tabel Film.....	58
Gambar 1.14. Membuat Repository Baru Di Pentaho.....	58
Gambar 1.15. Buka File Berisi File Transformasi.....	59
Gambar 1.16. Memastikan Direktori File Transformasi.....	59
Gambar 1.17. Membuat Koneksi Baru ke Database dvdrental.....	60
Gambar 1.18. Membuat Koneksi Baru di Database dvdrental_staging.....	60
Gambar 2.1 Membuat Database dw_rental.....	61
Gambar 2.2. Menghubungkan Target Schema.....	62
Gambar 2.3 Isi dari Tabel Input Customer.....	62
Gambar 2.4 Preview Data.....	62
Gambar 2.5 Penggabungan Kolom.....	63
Gambar 2.6 Penggunaan Select Values.....	64
Gambar 2.7. Isi dari Dimension Lookup.....	64
Gambar 2.8. Isi dari Dimension Lookup.....	64
Gambar 2.9. Isi dari Tabel Input Film.....	65
Gambar 2.10. Preview Data.....	65
Gambar 2.11. Pembersihan Simbol pada Data.....	66
Gambar 2.11. Rename Nama Tabel.....	66
Gambar 2.12. Menambahkan SK baru.....	67
Gambar 2.13. Isi dari Insert/Update.....	67
Gambar 2.14. Isi dari Tabel Input Inventory.....	67
Gambar 2.15. Preview Data.....	68
Gambar 2.16. Rename Nama Kolom.....	68
Gambar 2.17. Membuat SK Baru.....	69
Gambar 2.18. Isi dari Insert/Update.....	69

Gambar 2.19. Tabel Input Payment.....	70
Gambar 2.20. Preview Data.....	70
Gambar 2.21. Membuat SK Baru.....	71
Gambar 2.22. Isi dari Insert/Update.....	71
Gambar 2.23. Tabel Input Dimensi Staff.....	71
Gambar 2.24. Preview Data.....	72
Gambar 2.25. Penggabungan Kolom.....	72
Gambar 2.26. Rename Kolom.....	73
Gambar 2.27. Isi dari Dimension Lookup.....	73
Gambar 2.28. Isi dari Dimension Lookup.....	73
Gambar 2.29. Tabel Input Dimensi Store.....	74
Gambar 2.30. Preview Data.....	74
Gambar 2.31. Penggabungan Kolom.....	75
Gambar 2.32. Rename Kolom.....	75
Gambar 2.23. Isi dari Dimension Lookup.....	76
Gambar 2.24. Isi dari Dimension Lookup.....	76
Gambar 2.25. Tabel Input Fact Table.....	77
Gambar 2.26. Preview Data.....	77
Gambar 2.28. Database Lookup Fact Table.....	78
Gambar 2.29. Memilih Kolom Relevan.....	79
Gambar 2.30. Membuat SK baru Untuk Fact Table.....	79
Gambar 2.31. Eksekusi Tabel Output.....	80
Gambar 3.1. Buat Koneksi Baru.....	80
Gambar 3.2. Tabel Input Datamart.....	81
Gambar 3.3. Preview Data.....	81
Gambar 3.4. Stream Lookup Dimensi.....	81
Gambar 3.5. Tabel Input Inventory.....	82
Gambar 3.6. Memilih Kolom Relevan.....	82
Gambar 3.7. Eksekusi Tabel Output Datamart.....	83

## ABSTRAK

Industri hiburan, khususnya bisnis penyewaan DVD, menghadapi tantangan besar seiring perkembangan teknologi dan munculnya layanan streaming digital. Permasalahan utama yang dihadapi adalah memahami tingkat popularitas film untuk mengelola inventaris dan merancang strategi pemasaran yang lebih efektif. Penelitian ini bertujuan untuk menganalisis popularitas film berdasarkan frekuensi penyewaan, dengan fokus pada identifikasi film populer, analisis preferensi kategori, pola penyewaan musiman, dan optimalisasi stok. Metode yang digunakan meliputi pengolahan data historis melalui proses ETL, desain data warehouse, dan pembuatan dashboard interaktif untuk visualisasi hasil analisis. Data yang digunakan berasal dari DVD Rental Database yang mencakup informasi tentang film, pelanggan, kategori, dan penyewaan. Hasil analisis diharapkan memberikan wawasan untuk pengelolaan inventaris yang lebih efisien, pengembangan strategi pemasaran yang lebih tepat sasaran, serta meningkatkan pengalaman pelanggan dan efisiensi operasional bisnis penyewaan DVD di era digital.

**Kata Kunci:** Penyewaan DVD, Popularitas film, Analisis data.

## *ABSTRACT*

*The entertainment industry, especially the DVD rental business, is facing major challenges as technology advances and the emergence of digital streaming services. The main problem faced is understanding the popularity level of films in order to manage inventory and design more effective marketing strategies. This research aims to analyze film popularity based on rental frequency, with a focus on identifying popular films, analyzing category preferences, seasonal rental patterns, and stock optimization. The methods used include processing historical data through the ETL process, data warehouse design, and creating interactive dashboards for visualization of analysis results. The data used comes from the DVD Rental Database which includes film, customer, category and rental information. The results of the analysis are expected to provide insight for more efficient inventory management, development of more targeted marketing strategies, as well as improving customer experience and operational efficiency of DVD rental businesses in the digital era.*

**Keywords:** *DVD rental, film popularity, data analysis.*

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Industri hiburan memiliki peran penting dalam kehidupan masyarakat, salah satunya melalui bisnis penyewaan film dalam format *DVD*. Pada masa kejayaannya, penyewaan *DVD* menjadi pilihan utama bagi masyarakat untuk menikmati berbagai jenis film secara fleksibel di rumah. Namun, seiring dengan perkembangan teknologi, khususnya kehadiran layanan *streaming digital*, bisnis penyewaan *DVD* mulai menghadapi tantangan besar. Persaingan yang semakin ketat ini mendorong pelaku usaha untuk mencari strategi berbasis data guna memahami preferensi pelanggan dan mengoptimalkan layanan yang diberikan.

Salah satu permasalahan utama yang sering dihadapi dalam bisnis penyewaan *DVD* adalah memahami tingkat popularitas film. Informasi ini diperlukan untuk mengelola inventaris dengan lebih efektif, menentukan jenis film yang perlu diperbanyak, serta merancang strategi pemasaran yang lebih tepat sasaran. Analisis terhadap frekuensi penyewaan dapat menjadi pendekatan untuk mengetahui preferensi pelanggan secara lebih spesifik, seperti film yang paling diminati, kategori yang sering dipilih, hingga pola musiman yang mempengaruhi permintaan. Dengan informasi ini, pengelolaan bisnis dapat dilakukan secara lebih terukur dan relevan dengan kebutuhan pasar.

Melalui analisis yang dilakukan, penelitian ini bertujuan untuk memberikan wawasan yang dapat digunakan sebagai dasar dalam pengambilan keputusan bisnis, seperti optimalisasi stok film, pengembangan program loyalitas pelanggan, atau pelaksanaan promosi yang lebih efektif. Dengan memanfaatkan data historis secara optimal, penelitian ini diharapkan dapat membantu meningkatkan efisiensi operasional dan daya saing bisnis penyewaan *DVD* di era yang penuh tantangan ini.

#### **1.2 Tujuan Project**

Penelitian ini bertujuan untuk menganalisis popularitas film berdasarkan frekuensi penyewaan guna memberikan wawasan yang dapat mendukung pengambilan keputusan strategis dalam pengelolaan bisnis penyewaan *DVD*. Adapun tujuan yang ingin dicapai dalam penelitian ini meliputi:

## **1. Mengidentifikasi Film yang Paling Populer**

Menentukan daftar film dengan frekuensi penyewaan tertinggi yang dapat menjadi indikator minat pelanggan terhadap konten tertentu.

## **2. Menganalisis Preferensi Kategori Film**

Memahami kategori film yang paling diminati pelanggan, seperti aksi, komedi, atau drama, sehingga dapat membantu perusahaan dalam menyusun strategi penyediaan inventaris.

## **3. Menganalisis Pola Penyewaan Berdasarkan Waktu**

Mengidentifikasi tren penyewaan, termasuk pola musiman atau periode tertentu, untuk mendukung perencanaan operasional yang lebih efektif.

## **4. Memberikan Rekomendasi untuk Optimalisasi Inventaris**

Memberikan rekomendasi berbasis data untuk pengelolaan stok film yang lebih tepat guna mengurangi resiko kelebihan atau kekurangan stok.

## **5. Mendukung Strategi Promosi yang Efektif**

Menyediakan dasar untuk merancang promosi atau penawaran khusus berdasarkan kategori atau film tertentu yang dapat mendorong peningkatan jumlah penyewaan.

Penelitian ini diharapkan mampu memberikan kontribusi yang signifikan dalam mendukung efisiensi operasional dan peningkatan layanan bagi pelanggan, sehingga bisnis penyewaan *DVD* dapat bersaing di tengah tantangan pasar yang terus berubah.

### **1.3 Manfaat Project**

Penelitian ini diharapkan memberikan manfaat yang signifikan baik dari sisi teknis maupun bisnis. Dari sisi teknis, penelitian ini mendukung pengembangan keterampilan dalam pengolahan data secara terstruktur, mulai dari proses ekstraksi hingga penyajian dalam bentuk visualisasi yang informatif. Sedangkan dari sisi bisnis, analisis data historis diharapkan dapat menghasilkan wawasan strategis untuk mendukung efisiensi operasional, pengelolaan inventaris, dan peningkatan kualitas layanan. Berikut adalah manfaat yang diharapkan dari penelitian ini:

#### **1.3.1 Manfaat Teknis**

##### **1. Peningkatan Kemampuan Analisis Data**

Proyek ini memberikan kesempatan untuk mempelajari dan menerapkan proses

pengolahan data, mulai dari *ETL* (*Extract, Transform, Load*), desain *data warehouse*, hingga pembuatan *dashboard* interaktif yang mendukung analisis lebih mendalam.

## 2. Penguasaan Teknologi Pendukung

Mengembangkan keterampilan teknis dalam penggunaan alat-alat seperti *PostgreSQL* untuk manajemen *database*, *Pentaho* untuk proses *ETL*, serta *Tableau* untuk visualisasi data.

## 3. Desain Infrastruktur Data yang Efisien

Memberikan pemahaman tentang bagaimana membangun *data warehouse* dan *data mart* yang mendukung analisis data secara terstruktur dan terorganisasi dengan baik.

## 4. Kemampuan Analisis Mendalam

Membantu peserta memahami metode untuk menganalisis data historis penyewaan film, mengidentifikasi pola, serta menghasilkan wawasan yang relevan berdasarkan data tersebut.

### 1.3.2 Manfaat Bisnis

#### 1. Pengelolaan Inventaris yang Lebih Tepat

Informasi tentang popularitas film dan kategori yang paling diminati dapat membantu perusahaan dalam menentukan stok inventaris yang sesuai dengan permintaan pelanggan, sehingga mengurangi risiko kelebihan atau kekurangan stok.

#### 2. Pengembangan Strategi Pemasaran yang Efektif

Dengan memahami preferensi pelanggan dan tren penyewaan, perusahaan dapat merancang promosi dan strategi pemasaran yang lebih relevan, seperti menawarkan diskon pada film tertentu atau menonjolkan kategori yang sedang populer.

#### 3. Meningkatkan Pengalaman Pelanggan

Pengetahuan tentang kebutuhan dan minat pelanggan memungkinkan perusahaan untuk menyediakan layanan yang lebih terfokus dan memuaskan, yang pada akhirnya dapat meningkatkan loyalitas pelanggan.

#### 4. Meningkatkan Efisiensi Operasional

Analisis data memungkinkan perusahaan untuk mengoptimalkan operasional, baik dalam pengelolaan inventaris maupun perencanaan bisnis berbasis data.

Melalui manfaat-manfaat tersebut, penelitian ini memberikan dampak signifikan, tidak hanya dalam peningkatan kemampuan teknis dan analisis data, tetapi juga dalam

mendukung pencapaian tujuan strategis yang relevan untuk keberhasilan bisnis penyewaan *DVD* secara keseluruhan.

#### **1.4 Lingkup Project**

Penelitian ini memiliki ruang lingkup yang terfokus untuk memastikan keberhasilan setiap tahap proses analisis data. Lingkup proyek mencakup kegiatan berikut:

- 1. Pembuatan Staging Area**

Data dari *DVD Rental Database* diimpor ke dalam *PostgreSQL* dan disiapkan dalam *staging area*. Proses ini mencakup pembersihan data, normalisasi, serta transformasi awal agar data dapat digunakan pada tahap selanjutnya.

- 2. Perancangan Data Warehouse**

Data yang telah disiapkan akan dimuat ke dalam *data warehouse* yang dirancang dengan struktur *star schema*. Tabel dimensi, seperti dimensi waktu, film, kategori, dan pelanggan, serta tabel fakta yang memuat frekuensi penyewaan akan dibuat untuk mendukung analisis data.

- 3. Pengembangan Data Mart**

Dari *data warehouse* yang telah dirancang, *data mart* dibuat untuk fokus pada analisis tertentu, yaitu popularitas film berdasarkan frekuensi penyewaan. *Data mart* ini dirancang untuk memberikan data yang terfokus dan relevan untuk kebutuhan analisis yang lebih spesifik.

- 4. Pembuatan Dashboard Interaktif**

Data dari *data mart* akan divisualisasikan dalam bentuk *dashboard* menggunakan *Tableau*. *Dashboard* ini dirancang untuk menyajikan informasi seperti daftar film terpopuler, kategori dengan penyewaan tertinggi, dan tren penyewaan berdasarkan waktu, sehingga memudahkan pengguna dalam memahami hasil analisis.

Proyek ini tidak mencakup aspek lain seperti pengelolaan data *real-time*, prediksi berbasis *machine learning*, atau integrasi dengan sistem eksternal. Fokus utama adalah pada pengolahan data historis untuk menghasilkan wawasan yang relevan bagi pengelolaan dan pengambilan keputusan strategis dalam bisnis penyewaan *DVD*.

## **BAB II**

# **METODOLOGI PROJECT**

## 2.1 Deskripsi Database

Penelitian ini menggunakan *DVD Rental Database* sebagai sumber data utama. Database ini mencakup data terkait film, kategori, pelanggan, penyewaan, inventaris, lokasi serta bahasa. Struktur database terdiri dari beberapa tabel utama yaitu:

### 1. Tabel *city*

Tabel ini menyimpan informasi terkait dengan lokasi geografis, yaitu kota tempat toko atau pelanggan berada. Ini bisa digunakan untuk menganalisis data berdasarkan lokasi, misalnya, untuk melihat apakah ada perbedaan dalam popularitas film antar kota.

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
<a href="#">123</a> city_id	1	serial4			[v]	nextval('city_city_id_seq'::regclass)	
<a href="#">A-Z</a> city	2	varchar(50)		<u>default</u>	[v]		
<a href="#">123</a> country_id	3	int2			[v]		
<a href="#">∅</a> last_update	4	timestamp			[v]	now()	

### **Gambar 2.2.1. Tabel City**

## 2. Tabel *country*

Tabel ini menyimpan informasi tentang negara tempat pelanggan atau toko berada. Tabel ini dapat digunakan untuk analisis yang lebih luas, seperti mengetahui tren popularitas film di berbagai negara.

Column Name	#	Data type	Identity	Collation	Not Null	Default	
<code>country_id</code>	1	serial4			[v]	nextval('country_country_id_seq'::regclass)	
<code>country</code>	2	varchar(50)		<u>default</u>	[v]		
<code>last_update</code>	3	timestamp			[v]	now()	

**Gambar 2.2.2. Tabel Country**

### 3. Tabel *customer*

Tabel ini menyimpan informasi tentang pelanggan yang menyewa film. Tabel ini berguna untuk analisis demografis dan untuk melacak frekuensi penyewaan berdasarkan pelanggan.

Column Name	#	Data type	Identity	Collation	Not Null	Default	
<code>customer_id</code>	1	serial4			[v]	nextval('customer_customer_id_seq'::regclass)	
<code>store_id</code>	2	int2			[v]		
<code>first_name</code>	3	varchar(45)		<u>default</u>	[v]		
<code>last_name</code>	4	varchar(45)		<u>default</u>	[v]		
<code>email</code>	5	varchar(50)		<u>default</u>	[ ]		
<code>address_id</code>	6	int2			[v]		
<input checked="" type="checkbox"/> <code>activebool</code>	7	bool			[v]	true	
<code>create_date</code>	8	date			[v]	'now'::text::date	
<code>last_update</code>	9	timestamp			[ ]	now()	
<code>active</code>	10	int4			[ ]		

**Gambar 2.2.3. Tabel Customer**

#### 4. Tabel film

Tabel ini menyimpan informasi dasar tentang setiap film, seperti judul, deskripsi, tahun rilis, *rating* dan bahasa film. Tabel ini digunakan untuk analisis popularitas film berdasarkan berbagai kriteria, seperti genre, bahasa, atau tahun rilis.

Column Name	#	Data type	Identity	Collation	Not Null	Default
film_id	1	serial4			[v]	nextval('film_film_id_seq'::regclass)
title	2	varchar(255)		<u>default</u>	[v]	
description	3	text		<u>default</u>	[ ]	
release_year	4	public."year"			[ ]	
language_id	5	int2			[v]	
rental_duration	6	int2			[v]	3
rental_rate	7	numeric(4, 2)			[v]	4.99
length	8	int2			[ ]	
replacement_cost	9	numeric(5, 2)			[v]	19.99
rating	10	public.mpaa_rating			[ ]	'G'::mpaa_rating
last_update	11	timestamp			[v]	now()
special_features	12	_text		<u>default</u>	[ ]	
fulltext	13	tsvector			[v]	

**Gambar 2.2.4. Tabel Film**

#### 5. Tabel *film\_category*

Tabel ini digunakan untuk menghubungkan film dengan kategori/genre. Ini memungkinkan analisis berdasarkan genre, misalnya, untuk mengetahui genre film mana yang paling populer di kalangan pelanggan.

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
film_id	1	int2			[v]		
category_id	2	int2			[v]		
last_update	3	timestamp			[v]	now()	

**Gambar 2.2.5. Tabel Film Category**

## 6. Tabel *inventory*

Tabel ini menyimpan informasi terkait dengan salinan fisik film yang tersedia di berbagai toko. Hal ini penting untuk mengelola stok film yang tersedia untuk disewa.

Column Name	#	Data type	Identity	Collation	Not Null	Default
inventory_id	1	serial4			[v]	nextval('inventory_inventory_id_seq'::regclass)
film_id	2	int2			[v]	
store_id	3	int2			[v]	
last_update	4	timestamp			[v]	now()

**Gambar 2.2.6.** Tabel Inventory

## 7. Tabel *language*

Tabel ini menyimpan informasi tentang bahasa yang digunakan dalam film. Ini memungkinkan analisis berdasarkan bahasa film dan bisa digunakan untuk melihat popularitas film dalam berbagai bahasa.

Column Name	#	Data type	Identity	Collation	Not Null	Default
language_id	1	serial4			[v]	nextval('language_language_id_seq'::regclass)
name	2	bpchar(20)		<u>default</u>	[v]	
last_update	3	timestamp			[v]	now()

**Gambar 2.2.7. Tabel Language**

## 8. Tabel *rental*

Tabel ini menyimpan informasi transaksi penyewaan, termasuk pelanggan, salinan film yang disewa, dan waktu penyewaan. Ini adalah tabel utama untuk menghitung frekuensi penyewaan film dan untuk analisis popularitas film.

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comments
<code>123 rental_id</code>	1	serial4			[v]	<code>nextval('rental_rental_id_seq'::regclass)</code>	
<code>123 rental_date</code>	2	timestamp			[v]		
<code>123 inventory_id</code>	3	int4			[v]		
<code>123 customer_id</code>	4	int2			[v]		
<code>123 return_date</code>	5	timestamp			[ ]		
<code>123 staff_id</code>	6	int2			[v]		
<code>123 last_update</code>	7	timestamp			[v]	<code>now()</code>	

**Gambar 2.2.8. Tabel Rental**

Penggunaan tabel-tabel ini memungkinkan untuk melakukan analisis mendalam tentang hubungan antar data, seperti frekuensi penyewaan berdasarkan kategori film, tren penyewaan dalam periode tertentu, serta profil pelanggan yang berkontribusi terhadap penyewaan tertinggi. Dengan relevansi data yang mencerminkan pola interaksi pelanggan dengan layanan penyewaan, database ini menjadi dasar yang ideal untuk mendukung penelitian dalam menganalisis popularitas film.

Dalam menganalisis popularitas film berdasarkan frekuensi penyewaan, hubungan antar tabel di *database* penting untuk memastikan data dapat diakses dan dianalisis dengan efisien. Berikut penjelasan hubungan antar tabel yang digunakan:

### 1. Hubungan antara tabel *city* dan *customer*

Setiap pelanggan berasal dari sebuah kota, yang berarti satu kota dapat memiliki banyak pelanggan (*one-to-many* atau satu ke banyak). Relasi ini

terjalin melalui kolom *city\_id* yang ada di tabel customer yang merujuk ke *city\_id* di tabel *city*.

## 2. Hubungan antara tabel *country* dan *city*

Setiap kota terletak di suatu negara, yang berarti satu negara dapat memiliki banyak kota (*one-to-many* atau satu ke banyak). Relasi ini terjalin melalui kolom *country\_id* yang ada di tabel *city* yang merujuk ke *country\_id* di tabel *country*.

## 3. Hubungan antara tabel *film* dan *film\_category*

Setiap film dapat dikategorikan ke dalam beberapa kategori, dan setiap kategori dapat berisi banyak film (*many-to-many* atau banyak ke banyak). Relasi ini dijembatani oleh tabel perantara *film\_category*, yang menghubungkan *film\_id* dan *category\_id*.

## 4. Hubungan antara tabel *film* dan *rental*

Setiap film dapat disewa banyak dalam transaksi yang tercatat di tabel *rental* (*one-to-many* atau satu ke banyak). Relasi ini terjalin melalui kolom *film\_id* di tabel *rental* yang merujuk ke *film\_id* di tabel *film*.

## 5. Hubungan antara tabel *customer* dan *rental*

Setiap pelanggan dapat menyewa banyak film, yang berarti satu pelanggan dapat memiliki banyak transaksi penyewaan di tabel *rental* (*one-to-many* atau satu ke banyak). Relasi ini terjalin melalui kolom *customer\_id* di tabel *rental* yang merujuk ke *customer\_id* di tabel *customer*.

## 6. Hubungan antara tabel *inventory* dan *film*

Setiap film dapat memiliki banyak salinan fisik yang tersedia untuk disewa, yang disimpan dalam tabel *inventory* (*one-to-many* atau satu ke banyak). Relasi ini terjalin melalui kolom *film\_id* di tabel *inventory* yang merujuk ke *film\_id* di tabel *film*.

## 7. Hubungan antara tabel *inventory* dan *rental*

Setiap salinan film yang ada di dalam inventaris dapat disewa berkali-kali (*one-to-many* atau satu ke banyak). Relasi ini terjalin melalui kolom *inventory\_id* di tabel rental yang merujuk ke *inventory\_id* di tabel *inventory*.

## 8. Hubungan antara tabel *language* dan film

Setiap film memiliki satu bahasa yang digunakan, yang berarti satu bahasa dapat digunakan untuk banyak film (*one-to-many* atau satu ke banyak). Relasi ini terjalin melalui kolom *language\_id* di tabel film yang merujuk ke *language\_id* di tabel *language*.

## 2.2 Tahapan Project

- **Input Database ke RDBMS (PostgreSQL)**

langkah -langkah:

1. **Instalasi PostgreSQL**

- Pastikan PostgreSQL terinstal dengan benar pada server.

2. **Input Database**

- Masukkan database baru di PostgreSQL, pada proyek ini yang dipakai yaitu *dvd rental database*.

3. **Connect ke Dbeaver:**

- Buka DBeaver dan tambahkan koneksi baru
- Di DBeaver, klik ikon "*New Database Connection*" (ikon database dengan tanda tambah) di toolbar atau pilih *Database* → *New Database Connection*.
- Pilih PostgreSQL dari daftar pilihan *database*, kemudian klik *Next*.
- Konfigurasi koneksi
- Setelah mengisi semua informasi, klik *Test Connection* untuk memastikan DBeaver dapat terhubung ke PostgreSQL. Jika berhasil, maka akan melihat pesan "*Success*".
- Klik *Finish* setelah koneksi berhasil.

4. **Validasi dan Pemeriksaan Data:**

- Pastikan data telah terinput dengan benar dan sesuai formatnya.

- Cek integritas data dengan melakukan pengecekan apakah ada duplikasi atau data yang tidak valid.
- **Pembuatan *Staging Area* (menggunakan Pentaho)**

Langkah - langkah:

### **1. Menyiapkan *Staging Area* di PostgreSQL**

- *Staging Area* adalah skema atau tabel sementara di database yang menyimpan data yang akan diproses lebih lanjut. Di PostgreSQL, dapat dibuat skema terpisah untuk *staging area*.

### **2. Instalasi dan Setup Pentaho Data Integration (PDI)**

- *Pentaho Data Integration (PDI)* atau *Kettle* adalah alat untuk menjalankan proses ETL. *Instal Pentaho Data Integration (PDI)* terlebih dahulu.

**Langkah-langkah instalasi Pentaho:**

1. *Download Pentaho Data Integration (PDI)* di situs resmi Pentaho.
2. Ekstrak file ZIP yang diunduh dan buka folder *data integration* untuk menjalankan Pentaho.

**Menyiapkan Pentaho:**

- Setelah membuka *Pentaho Data Integration (PDI)*, terdapat *Spoon*, yaitu antarmuka grafis untuk mendesain transformasi dan job.
- **Spoon** adalah alat untuk mendesain proses ETL yang akan digunakan untuk mengimpor data ke *staging area*.

### **3. Membuat Transformasi ETL untuk *Staging Area***

- Buka folder *staging* dan membuat koneksi *database*.
- Transformasi data, tambahkan beberapa langkah untuk memanipulasi data sebelum memuatnya ke *staging area*.
- Setelah data siap untuk dimuat ke *staging area*, tambahkan langkah *Database Output* untuk memuat data ke PostgreSQL. Pilih *Database*

*Output* dan pilih koneksi ke database PostgreSQL yang telah dibuat sebelumnya di DBeaver.

- Setelah selesai, klik *Run* untuk menjalankan transformasi dan pastikan data berhasil diproses dan dimuat ke dalam tabel *staging* di PostgreSQL.

#### 4. Validasi dan Pemeriksaan Data di Staging Area

- Pastikan bahwa data tidak mengandung nilai yang hilang (*missing*) atau salah format (misalnya tanggal yang tidak valid).

#### • Desain *Data Warehouse*

Langkah - langkah:

##### 1. Mendesain Tabel Fakta dan Tabel Dimensi yang relevan

- Tabel fakta berisi data numerik yang penting untuk analisis, seperti jumlah penyewaan dan pendapatan dari penyewaan. Tabel ini juga akan berisi *foreign key* yang mengacu pada tabel dimensi yang relevan.
- Tabel dimensi berisi data deskriptif yang akan digunakan untuk analisis lebih mendalam, misalnya, untuk melihat penyewaan berdasarkan kategori film, waktu, atau lokasi penyewaan.

##### 2. Memindahkan Data dari *Staging Area* ke *Data Warehouse*

- Setelah tabel dimensi dan fakta dibuat, langkah berikutnya adalah memindahkan data dari *Staging Area* ke *Data Warehouse*. Biasanya, data dalam *Staging Area* akan memiliki format yang lebih mentah dan perlu diproses atau ditransformasi sebelum dimuat ke tabel dimensi dan fakta.

##### 3. Pengujian dan Validasi Data

- Validasi Data di Tabel Dimensi: Pastikan data dimensi terisi dengan benar dan tidak ada duplikasi yang tidak diinginkan.
- Validasi Data di Tabel Fakta: Pastikan data cocok dengan yang tercatat di *staging area*.

- **Pembuatan *Data Mart***

*Data Mart* adalah subset dari *Data Warehouse* yang fokus pada topik analisis tertentu.

Langkah - langkah:

- 1. Menentukan Tujuan dan Fokus *Data Mart***

- Sebelum mulai membuat *Data Mart*, sangat penting untuk menentukan tujuan analisis yang ingin dicapai. Dalam hal ini, tujuan utama adalah untuk menganalisis popularitas film berdasarkan frekuensi penyewaan.

- 2. Merancang Struktur *Data Mart***

- Data Mart akan mengadopsi Star Schema, yang terdiri dari satu tabel fakta dan beberapa tabel dimensi. Tabel fakta akan menyimpan data transaksi penyewaan, sedangkan tabel dimensi akan menyimpan atribut deskriptif.

- 3. Memasukkan Data ke Tabel Dimensi**

- Setelah desain tabel selesai, langkah selanjutnya adalah memasukkan data dari *Staging Area* ke tabel dimensi. Data akan diproses untuk memasukkan atribut yang relevan ke setiap tabel dimensi.

- 4. Memasukkan Data ke Tabel Fakta**

- Setelah data dimensi terisi, langkah berikutnya adalah memindahkan data transaksi penyewaan ke dalam tabel fakta. Di sini, akan dihubungkan data dari *Staging Area* dengan *foreign key* yang sesuai dari tabel dimensi.

- 5. Pengujian dan Validasi Data**

- Validasi Data di Tabel Dimensi: Cek apakah data di tabel dimensi sudah terisi dengan benar dan tanpa duplikasi yang tidak diinginkan.
- Validasi Data di Tabel Fakta: Pastikan data di tabel fakta sudah sesuai dengan transaksi penyewaan yang tercatat di *staging area*.

- **Pembuatan *Dashboard* (menggunakan *Tableau*)**

Langkah - langkah:

- 1. Koneksi *Tableau* ke Database *PostgreSQL***

- Sebelum mulai membuat visualisasi, pastikan *Tableau* dapat terhubung dengan *database* yang menyimpan data mart.

## **2. Membuat *Data Source* di *Tableau***

- Setelah koneksi berhasil, konfigurasi *Data Source* di *Tableau*.

## **3. Membuat Visualisasi (*Charts*)**

- Setelah data berhasil dihubungkan, dapat mulai membuat visualisasi berdasarkan data yang tersedia.

## **4. Membuat *Dashboard* di *Tableau***

- Membuat Dashboard Baru
  1. Pilih Dashboard di bagian bawah Tableau.
  2. Klik New Dashboard untuk membuat dashboard kosong.
- Menambahkan Visualisasi ke *Dashboard*
  1. *Drag and Drop sheet* yang telah dibuat (seperti visualisasi berdasarkan kategori, lokasi, tren waktu, dll.) ke dalam *dashboard*.
  2. Sesuaikan ukuran dan posisi visualisasi agar *dashboard* terlihat rapi.
  3. Gunakan *Tiled* atau *Floating* untuk mengatur layout visualisasi di *dashboard*.
- Menambahkan *Filter* Interaktif. Untuk membuat *dashboard* interaktif, dapat menambahkan *filter*:
  1. Pilih *filter* yang relevan.
  2. *Drag filter* ke *dashboard* untuk memungkinkan pengguna memilih kategori, waktu, atau lokasi tertentu.
  3. Gunakan *Actions* untuk membuat interaksi antar visualisasi.

## **2.3 Tools dan Teknologi**

### **1. *PostgreSQL (Relational Database Management System - RDBMS)***

*PostgreSQL* digunakan untuk menyimpan dan mengelola database proyek. Ini termasuk mengelola *Data Mart* yang terdiri dari tabel fakta dan dimensi. *PostgreSQL* merupakan database relasional yang kuat dan stabil, sangat cocok untuk mengelola data yang besar dan kompleks, seperti data penyewaan film yang terkait dengan pelanggan, film, lokasi, dan waktu.

#### **Alasan Pemilihan:**

- **Kinerja Tinggi dan Skalabilitas:** *PostgreSQL* memiliki kemampuan untuk menangani jumlah data yang sangat besar dengan kinerja yang optimal. Untuk proyek yang mengelola data penyewaan film dalam skala besar, *PostgreSQL* sangat ideal karena dapat mengelola query yang kompleks dengan cepat.
- **Dukungan untuk SQL dan Fungsi Lanjutan:** *PostgreSQL* mendukung bahasa *query SQL* yang sangat kuat dan memiliki banyak fungsi tambahan untuk analisis data, seperti agregasi, *join*, dan *window functions*.
- **Kompatibilitas dengan Tools Lain:** *PostgreSQL* memiliki konektivitas yang baik dengan berbagai tools *ETL* (seperti *Pentaho*) dan *Business Intelligence* (seperti *Tableau*), yang memungkinkan integrasi yang mulus dalam alur kerja proyek.
- **Open Source:** *PostgreSQL* adalah perangkat lunak sumber terbuka, sehingga mengurangi biaya lisensi untuk organisasi dan memungkinkan fleksibilitas dalam penggunaan.

## 2. *DBeaver (Database Management Tool)*

*DBeaver* adalah alat manajemen database yang mendukung berbagai jenis database, termasuk *PostgreSQL*, *MySQL*, *Oracle*, dan lainnya. Alat ini digunakan untuk mengelola, mengakses, dan menganalisis data dalam *database* melalui antarmuka yang *user-friendly*. Dalam proyek ini, *DBeaver* digunakan untuk mengakses *Data Mart* di *PostgreSQL*, menjalankan *query SQL*, dan memverifikasi hasil transformasi data dari proses *ETL*.

### Alasan Pemilihan:

- **Kompatibilitas Multi-Database:** Mendukung banyak sistem database, termasuk *PostgreSQL*, memungkinkan pengelolaan berbagai jenis database dalam satu platform.
- **Antarmuka Ramah Pengguna:** Memiliki tampilan yang intuitif dan fitur visual untuk mempermudah eksplorasi tabel, skema, dan data.
- **Fungsi Query yang Kuat:** Menyediakan editor *SQL* dengan fitur *autocomplete*, *syntax highlighting*, dan *query execution* untuk mempermudah penulisan dan *debugging query*.
- **Visualisasi Data:** Mendukung visualisasi hasil *query* dalam bentuk tabel, diagram, atau grafik sederhana, membantu memahami data lebih cepat.
- **Integrasi dengan ETL dan BI Tools:** *DBeaver* dapat digunakan bersamaan dengan *Pentaho* atau *Tableau*, mempermudah alur kerja analisis data dan integrasi antar alat.

## 3. *Pentaho (ETL dan Integrasi Data)*

Pentaho digunakan sebagai alat ***ETL (Extract, Transform, Load)*** untuk memindahkan data dari sumber ke dalam *Data Mart* di *PostgreSQL*. Dalam proyek ini, Pentaho digunakan untuk membersihkan, mentransformasi, dan memuat data penyewaan film dari sumber yang berbeda ke dalam database yang telah disiapkan.

#### **Alasan Pemilihan:**

- **Mudah Digunakan dengan *UI* yang Ramah Pengguna:** Pentaho memiliki antarmuka grafis yang memudahkan pengguna dalam mendesain alur *ETL* tanpa harus menulis banyak kode. Hal ini membantu dalam mempercepat proses pengembangan.
- **Integrasi dengan Berbagai Sumber Data:** Pentaho dapat mengintegrasikan berbagai sumber data seperti file CSV, *database SQL*, aplikasi *cloud*, dan API. Ini memudahkan pengambilan data dari berbagai sumber dan pengolahan sebelum dimuat ke dalam *Data Mart*.
- **Fleksibilitas dan Skalabilitas:** Pentaho dapat menangani data dalam volume besar, yang sangat penting untuk proyek yang bekerja dengan banyak data penyewaan film.
- ***Open Source* dengan *Versi Enterprise*:** Pentaho memiliki versi *open-source* yang kuat dan juga tersedia dalam versi *enterprise* untuk fitur tambahan dan dukungan teknis, memungkinkan fleksibilitas dalam memilih sesuai dengan anggaran dan kebutuhan proyek.
- **Fitur Transformasi yang Kuat:** Pentaho menyediakan berbagai alat transformasi data, seperti agregasi, penggabungan, pembersihan data, dan transformasi kompleks yang sangat membantu dalam menyiapkan data untuk analisis.

#### **4. *Tableau* (Visualisasi Data)**

*Tableau* adalah alat *Business Intelligence (BI)* yang digunakan untuk visualisasi data dan pembuatan *dashboard* interaktif. Tujuan utamanya adalah untuk memungkinkan pengguna (analisis, manajer, atau pemangku kepentingan lainnya) untuk mengakses, menganalisis, dan menggali wawasan terkait analisis secara intuitif dan efisien.

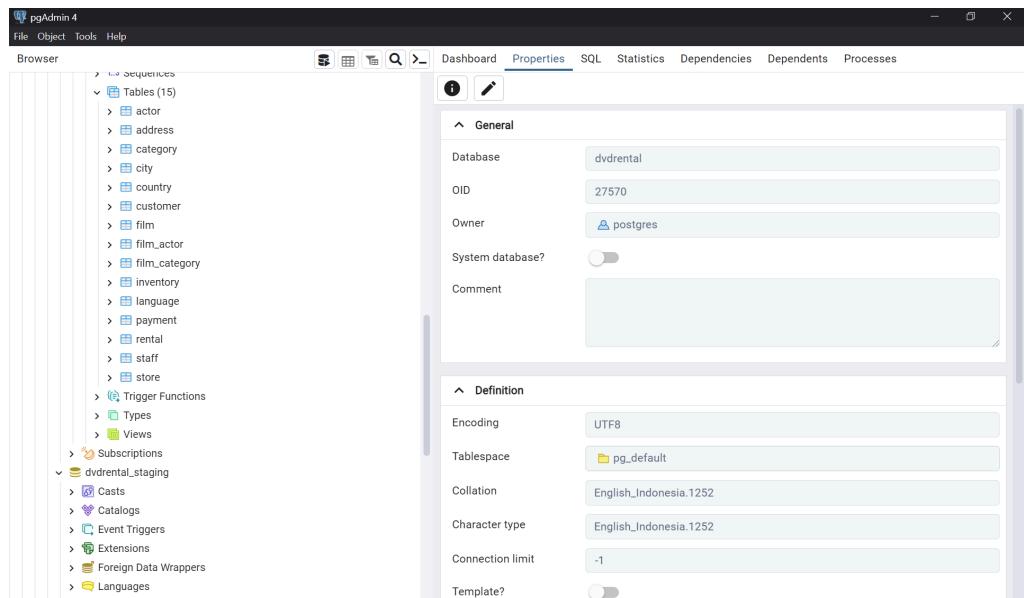
### **Alasan Pemilihan:**

- **Visualisasi Interaktif yang Kuat:** Tableau dikenal dengan kemampuannya untuk membuat visualisasi yang menarik dan interaktif, memungkinkan pengguna untuk menggali lebih dalam data dan membuat keputusan berbasis data dengan lebih mudah.
- **Kemampuan untuk Mengelola Data dalam Skala Besar:** *Tableau* dapat terhubung langsung ke berbagai sumber data (termasuk *PostgreSQL*) dan menangani volume data besar, yang penting untuk proyek dengan data penyewaan film dalam skala besar.
- **User-Friendly:** Tableau memungkinkan pembuatan dashboard dan visualisasi dengan *drag and drop*, tanpa perlu keterampilan pengkodean yang mendalam, sehingga cocok untuk pengguna yang lebih berorientasi pada bisnis.
- **Kompatibilitas dengan *PostgreSQL*:** Tableau terintegrasi dengan sangat baik dengan *PostgreSQL*, memungkinkan penggunaan data dari *Data Mart* secara langsung untuk analisis dan visualisasi.

## BAB III

### IMPLEMENTASI & HASIL

#### 3.1 Input Database



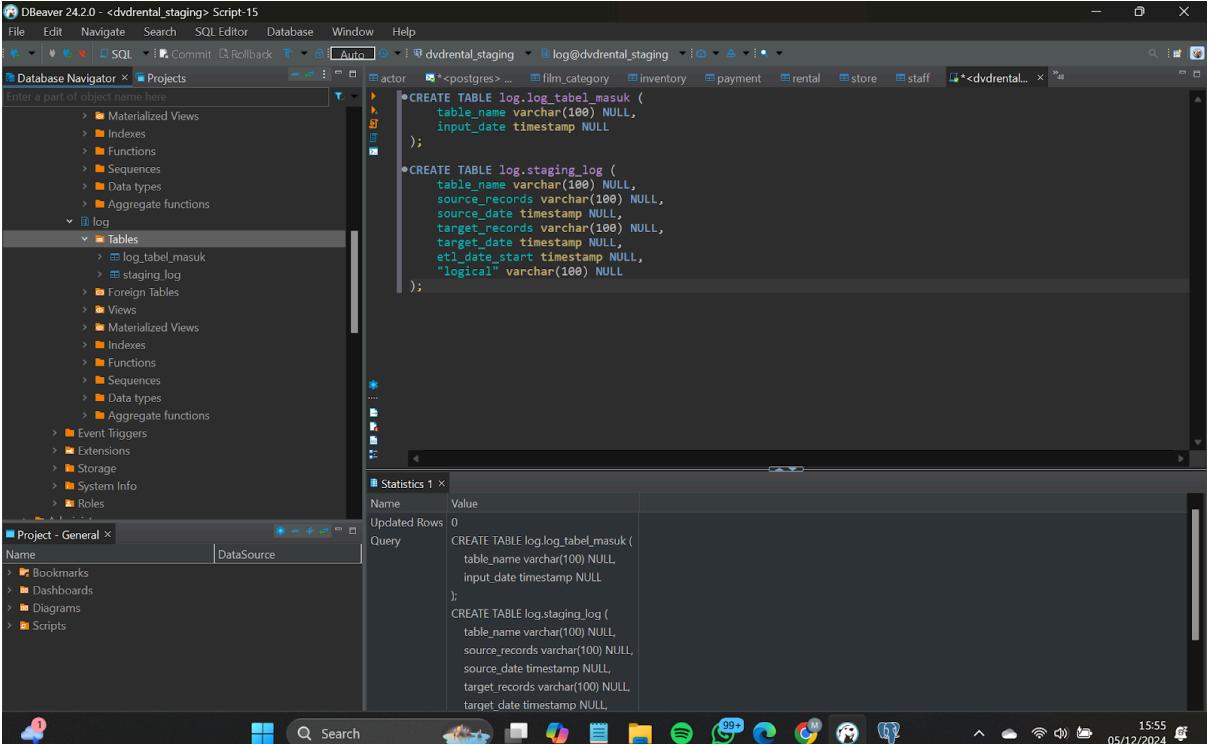
**Gambar 3.1.1 Hasil restore database**

Gambar 3.1.1 menunjukkan hasil *restore database dvrental* menggunakan pgAdmin 4. Database ini memiliki berbagai tabel yang mencakup elemen-elemen seperti *actor*, *address*, *category*, *customer*, *film*, dan lainnya, yang tersusun rapi dalam struktur hierarkis. Database ini disimpan dengan *encoding UTF8* dan menggunakan *Tablespace pg\_default*. Proses ini merupakan langkah awal untuk mempersiapkan data dalam *PostgreSQL* sebelum dipindahkan ke *DBeaver* untuk tahap awal *staging*.

Proses *restore* ini bertujuan untuk memastikan semua data termasuk struktur tabel, relasi, dan tipe data, telah diimpor dengan benar ke dalam *PostgreSQL*. Selanjutnya, *database* ini akan diakses menggunakan *DBeaver* untuk mendukung proses *Extract, Transform, dan Load (ETL)* sebagai bagian dari integrasi data yang lebih luas. Hal ini memastikan bahwa *database* siap digunakan untuk analisis data atau kebutuhan lainnya.

### 3.2 Pembuatan Staging Area

Proses *staging area* dilakukan dengan menggunakan *DBeaver* dan *Pentaho Data Integration (PDI)*, dimulai dengan menyalin struktur tabel dari database asli *dvdrental*. Tabel-tabel ini kemudian dipindahkan ke koneksi baru bernama *dvdrental\_staging*, dengan melakukan penyesuaian pada *query SQL* untuk tahap *staging*, seperti mengganti nama *schema public* menjadi *staging* dan menghapus foreign key untuk menghilangkan dependensi antar tabel. Selanjutnya, properti kolom pada tabel tertentu, seperti tabel *address*, dimodifikasi agar lebih fleksibel, dan beberapa indeks yang tidak diperlukan dihapus, termasuk *film\_fulltext.idx* pada tabel *film*. Untuk mendukung kebutuhan *staging*, tipe data pada kolom tertentu, seperti kolom *fulltext* di tabel *film*, juga diubah menjadi *varchar(255)*. Seluruh langkah-langkah ini telah didokumentasikan dan dimasukkan ke lampiran laporan.



The screenshot shows the DBeaver interface with the following details:

- Toolbar:** File, Edit, Navigate, Search, SQL Editor, Database, Window, Help.
- Database Navigator:** Shows a tree view of objects in the current database:
  - Tables: log\_table\_masuk, staging\_log.
  - Views.
  - Materialized Views.
  - Indexes.
  - Functions.
  - Sequences.
  - Data types.
  - Aggregate functions.
  - Event Triggers.
  - Extensions.
  - Storage.
  - System Info.
  - Roles.
- SQL Editor:** Contains two CREATE TABLE statements:

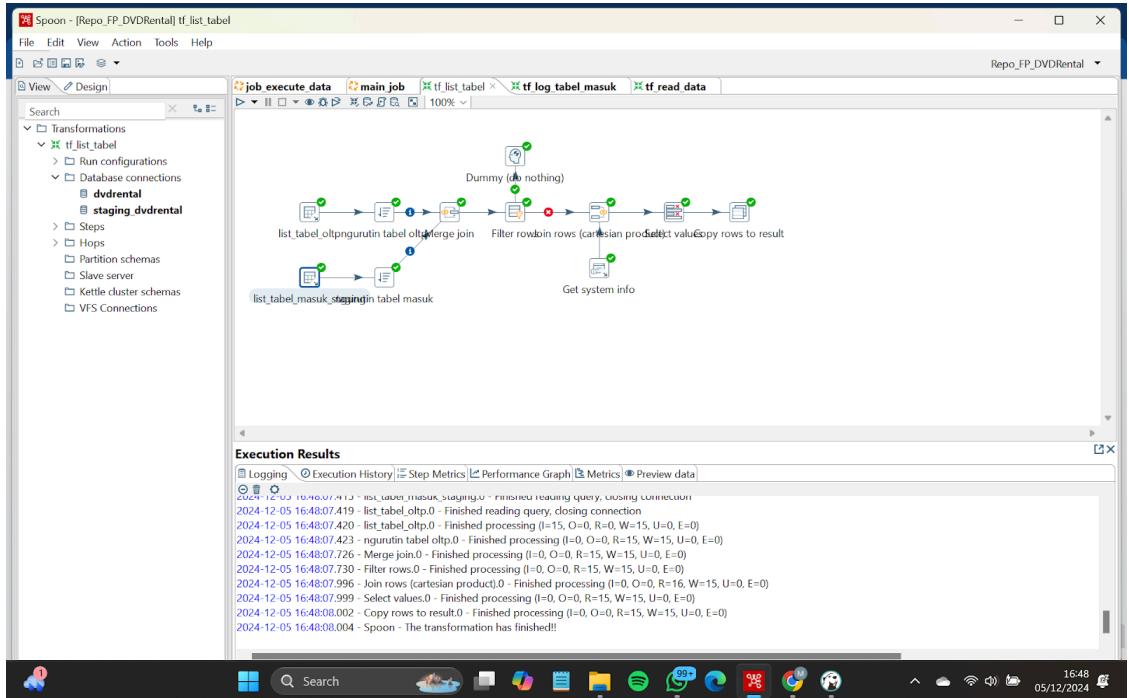
```
CREATE TABLE log.log_table_masuk (
    table_name varchar(100) NULL,
    input_date timestamp NULL
);

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
);
```
- Statistics 1:** A table showing statistics for the queries:

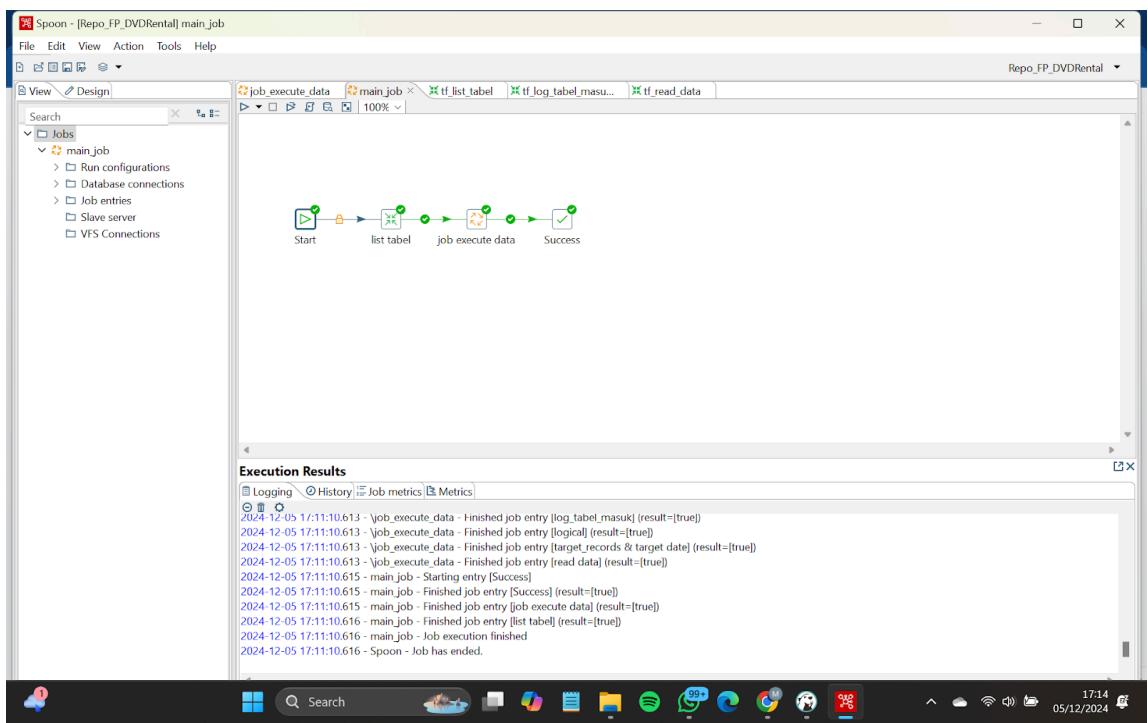
Name	Value
Updated Rows	0
Query	CREATE TABLE log.log_table_masuk (     table_name varchar(100) NULL,     input_date timestamp NULL );  CREATE TABLE log.staging_log (     table_name varchar(100) NULL,     source_records varchar(100) NULL,     source_date timestamp NULL,     target_records varchar(100) NULL,     target_date timestamp NULL,     etl_date_start timestamp NULL,     "logical" varchar(100) NULL );

**Gambar 3.2.1. Membuat schema log**

Setelah struktur tabel *staging* selesai, schema *log* dibuat sesuai dengan gambar 3.2.1. untuk mencatat aktivitas dan data yang masuk ke *staging*. Tabel *Masuk Staging* akan digunakan untuk menyimpan data yang akan diproses masuk ke *staging*. Sedangkan Tabel *Log Staging* digunakan untuk mencatat proses log dan aktivitas pada *staging*. Kemudian proses dilanjutkan dengan membuka aplikasi PDI untuk membuat repository baru dan mempersiapkan direktori file transformasi yang akan digunakan.



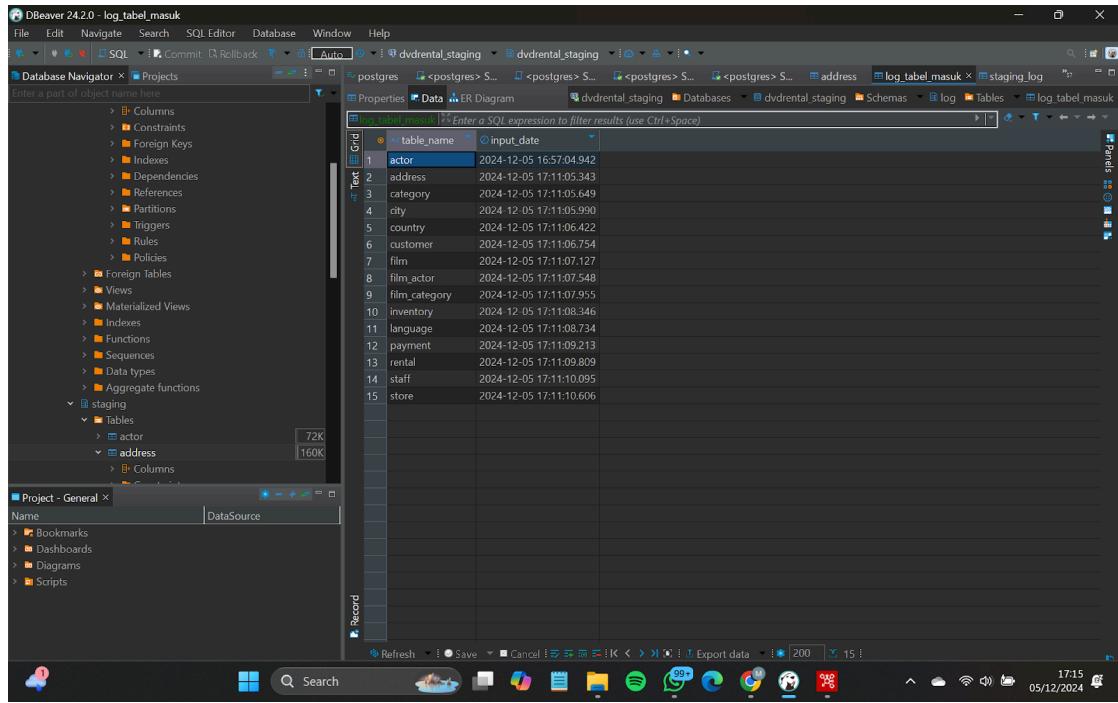
**Gambar 3.2.2. Menjalankan Transformasi *tf\_list\_tabel***



**Gambar 3.2.3. Menjalankan *main.job***

Langkah berikutnya adalah membuat koneksi database di PDI untuk *dvrental* dan *dvrental\_staging*, yang kemudian diuji untuk memastikan keberhasilan koneksi. Transformasi seperti *tf\_list\_tabel* dijalankan sesuai dengan gambar 3.2.2, diikuti

dengan eksekusi atau running main.job sesuai dengan gambar 3.2.3. Hal ini penting dilakukan untuk memproses data dari sumber ke staging area.



**Gambar 3.2.4. Membuka Dbeaver dan Memeriksa Tabel log\_table\_masuk**

table_name	source_records	source_date	target_records	target_date	etl_date_start	logical
actor	200	2024-12-05 16:57:04.818	200	2024-12-05 16:57:04.863	2024-12-05 16:57:03.000	success
address	603	2024-12-05 16:57:59.850	603	2024-12-05 17:11:05.156	2024-12-05 16:57:58.000	success
category	603	2024-12-05 17:03:12.098	603	2024-12-05 17:11:05.156	2024-12-05 17:03:11.000	success
city	603	2024-12-05 17:05:10.528	603	2024-12-05 17:11:05.156	2024-12-05 17:05:09.000	success
country	16	2024-12-05 17:11:05.411	16	2024-12-05 17:11:05.577	2024-12-05 17:11:04.000	success
customer	600	2024-12-05 17:11:05.842	600	2024-12-05 17:11:05.912	2024-12-05 17:11:04.000	success
film	1000	2024-12-05 17:11:06.825	1000	2024-12-05 17:11:06.350	2024-12-05 17:11:04.000	success
film_actor	5462	2024-12-05 17:11:07.190	5462	2024-12-05 17:11:07.462	2024-12-05 17:11:04.000	success
film_category	1000	2024-12-05 17:11:07.623	1000	2024-12-05 17:11:07.883	2024-12-05 17:11:04.000	success
inventory	4581	2024-12-05 17:11:08.024	4581	2024-12-05 17:11:08.160	2024-12-05 17:11:04.000	success
language	6	2024-12-05 17:11:08.408	6	2024-12-05 17:11:08.552	2024-12-05 17:11:04.000	success
payment	14596	2024-12-05 17:11:08.798	14596	2024-12-05 17:11:09.138	2024-12-05 17:11:04.000	success
rental	16044	2024-12-05 17:11:09.267	16044	2024-12-05 17:11:09.728	2024-12-05 17:11:04.000	success
staff	2	2024-12-05 17:11:09.888	2	2024-12-05 17:11:09.916	2024-12-05 17:11:04.000	success
store	2	2024-12-05 17:11:10.155	2	2024-12-05 17:11:10.314	2024-12-05 17:11:04.000	success

**Gambar 3.2.5. Pengecekan pada Tabel staging\_log**

Setelah proses di Pentaho selesai dapat membuka *DBeaver* kembali dan melakukan pengecekan pada data yang masuk ke tabel *log\_tabel\_masuk* sesuai dengan gambar 3.2.4 dan tabel *staging\_log* sesuai dengan gambar 3.2.5. Hal ini penting dilakukan guna memastikan semua aktivitas tercatat dengan baik dan sudah sesuai. Kemudian pilih salah satu tabel dalam *schema staging* dan lakukan pengecekan isi tabel. Dalam langkah ini dapat memverifikasi bahwa data yang dimigrasikan dari sumber telah masuk dengan benar dan sesuai dengan struktur tabel di staging area, sesuai dengan gambar 3.2.6 dibawah ini.

The screenshot shows the DBeaver interface with the following details:

- File Bar:** File, Edit, Navigate, Search, SQL Editor, Database, Window, Help.
- Toolbar:** Auto, Commit, Rollback, etc.
- Database Navigator:** Shows connections to 'postgres' and 'dvrental\_staging' databases, and tables like 'address' and 'log\_tabel\_masuk'.
- Properties View:** Shows columns for 'address\_id', 'address', 'district', 'city\_id', 'postal\_code', and 'phone'.
- Table View:** A grid showing 27 rows of address data. The first few rows are:
 

	address_id	address	district	city_id	postal_code	phone
1	1	47 MySakila Drive	[NULL]	Alberta	300	[NULL]
2	2	28 MySQL Boulevard	[NULL]	QLD	576	[NULL]
3	3	23 Workhaven Lane	[NULL]	Alberta	300	[NULL]
4	4	4 1411 Lillydale Drive	[NULL]	QLD	576	[NULL]
5	5	1913 Hanoi Way	[NULL]	Nagasaki	463	35200
6	6	1121 Loja Avenue	[NULL]	California	449	17886
7	7	692 Jolley Street	[NULL]	Attika	38	83579
8	8	8 1566 Ingle Manor	[NULL]	Mandalay	349	53561
9	9	9 53 Idu Parkway	[NULL]	Nantou	361	42399
10	10	10 1795 Santiago de Compostela Way	[NULL]	Texas	295	18743
11	11	11 900 Santiago de Compostela Parkway	[NULL]	Central Serbia	280	93896
12	12	12 478 Jolley Way	[NULL]	Hamilton	200	77948
13	13	13 613 Korolev Drive	[NULL]	Masqat	329	45844
14	14	14 1531 Sal Drive	[NULL]	Esfahan	162	53628
15	15	15 1542 Tarlac Parkway	[NULL]	Kanagawa	440	1027
16	16	16 808 Rhopal Manor	[NULL]	Haryana	582	10672
17	17	17 270 Amroha Parkway	[NULL]	Osmanniye	384	29610
18	18	18 770 Bydgoszcz Avenue	[NULL]	California	120	16266
19	19	19 419 Iligan Lane	[NULL]	Madhya Pradesh	76	72878
20	20	20 360 Toulouse Parkway	[NULL]	England	495	54308
21	21	21 270 Toulon Boulevard	[NULL]	Kalmykia	156	81766
22	22	22 320 Brest Avenue	[NULL]	Kaduna	252	43331
23	23	23 1417 Lancaster Avenue	[NULL]	Northern Cape	267	72192
24	24	24 1688 Okara Way	[NULL]	Northwest Border Prov	327	21954
25	25	25 262 A Corus (La Corus) Parkway	[NULL]	Dhaka	525	34418
26	26	26 28 Charlotte Amalie Street	[NULL]	Rabat-Sal-Zemmour-Z	443	37551
27	27	27 1780 Hino Boulevard	[NULL]	Liepaja	303	7716
- Bottom Status Bar:** Refresh, Save, Cancel, etc. Shows '200 rows fatched' and the date '05/12/2024'.

**Gambar 3.2.6. Memeriksa Salah Satu Tabel di Staging Area**

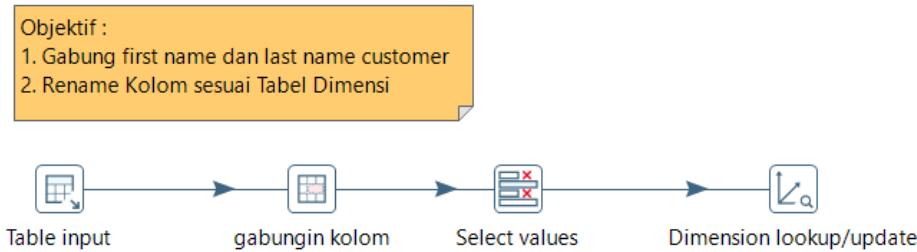
Dengan selesainya tahap staging dan verifikasi data, proses ini memberikan landasan kuat untuk langkah berikutnya, yaitu merancang *Data Warehouse* (DWH) yang siap digunakan untuk analisis data tingkat lanjut.

### 3.3 Desain Data Warehouse

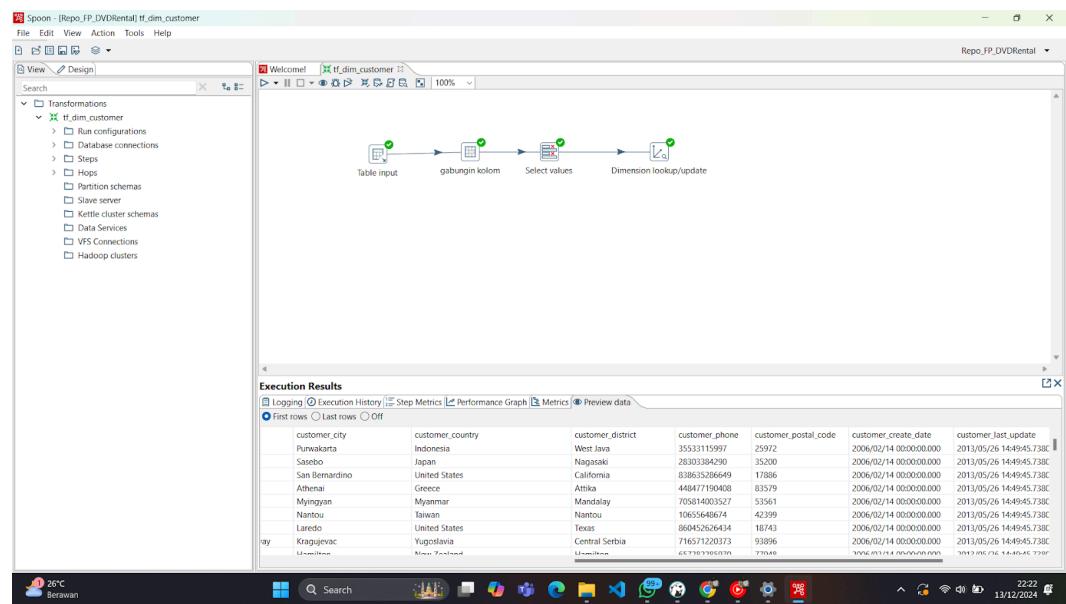
Buat database baru di DBeaver dengan nama *dw\_rental*. Untuk membuat database baru di DBeaver dengan nama *dw\_rental* menggunakan PostgreSQL, pastikan DBeaver telah terhubung ke server PostgreSQL. Setelah koneksi berhasil, klik kanan pada koneksi PostgreSQL yang aktif di panel Database Navigator, lalu pilih opsi *SQL Editor* untuk membuka editor SQL. Di editor tersebut, masukkan perintah berikut untuk membuat database baru ***CREATE DATABASE dw\_rental;***

Setelah mengetik perintah, jalankan dengan menekan tombol *Execute SQL Statement* (ikon petir) atau tekan *Ctrl+Enter*. Jika perintah berhasil dijalankan, database baru bernama *dw\_rental* akan otomatis muncul di daftar database di Database Navigator dan dapat menggunakan untuk membuat tabel atau menyimpan data sesuai kebutuhan.

Untuk menghubungkan Target Schema, drag step *Table Input* ke canvas di Pentaho. Klik dua kali pada step tersebut, lalu buat koneksi ke database dengan mengisi detail seperti *host*, *port*, nama *database*, *username*, dan *password*. Setelah koneksi dibuat, tambahkan query SQL untuk mengambil data dari tabel yang diinginkan. Pastikan koneksi berhasil dengan mengetesnya, lalu lakukan preview data untuk memastikan data dapat diambil dengan benar.



**Gambar 3.3.1. ETL Customer**



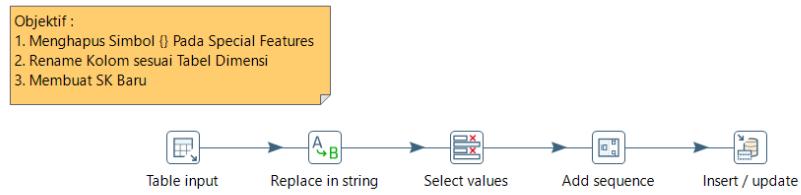
**Gambar 3.3.2 Running Pentaho**

- Setelah sudah membuat connection sambungkan target schema nya dengan table customer yg telah distaging. Digunakan pula join dengan tujuan sebagai berikut:

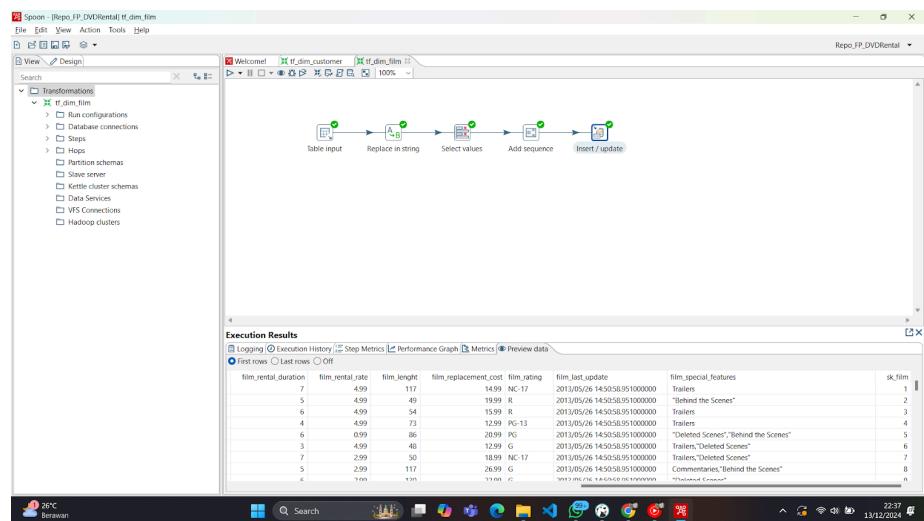
#### ➤ Relasi Antar Tabel

- address menyimpan data alamat pelanggan.
- city menyimpan informasi kota dari alamat.
- country menyimpan informasi negara dari kota tersebut.

4. JOIN digunakan untuk menggabungkan data dari tabel-tabel tersebut agar lebih lengkap.
- Menghindari Redudansi
  - Pengambilan Data yang Lebih Komprehensif, Contohnya:
    1. address untuk detail alamat pelanggan.
    2. city\_name untuk nama kota pelanggan.
    3. country\_name untuk nama negara pelanggan.
  - Menggabungkan kolom first name dan last name pada customer dengan menggunakan Concat Field.
  - Menggunakan Select Value untuk mengubah nama tabel sesuai dengan nama dimensinya.
  - Pada langkah terakhir, gunakan Dimension Lookup/Update untuk membuat ID baru (sk\_customer) pada tabel dimensi customer. Diperlukan juga untuk execute SQL terlebih dahulu sebelum menjalankan Pentaho.
  - Setelah SQL dieksekusi, jalankan ETL di Pentaho untuk menyimpan data ke database.



**Gambar 3.3.3 ETL Dimensi Film**

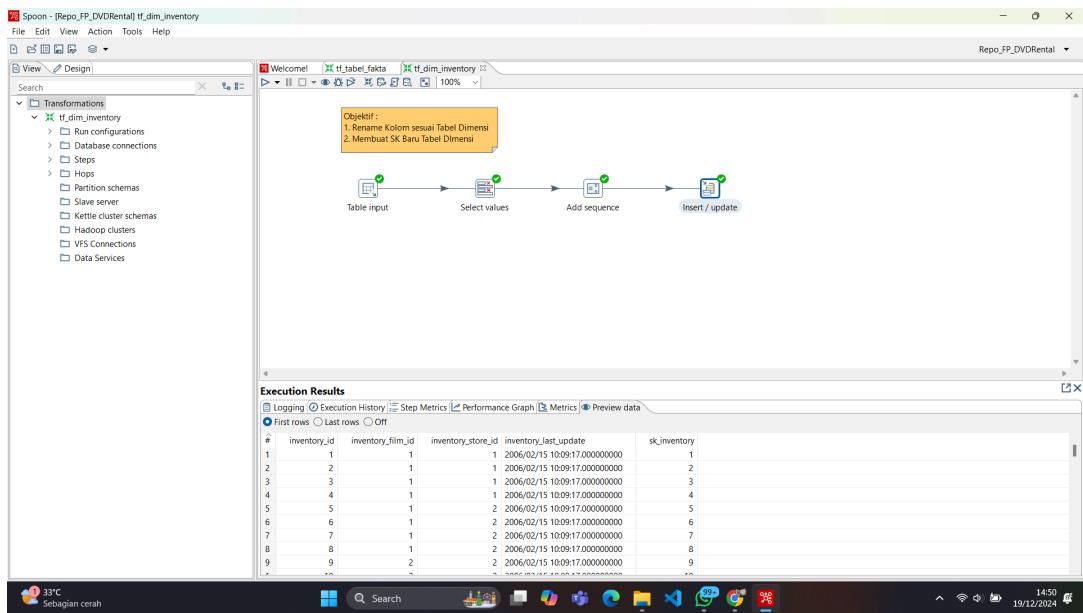


**Gambar 3.3.4 Running Pentaho**

- Buat koneksi ke database seperti yang dilakukan sebelumnya pada dimensi customer. Pastikan konfigurasi koneksi benar sehingga Pentaho dapat terhubung ke database yang akan digunakan untuk proses transformasi data.
- Ambil data dari tabel staging film. Tabel ini berisi data mentah yang akan diproses dan dimasukkan ke dalam tabel dimensi yang sesuai. Gunakan step input untuk membaca data dari tabel staging.
- Lakukan preview data untuk memastikan bahwa data dari tabel staging berhasil diambil dan ditampilkan dengan benar. Langkah ini penting untuk memastikan tidak ada kesalahan dalam konfigurasi input atau koneksi.
- Bersihkan data menggunakan *Replace in String*. Pada tahap ini, hapus tanda kurung {} yang terdapat pada kolom *special\_features* untuk memastikan data lebih rapi dan siap diproses lebih lanjut.
- Ubah nama tabel menggunakan Select Values agar tabel yang dihasilkan sesuai dengan nama dimensi, seperti "*film\_filmtitle*". Selain itu, pada tahap ini, kolom yang tidak relevan dapat dihapus, dan urutan kolom dapat disesuaikan sesuai kebutuhan struktur dimensi.
- Tambahkan surrogate key (SK) untuk dimensi film dengan menggunakan step add sequence. SK adalah kunci unik yang digunakan untuk setiap entri dalam tabel dimensi, memastikan integritas data dalam sistem data warehouse.
- Masukkan data ke dalam tabel database dengan menggunakan Insert/Update. Konfigurasikan step ini agar data dimasukkan ke tabel dimensi yang sesuai. Proses ini memastikan data yang telah diproses tersimpan dengan benar di database.
- Jalankan perintah SQL terlebih dahulu untuk memvalidasi struktur tabel dimensi di database, seperti mengecek kolom dan tipe data yang sesuai, sebelum transformasi dijalankan.
- Eksekusi transformasi di Pentaho untuk menyelesaikan proses pemindahan data dari tabel staging film ke tabel dimensi. Jika langkah-langkah ini dilakukan dengan benar, data akan tersimpan di tabel dimensi tanpa kesalahan.



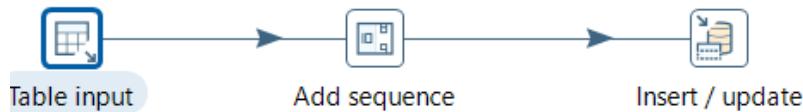
**Gambar 3.3.5 ETL Dimensi Inventory**



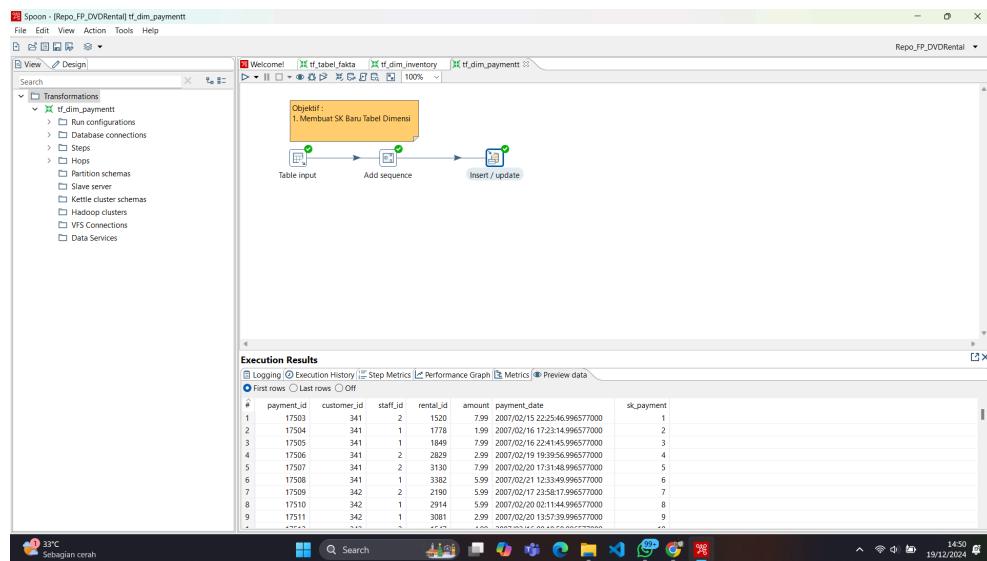
**Gambar 3.3.6. Running Pentaho**

- Mulailah dengan membuat koneksi ke database seperti langkah yang dilakukan sebelumnya pada dimensi film. Koneksi ini memastikan Pentaho dapat terhubung ke database yang akan digunakan untuk mengambil dan menyimpan data.
- Ambil data dari tabel staging inventory yang akan diproses lebih lanjut. Gunakan step input untuk membaca data dari staging inventory dan memasukkannya ke dalam pipeline transformasi.
- Lakukan preview data untuk memastikan bahwa data dari tabel staging berhasil diambil. Pastikan data muncul dengan benar di layar preview tanpa adanya error pada konfigurasi input atau koneksi database.
- Gunakan step Select Values untuk mengganti nama kolom agar sesuai dengan struktur yang dibutuhkan oleh dimensi inventory. Pada tahap ini, kolom yang tidak relevan juga dapat dihapus, dan nama kolom dapat disesuaikan untuk menjaga konsistensi dengan standar penamaan.
- Tambahkan sk\_inventory sebagai primary key untuk tabel dim inventory menggunakan step Add Sequence. Primary key ini memberikan identifikasi unik pada setiap entri dalam tabel dimensi, memastikan data terorganisasi dengan baik.
- Tambahkan data yang sudah diproses ke database menggunakan step Insert/Update. Langkah ini menyimpan data ke tabel dim inventory di database target. Pastikan konfigurasi tabel dan kolom dalam step ini sesuai dengan struktur tabel di database.
- Sebelum menjalankan transformasi di Pentaho, eksekusi perintah SQL terlebih dahulu pada step terakhir, agar tabel masuk ke database terlebih dahulu.
- Eksekusi transformasi di Pentaho untuk menyelesaikan proses dan memindahkan data dari staging inventory ke tabel dim inventory. Data akan

tersimpan di database jika semua langkah di atas telah dilakukan dengan benar.



**Gambar 3.3.7. ETL Dimensi Payment**

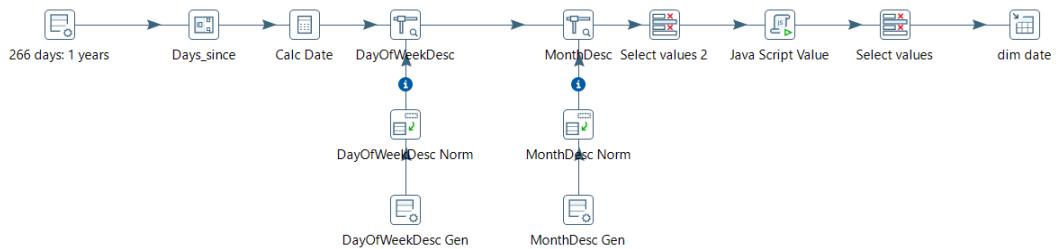


**Gambar 3.3.8. Running Pentaho**

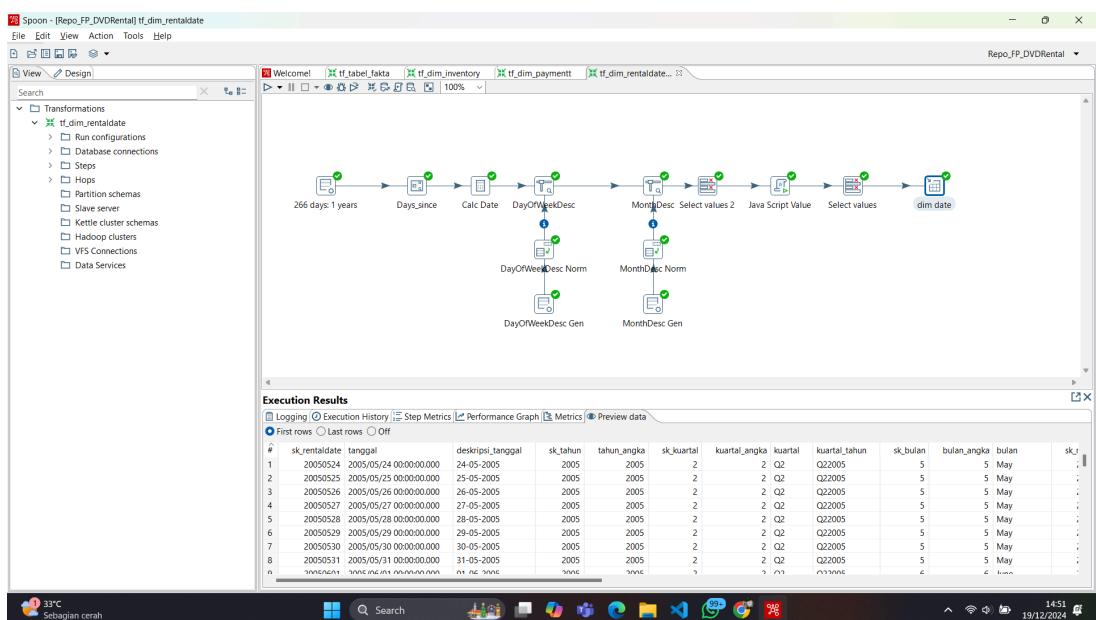
- Mulailah dengan membuat koneksi ke database seperti langkah sebelumnya pada dimensi inventory. Langkah ini memastikan bahwa Pentaho dapat terhubung dengan server database yang akan digunakan. Pastikan semua konfigurasi koneksi, seperti nama host, port, nama database, username, dan password, sudah benar untuk menghindari kendala teknis.
- Setelah koneksi berhasil dibuat, data dari tabel **staging payment** diambil sebagai sumber data mentah. Gunakan step input di Pentaho untuk membaca data dari tabel ini. Data dalam staging payment akan diproses lebih lanjut untuk dimasukkan ke dalam tabel dimensi yang sesuai.
- Lakukan **preview data** pada step input untuk memastikan bahwa data dari tabel staging payment berhasil diambil dan ditampilkan dengan benar. Langkah ini penting untuk memverifikasi bahwa konfigurasi koneksi, query, atau mapping kolom telah dilakukan dengan benar, sehingga data dapat digunakan tanpa kesalahan.
- Tambahkan **sk\_payment** sebagai surrogate key untuk tabel dimensi payment menggunakan step **Add Sequence**. Surrogate key ini berfungsi sebagai

primary key pada tabel dimensi, yang memberikan identifikasi unik pada setiap entri di tabel tersebut. Proses ini memastikan integritas data dalam data warehouse dan mempermudah pelacakan data.

- Pada langkah terakhir, gunakan step **Insert/Update** untuk memasukkan data yang telah diproses ke tabel dimensi di database. Konfigurasikan step ini agar sesuai dengan struktur tabel di database target, sehingga data dapat disimpan dengan benar.
- Sebelum menjalankan transformasi di Pentaho, eksekusi perintah SQL terlebih dahulu pada step terakhir, agar tabel masuk ke database terlebih dahulu.
- Setelah semua langkah selesai, jalankan transformasi di Pentaho untuk menyelesaikan proses. Data dari staging payment akan dipindahkan ke tabel dim payment di database, dan transformasi dianggap berhasil jika semua data tersimpan tanpa error.

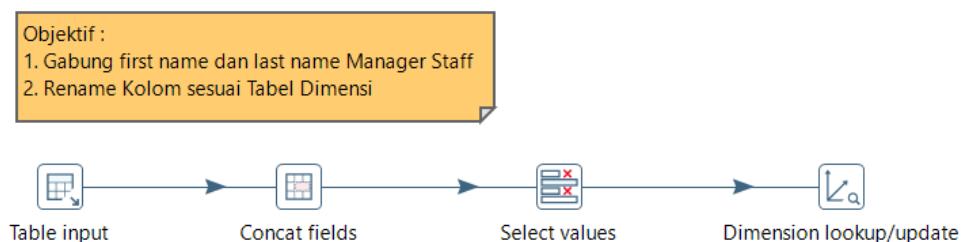


**Gambar 3.3.9. ETL Dimensi Rental Date**

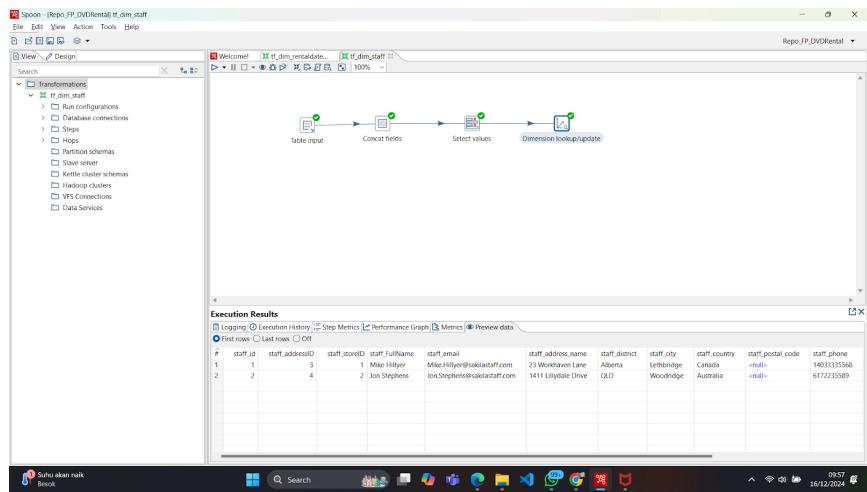


**Gambar 3.3.10. Running Pentaho**

- 266 days: 1 years: Tahap ini digunakan untuk menentukan rentang waktu (periode) yang ingin diproses, yaitu 266 hari atau sekitar 1 tahun. Rentang ini akan menjadi dasar untuk membuat dimensi tanggal.
- Days\_since: Di tahap ini, dilakukan perhitungan jumlah hari sejak suatu tanggal awal. Biasanya, ini digunakan untuk menghasilkan data tanggal secara bertahap atau berkelanjutan.
- Calc Date: Langkah ini menghitung tanggal yang sesuai berdasarkan perhitungan di tahap sebelumnya. Dengan kata lain, Calc Date menghasilkan kolom tanggal yang akan digunakan dalam dimensi.
- DayOfWeekDesc: Proses ini menghitung atau menambahkan deskripsi nama hari (Senin, Selasa, Rabu, dst.) ke dalam dataset berdasarkan kolom tanggal.
- DayOfWeekDesc Norm: Normalisasi deskripsi hari agar sesuai format standar.
- DayOfWeekDesc Gen: Tahap ini membuat atau menghasilkan data deskripsi nama hari.
- MonthDesc: Tahap ini menambahkan deskripsi nama bulan (Januari, Februari, Maret, dst.) berdasarkan kolom tanggal.
- MonthDesc Norm: Normalisasi nama bulan agar sesuai format standar.
- MonthDesc Gen: Tahap ini menghasilkan atau membuat data deskripsi nama bulan.
- Select values 2: Proses ini berfungsi untuk memilih atau menyaring kolom-kolom tertentu yang dibutuhkan dalam tahapan selanjutnya.
- Java Script Value: Pada langkah ini, digunakan skrip JavaScript untuk melakukan transformasi tambahan terhadap data.
- Select values: Tahap ini kembali menyaring kolom yang akan dimasukkan ke dalam tabel dim date (dimensi tanggal). Hanya kolom yang relevan yang akan dipilih.
- dim date: Ini adalah output akhir berupa tabel dimensi tanggal. Tabel ini berisi data tanggal, deskripsi hari, dan deskripsi bulan, yang siap digunakan untuk analisis data.



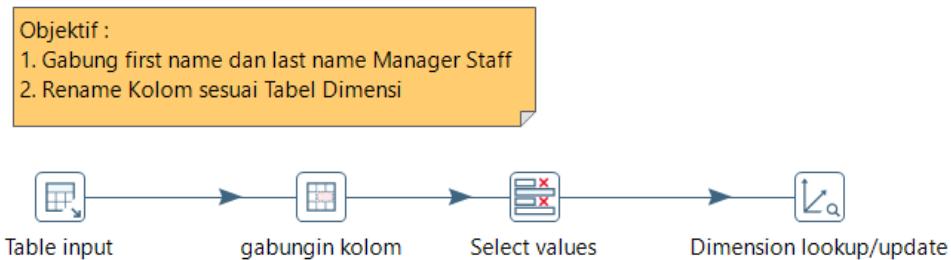
**Gambar 3.3.11. ETL Dimensi Staff**



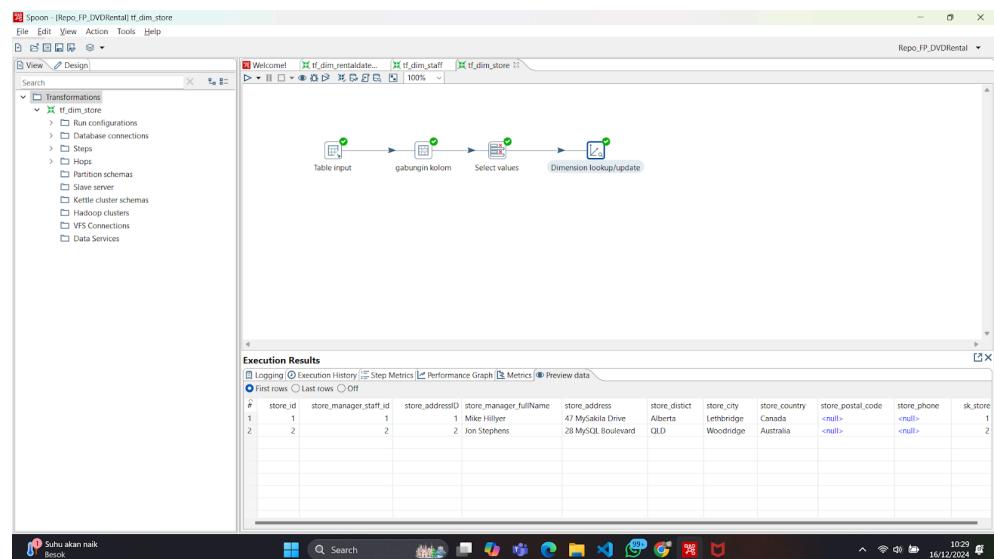
**Gambar 3.3.12.** Running Pentaho

- Mulailah dengan membuat koneksi ke database seperti langkah sebelumnya pada dimensi inventory. Langkah ini bertujuan untuk memastikan Pentaho dapat terhubung ke database target dengan konfigurasi yang benar, seperti nama host, port, nama database, username, dan password.
- Setelah koneksi berhasil, ambil data dari tabel staging staff sebagai sumber utama. Data ini akan diproses lebih lanjut untuk mendapatkan informasi lengkap tentang karyawan.
- Dilakukan join untuk Menggabungkan Informasi dari Tabel yang Berbeda untuk mendapatkan informasi lengkap tentang seorang karyawan, termasuk alamat lengkapnya (kota dan negara).
  - Tabel staff: Berisi informasi dasar tentang karyawan, seperti nama, jabatan, dan lainnya.
  - Tabel address: Berisi informasi alamat lengkap karyawan, termasuk nomor jalan atau lokasi spesifik lainnya.
  - Tabel city: Menyediakan informasi kota tempat karyawan berada.
  - Tabel country: Memberikan informasi negara tempat karyawan tinggal
- Gunakan *Concat Field* untuk menggabungkan kolom *first\_name* dan *last\_name* dari tabel staff menjadi satu kolom, misalnya dengan format "*FirstName LastName*". Hasil penggabungan ini akan menjadi nama lengkap karyawan dalam tabel dimensi.
- Gunakan *Select Values* untuk mengganti nama kolom sesuai dengan struktur tabel dimensi staff. Langkah ini juga memungkinkan untuk menghapus kolom yang tidak diperlukan, mengurutkan kolom, atau mengubah tipe data sesuai dengan kebutuhan tabel dimensi.
- Pada tahap akhir, gunakan step Dimension Lookup/Update untuk memasukkan data ke dalam tabel dimensi staff di database target. Pastikan step ini dikonfigurasi dengan benar agar data dimasukkan atau diperbarui sesuai kebutuhan.

- Sebelum menjalankan transformasi di Pentaho, eksekusi perintah SQL terlebih dahulu pada step terakhir, agar tabel masuk ke database terlebih dahulu.
- Setelah SQL dieksekusi, jalankan transformasi di Pentaho untuk memindahkan data dari staging staff ke tabel dimensi staff di database. Proses ini selesai jika data telah tersimpan di tabel target tanpa error.



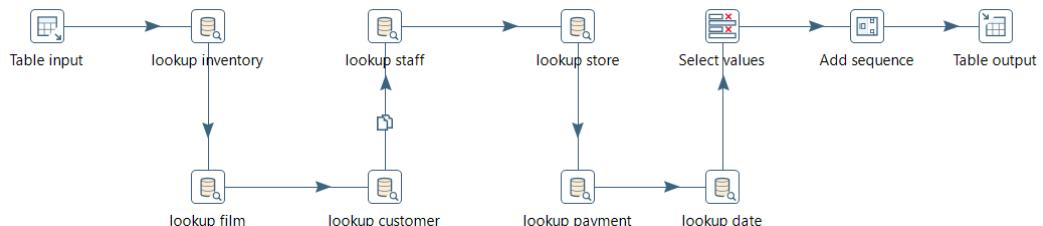
**Gambar 3.3.13 ETL Dimensi Store**



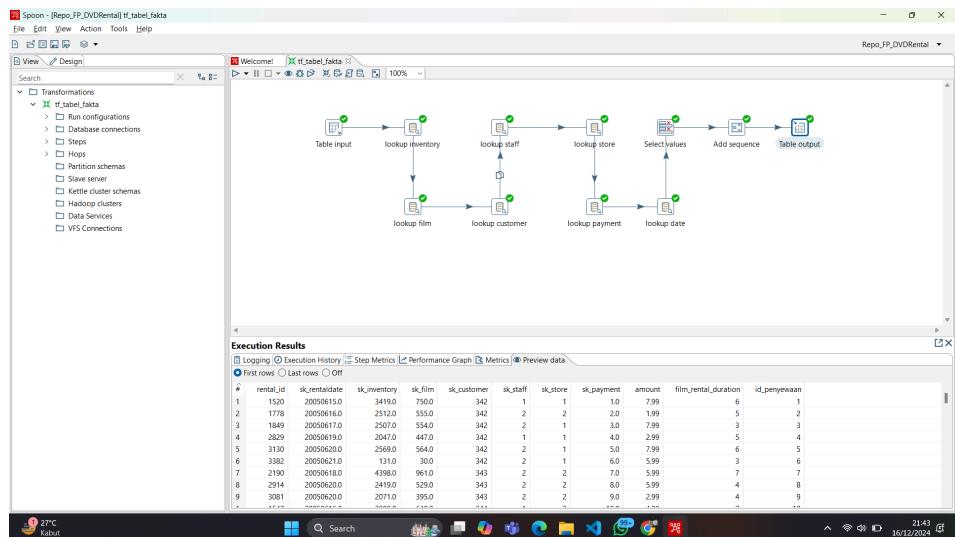
**Gambar 3.3.14. Running Pentaho**

- Langkah pertama adalah membuat koneksi ke database seperti yang dilakukan sebelumnya pada dimensi staff. Koneksi ini memastikan Pentaho terhubung dengan database target, sehingga data dari staging store dapat diakses dan diproses. Pastikan konfigurasi koneksi, seperti nama host, port, nama database, username, dan password, sudah benar agar koneksi berhasil.
- Setelah koneksi terhubung, ambil data dari tabel staging store menggunakan step input di Pentaho. Tabel staging store berfungsi sebagai sumber data utama untuk dimensi store. Data ini akan diolah lebih lanjut sesuai kebutuhan. Tambahkan query SQL ke dalam konfigurasi step untuk mengambil kolom yang diperlukan dari tabel staging store.

- Setelah konfigurasi step *input* selesai, lakukan preview data untuk memastikan data dari tabel staging store berhasil diambil. Jika preview menunjukkan data yang sesuai tanpa error, maka dapat melanjutkan ke langkah berikutnya.
- Gunakan step *Concat Field* untuk menggabungkan kolom *manager\_firstname* dan *manager\_lastname* menjadi satu kolom baru, misalnya *manager\_fullname*. Langkah ini bertujuan untuk menyederhanakan penyimpanan informasi nama lengkap manajer dalam tabel dimensi store.
- Selanjutnya, gunakan step *Select Values* untuk mengganti nama kolom sehingga sesuai dengan struktur tabel dimensi store. Selain mengganti nama kolom, langkah ini juga memungkinkan menghapus kolom yang tidak diperlukan atau menyesuaikan tipe data agar sesuai dengan desain tabel dimensi.
- Pada tahap terakhir, gunakan step *Dimension Lookup/Update* untuk memasukkan data yang telah diproses ke tabel dimensi store di database. Pastikan step ini dikonfigurasi untuk menambahkan atau memperbarui data ke tabel target dengan benar.
- Sebelum menjalankan transformasi di Pentaho, eksekusi perintah SQL terlebih dahulu pada step terakhir, agar tabel masuk ke database terlebih dahulu.
- Setelah perintah SQL selesai dieksekusi, jalankan transformasi di Pentaho untuk memindahkan data dari staging store ke tabel dimensi *store*. Jika semua langkah dilakukan dengan benar, data akan tersimpan di tabel dimensi store tanpa error, dan proses selesai dengan sukses.



**Gambar 3.3.15. ETL Fact Tabel**

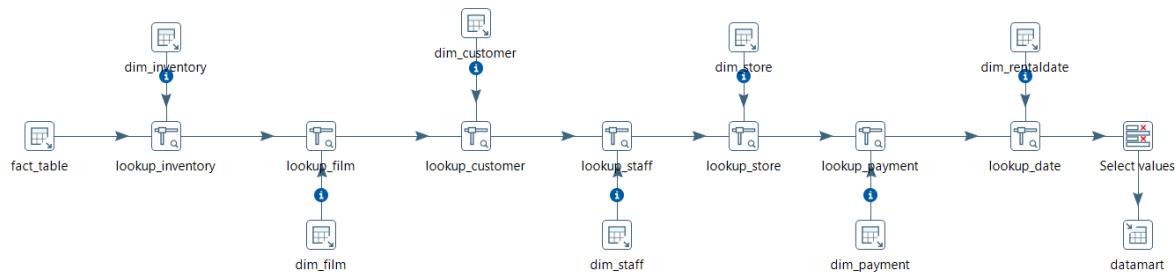


Gambar 3.3.16. Running Pentaho

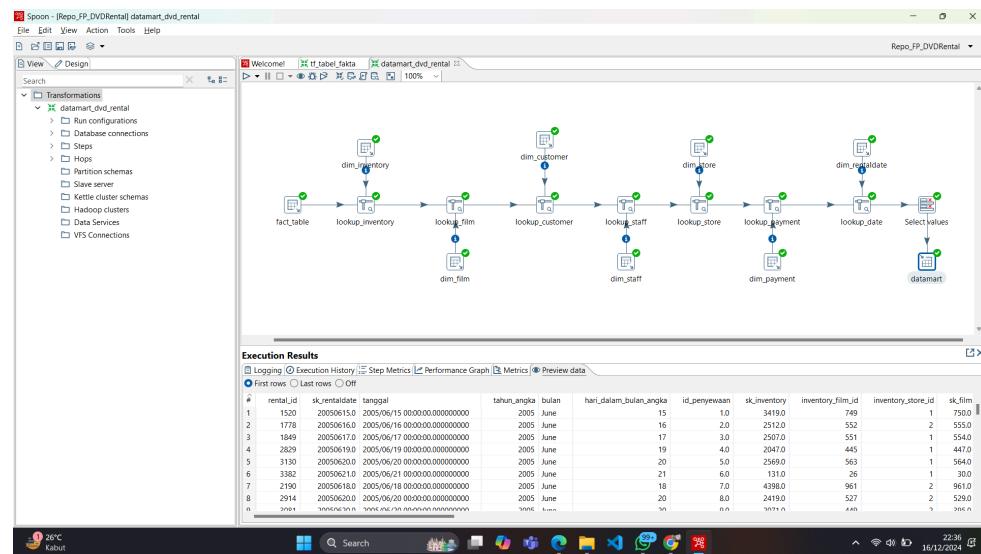
- Buat connection dulu seperti step sebelumnya yg sama seperti pembuatan dimensi
- Dalam proses ETL ini, setiap DB Lookup harus dilakukan secara berurutan karena setiap langkah lookup bergantung pada hasil dari langkah sebelumnya. Misalnya, lookup inventory dilakukan terlebih dahulu untuk mendapatkan inventory\_id, yang kemudian digunakan dalam lookup film untuk mengambil informasi terkait film. Selanjutnya, lookup seperti store dan staff bergantung pada data kunci dari lookup sebelumnya. Jika urutannya tidak diikuti, kolom kunci yang dibutuhkan untuk proses lookup selanjutnya tidak akan tersedia, sehingga menyebabkan kegagalan join atau data yang tidak lengkap. Oleh karena itu, menjaga urutan dalam DB Lookup sangat penting agar aliran data berjalan dengan benar dan menghasilkan output yang akurat.
  - Lookup Inventory → Mendapatkan inventory\_id.
  - Lookup Film → Mengambil informasi terkait film.
  - Lookup Store dan Staff → Berdasarkan kunci sebelumnya.
- Selanjutnya ambil data rental dari staging payment untuk di tabel input seperti gambar dibawah ini.
- Lakukan Preview untuk memastikan data berhasil diambil.
- Selanjutnya lookup database nya dengan menggunakan database lookup
- Lakukan lookup database hingga dimensi terakhir caranya sama dengan me lookup database seperti sebelumnya.
- Setelah proses lookup selesai, kita perlu **memilih kolom-kolom** yang relevan untuk dimuat ke dalam tabel output. Tahap ini memastikan hanya kolom yang diperlukan saja yang akan disertakan.
- Tahap ini berguna jika kita perlu menyusun data atau memberikan **identitas unik** pada setiap transaksi.

- Pada step terakhir ini yaitu untuk menambahkan data pada dbeaver nya dengan menggunakan step dimensi lookup update. Eksekusi SQL terlebih dahulu sebelum menjalankan Pentaho.

### 3.4 Pembuatan Datamart



**Gambar 3.4.1. ETL Datamart**



**Gambar 3.4.2. Running Pentaho**

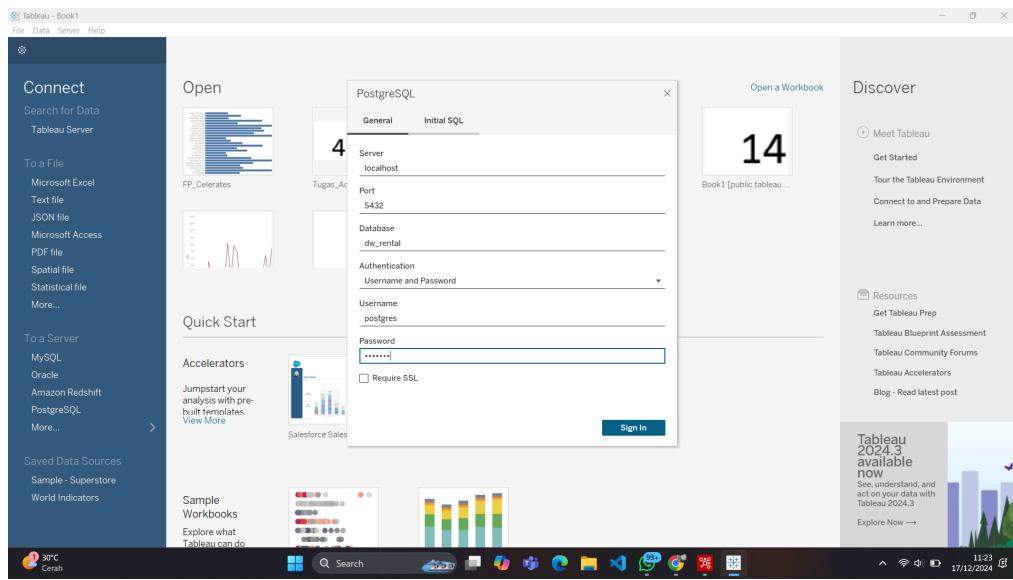
- Buat schema baru dengan nama **datamart** pada database dw\_rental, untuk membedakan hasil data warehouse dan staging.
- Buat ETL datamart pada pentaho dengan tutorial berikut :
  - Drag Table Input ke canvas dan buat koneksi untuk menghubungkan ke database.
  - Setelah sudah membuat connection sambungkan target schema nya dengan table fakta yg telah dibuat pada data warehouse.

- Pada proses **lookup inventory**, data dari fact\_table dicocokkan dengan tabel dimensi **dim\_inventory**. Tujuan dari proses ini adalah untuk melengkapi informasi terkait inventaris, seperti detail stok atau ID inventaris yang berhubungan dengan transaksi. Ini membantu dalam memahami barang atau item yang terlibat dalam transaksi.
- Setelah mendapatkan informasi dari inventory, data selanjutnya diperkaya melalui lookup ke tabel **dim\_film**. Proses ini menambahkan informasi detail mengenai film, seperti judul film, kategori, atau durasi film.
- Pada tahap lookup customer, data transaksi dihubungkan dengan tabel **dim\_customer** untuk melengkapi informasi pelanggan. Informasi ini mencakup nama pelanggan, alamat, dan detail kontak lainnya. Tujuan dari proses ini adalah memahami siapa pelanggan yang terlibat dalam transaksi.
- Proses lookup staff menghubungkan data transaksi dengan tabel **dim\_staff** untuk mendapatkan informasi terkait staf. Data seperti ID staf atau nama staf yang bertanggung jawab dalam transaksi ditambahkan di sini. Ini berguna untuk melacak kinerja staf dalam proses transaksi.
- Data kemudian melalui proses lookup store untuk mencocokkan informasi dari tabel **dim\_store**. Proses ini menambahkan detail lokasi toko, seperti nama toko atau alamat toko, di mana transaksi terjadi. Hal ini membantu dalam analisis penjualan berdasarkan wilayah atau lokasi.
- Pada tahap lookup payment, data transaksi dihubungkan dengan tabel **dim\_payment** untuk mendapatkan informasi pembayaran. Detail seperti metode durasi rental atau jumlah pembayaran dilengkapi di sini. Proses ini penting untuk analisis keuangan dan pendapatan.
- Tahap terakhir adalah lookup date, di mana data dihubungkan dengan tabel **dim\_rentaldate**. Proses ini menambahkan informasi waktu terkait transaksi, seperti tanggal sewa. Hal ini krusial untuk analisis berbasis waktu.
- Setelah semua proses lookup selesai, kolom-kolom yang relevan dipilih dalam Select Values. Proses ini memastikan hanya atribut penting yang dimasukkan ke dalam datamart.
- Pada step terakhir ini yaitu untuk menambahkan data padadbeaver nya dengan menggunakan step tabel output. Eksekusi SQL terlebih dahulu sebelum menjalankan Pentaho. Kali ini sambungkan schemanya dengan schema yg sudah dibuat tadi yaitu schema datamart.

### **3.5 Visualisasi dan Analisis Dashboard**

Jelaskan proses pembuatan dashboard, insight yang dihasilkan, dan bagaimana dashboard tersebut menjawab kebutuhan bisnis.

#### **1. Buka Tableau dan Sambungkan ke PostgreSQL**

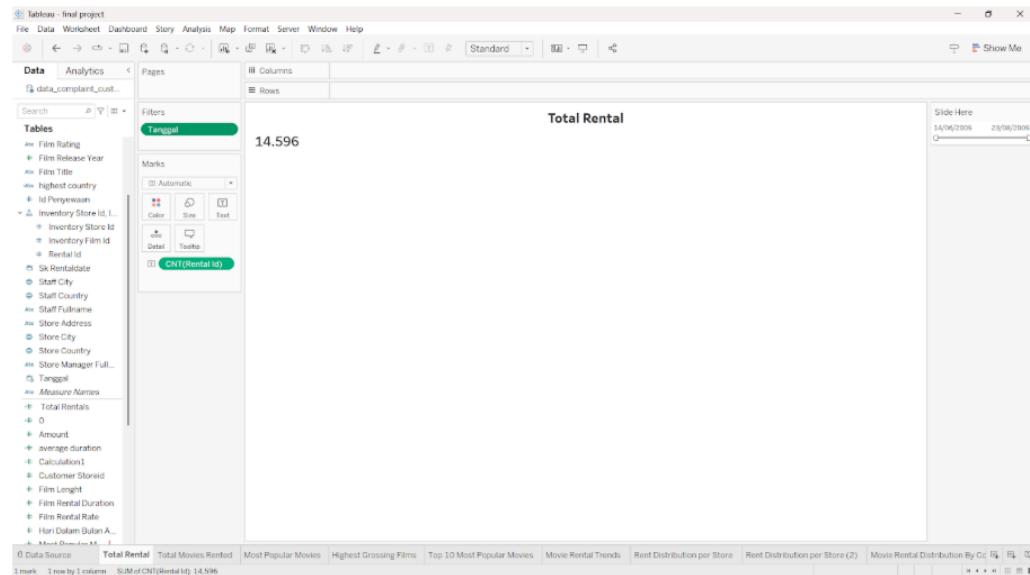


*Gambar 3.5.1.1. Tampilan awal Tableau*

## 2. Lakukan visualisasi dengan menggunakan sheet.

- **Total Rental**

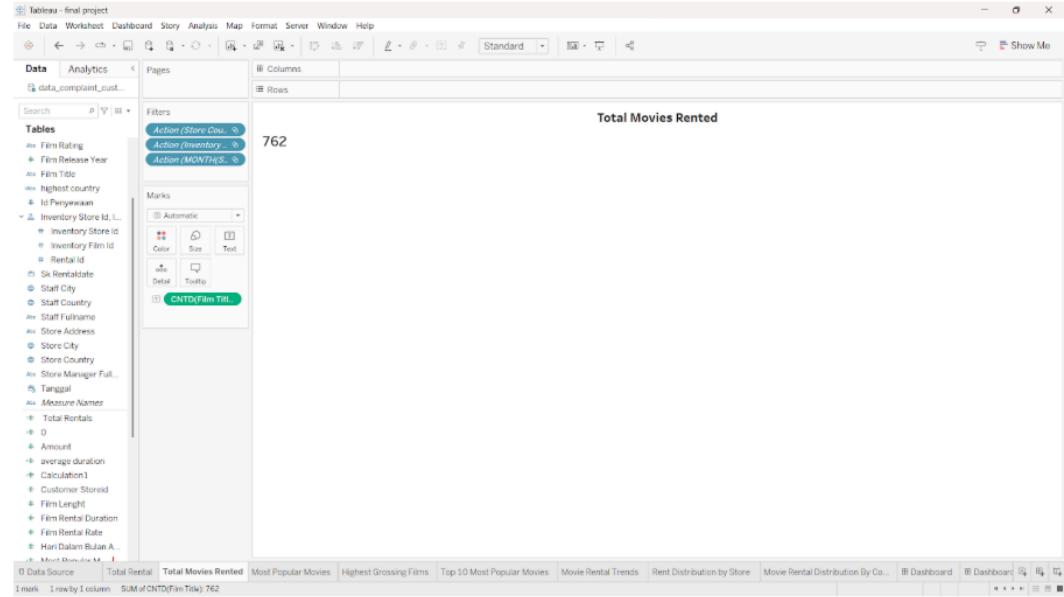
Buat worksheet baru bernama total rental untuk menunjukkan jumlah total penyewaan film di database, kemudian letakkan CNT(Rental Id) sebagai text untuk ditampilkan dan Tanggal sebagai filter.



*Gambar 3.5.2.1. Visualisasi Total Rental*

- **Total Movies Rented**

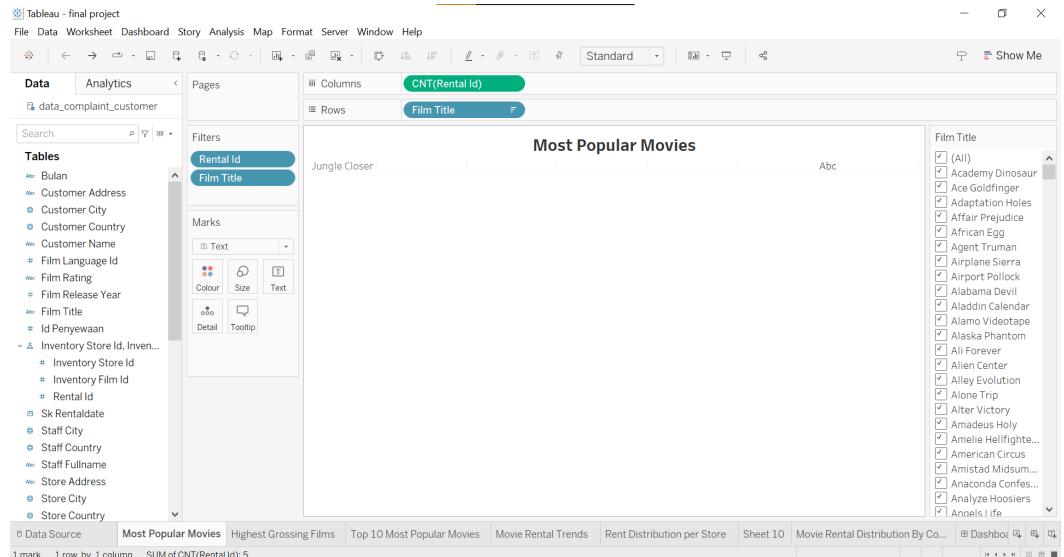
Buat worksheet baru bernama Total Movies Rented untuk melihat total penyewaan film. Letakkan CNTD (Film Titles) sebagai Text untuk ditampilkan. Kemudian Filters action yang mencakup Store Country, Inventory, dan Months akan muncul setelah penggerjaan dashboard.



**Gambar 3.5.2.2. Visualisasi Total Movies Rented**

### • Most Popular Movies

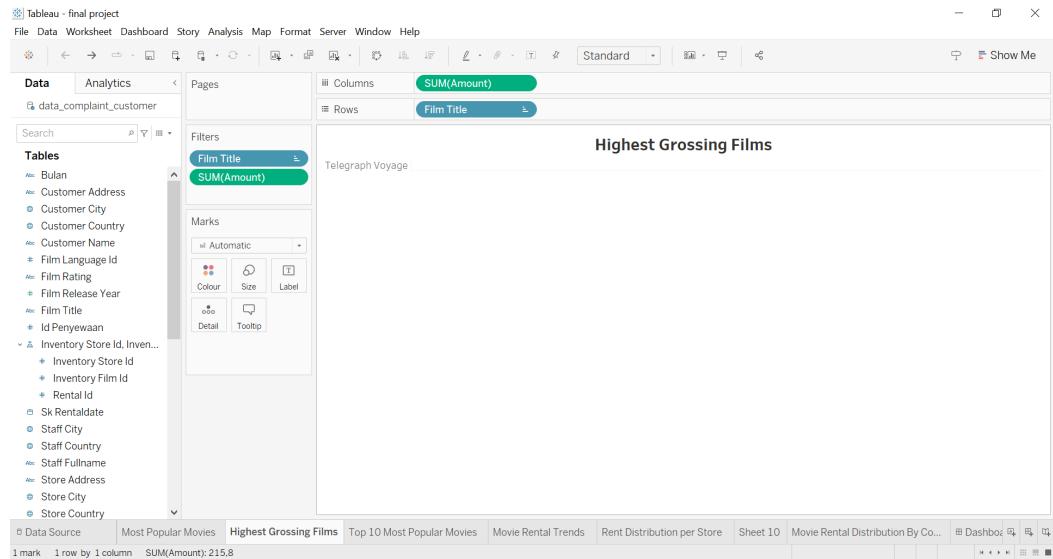
Buat worksheet baru bernama Most Popular Movies untuk menampilkan film dengan jumlah penyewaan tertinggi. Letakkan Film title sebagai Row dan CNT (Rental Id) sebagai Column. Posisikan juga Rental Id dan Film Title sebagai Filters.



**Gambar 3.5.2.3. Visualisasi Most Popular Film**

- **Highest Grossing Films**

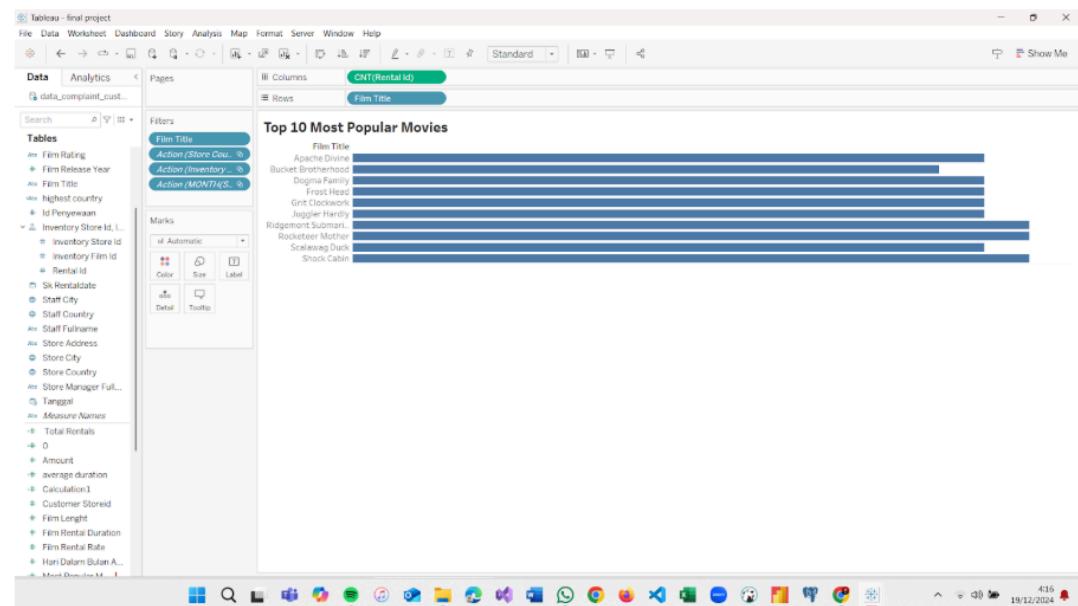
Buat worksheet baru bernama Highest Grossing Films untuk menampilkan film dengan pendapatan tertinggi berdasarkan data penyewaan. Letakkan Film title sebagai Row dan SUM (Amount) sebagai Column. Posisikan juga SUM (Amount) dan Film Title sebagai Filters.



**Gambar 3.5.2.4. Visualisasi Highest Grossing Film**

- **Top 10 Most Popular Movies**

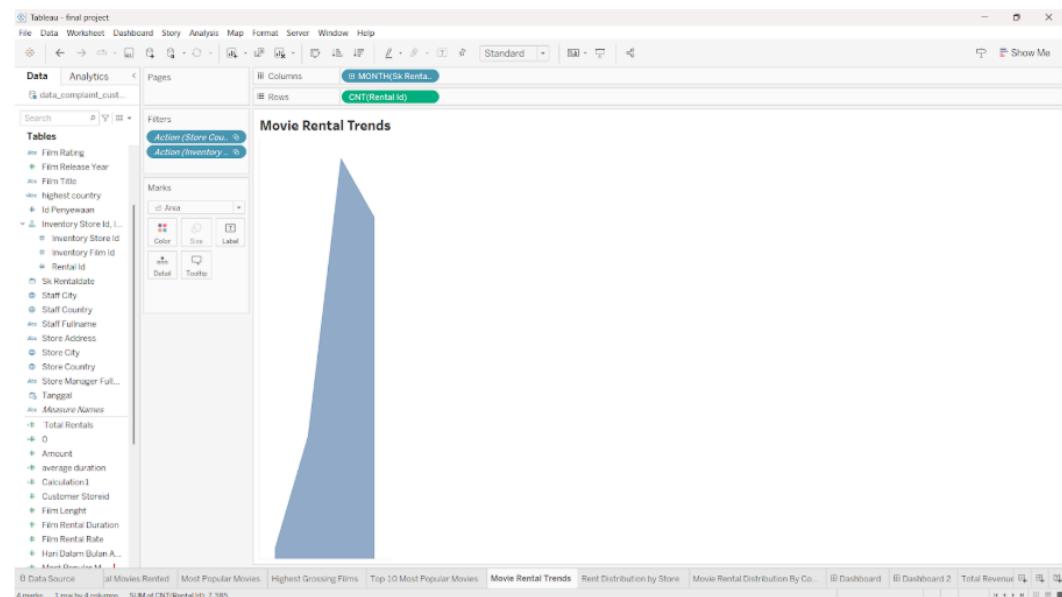
Buat worksheet baru untuk menampilkan 10 besar film dengan jumlah penyewaan tertinggi yang ditunjukkan dengan bar chart. Letakkan Film Title sebagai row dan CNT (Rental Id) sebagai column. Posisikan juga Film Title sebagai filter. Setelah penggeraan dashboard, maka filter actions yang mencakup store country, inventory, dan months akan muncul.



**Gambar 3.5.2.5.** Visualisasi Top 10 Most Popular Movies

### • Movie Rental Trends

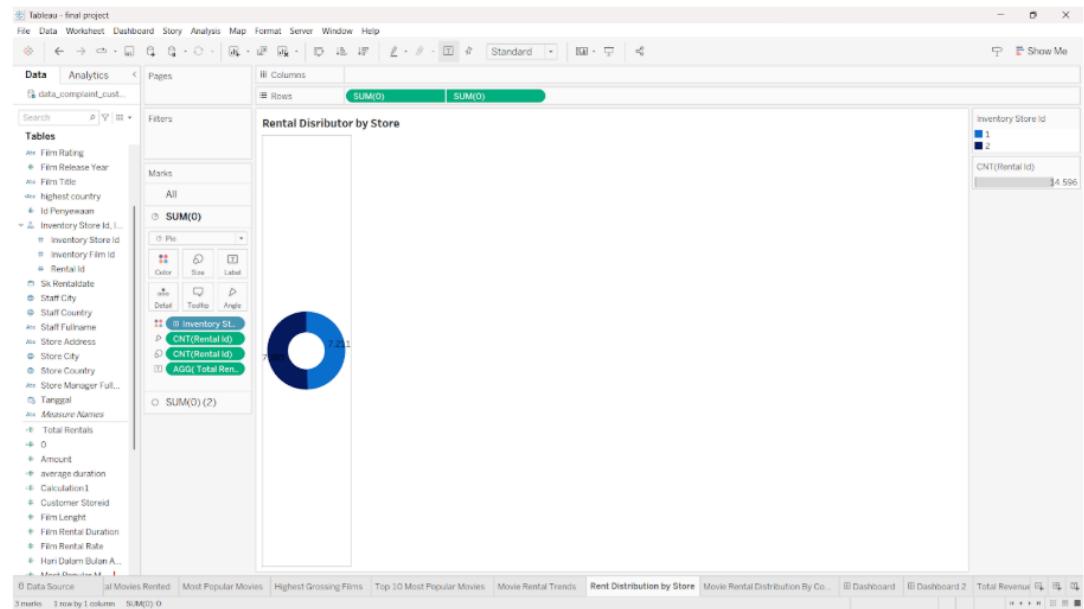
Buat worksheet baru yang bernama Movie Rental Trends untuk melacak tren jumlah penyewaan film dari waktu ke waktu yang ditunjukkan dengan line chart. Letakkan Rental ID dengan cara di Count pada Rows dan Bulan menggunakan SK Rentaldate pada Column.



**Gambar 3.5.2.6.** Visualisasi Movie Rental Trends

### • Rental Distribution by Store

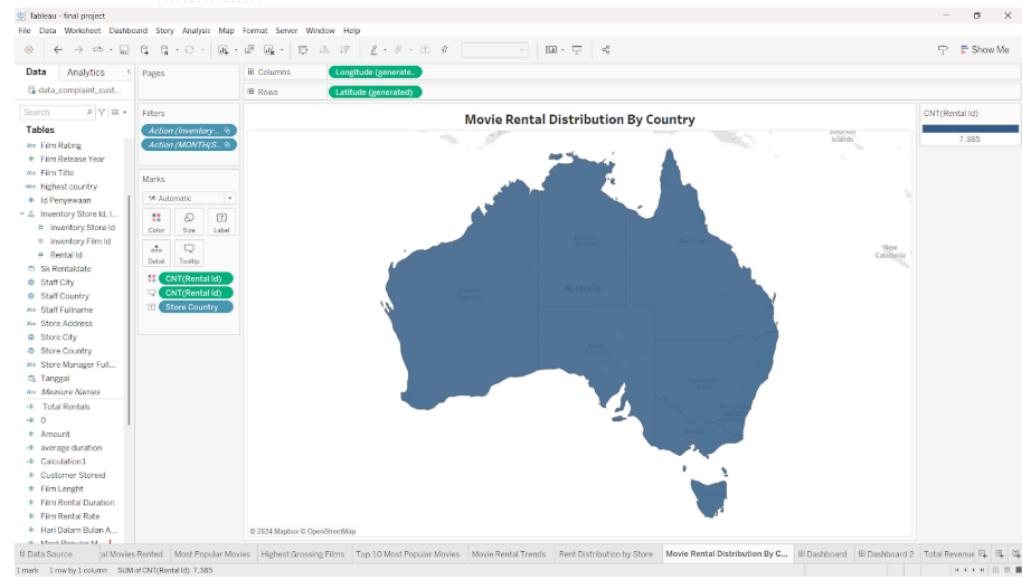
Buat worksheet baru dengan nama Rental Distribution by Store untuk menampilkan distribusi penyewaan berdasarkan toko dan ditunjukkan dengan donut chart. Dengan menggunakan **donut chart**, visualisasi ini memberikan gambaran proporsi kontribusi masing-masing toko terhadap keseluruhan penyewaan. Bagian-bagian dalam **donut chart** mewakili toko-toko, dengan ukuran segmen yang proporsional terhadap jumlah penyewaan yang dilakukan di setiap toko.



**Gambar 3.5.2.7. Visualisasi Rental Distribution by Store**

### • Movie Rental Distribution by Country

Buat worksheet baru bernama Movie Rental Distribution by Country untuk menunjukkan distribusi penyewaan film berdasarkan negara yang ditunjukkan dengan maps. Letakkan CNT (Rental Id) dalam warna dan tooltip serta store country sebagai teks. Setelah penggerjaan dashboard, maka filter actions yang mencakup inventory, dan months akan muncul.

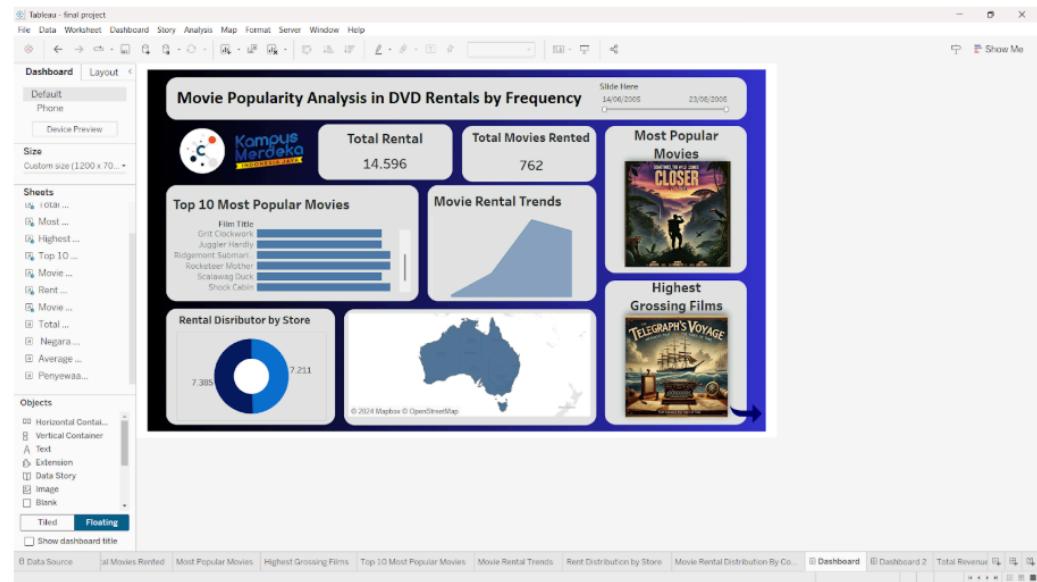


**Gambar 3.5.2.8.** Visualisasi Movie Rental Distribution by Country

### 3. Membuat Dashboard

Setelah visualisasi pada worksheets dilakukan, maka akan dibuat dashboard berdasarkan sheet-sheet yang telah dibuat. Pada dashboard 1 yang merupakan dashboard utama di bawah ini terdapat berbagai chart yang dibuat pada worksheets sebelumnya. Dashboard ini dilengkapi dengan title “Movie Popularity Analysis in DVD Rentals by Frequency” sesuai dengan studi kasus kelompok kami. Terdapat juga filter date untuk mencari data tanggal yang lebih spesifik. Terdapat pula visual logo-logo penting yang terkait dengan project ini. Charts yang tersedia untuk ditampilkan adalah Total Rental, Total Movies Rented, Top 10 Most Popular Movies, Movie Rental Trends, Rental Distributor by Store, dan Map untuk letak Geografis. Terdapat pula charts Most Popular Movies dan Highest Grossing Film yang dilengkapi dengan visual poster film agar dashboard terlihat lebih menarik. Icon navigation dihadirkan untuk beralih menuju dashboard 2.

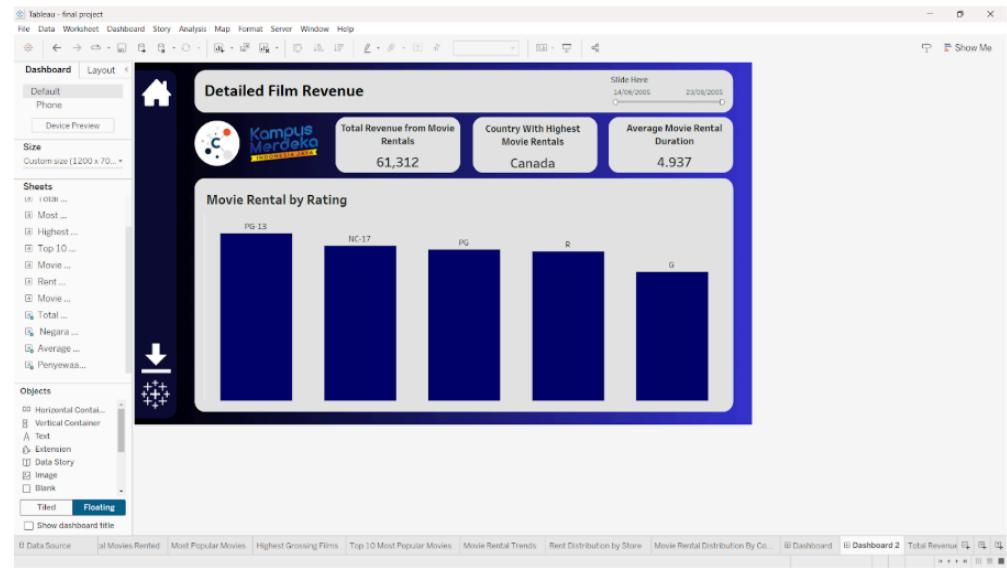
- **Dashboard 1**



**Gambar 3.5.3.1. Tampilan dashboard 1**

- **Dashboard 2**

Dashboard 2 berfungsi sebagai pusat informasi detail tentang film. Dashboard ini menyajikan data seperti total revenue dari penyewaan film, negara dengan jumlah penyewaan tertinggi, rata-rata durasi film, serta chart yang menunjukkan jumlah penyewaan film berdasarkan rating tertinggi. Dashboard ini juga dilengkapi dengan slider untuk memilih rentang tanggal. Selain itu, terdapat icon home yang mengarahkan pengguna ke dashboard utama, icon unduhan yang memungkinkan pengguna mengunduh dashboard dalam format PDF, dan icon Tableau yang mengarahkan ke Tableau Public untuk akses lebih lanjut.



*Gambar 3.5.3.1. Tampilan dashboard 2*

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Proyek ini berhasil memberikan wawasan strategis yang bermanfaat bagi bisnis penyewaan DVD untuk beradaptasi dengan tantangan yang dihadapi akibat perubahan teknologi dan persaingan dengan layanan streaming digital. Melalui analisis data frekuensi penyewaan, proyek ini mampu mengidentifikasi tingkat popularitas film, kategori yang paling diminati, serta pola musiman yang mempengaruhi permintaan pelanggan.

Secara teknis, proyek ini menghasilkan solusi berbasis data yang dapat digunakan untuk optimalisasi inventaris, peningkatan efisiensi operasional, dan pengembangan strategi pemasaran. Informasi terkait film-film yang paling sering disewa membantu pelaku usaha mengelola stok secara lebih efisien sehingga dapat memenuhi permintaan pelanggan dengan lebih baik. Analisis preferensi pelanggan memungkinkan pengelolaan bisnis yang lebih terukur, baik dari sisi logistik maupun strategi pemasaran, serta menjadi dasar untuk merancang promosi yang relevan dan program loyalitas pelanggan yang lebih efektif.

Dari sudut pandang bisnis, manfaat yang diraih meliputi peningkatan daya saing melalui pengambilan keputusan berbasis data, peningkatan kepuasan pelanggan, serta potensi pendapatan yang lebih besar dengan menyesuaikan layanan sesuai preferensi pasar. Dengan pendekatan yang terukur ini, proyek menunjukkan bahwa penerapan analisis data dalam bisnis penyewaan DVD mampu memperkuat daya saing di era digital, sekaligus memberikan nilai tambah dalam pengalaman pelanggan. Strategi ini tidak hanya relevan untuk masa kini, tetapi juga dapat menjadi pijakan untuk pengembangan inovasi bisnis di masa depan.

#### **4.2 Saran**

Untuk pengembangan lebih lanjut, berikut beberapa rekomendasi dari sisi teknis dan penerapan analisis data:

1. Pengembangan Sistem Prediksi Permintaan: Menggunakan teknologi machine learning untuk memprediksi permintaan film berdasarkan tren penyewaan sebelumnya, pola musiman, dan faktor eksternal seperti perilisan film baru atau liburan.

2. Segmentasi Pelanggan: Melakukan analisis segmentasi pelanggan dan memberikan rekomendasi film yang disesuaikan dengan preferensi masing-masing pelanggan atau menawarkan diskon untuk kategori film tertentu kepada segmen yang relevan.
3. Optimasi Program Loyalitas: Mengembangkan program loyalitas berbasis data, seperti poin yang dapat ditukar dengan penyewaan gratis atau akses awal ke koleksi film terbaru, untuk meningkatkan retensi pelanggan. Data historis dapat digunakan untuk merancang program yang memberikan insentif sesuai dengan pola konsumsi pelanggan.
4. Pemantauan Kompetitor dan Tren Pasar: Menggunakan data eksternal untuk memantau langkah kompetitor dan tren dalam industri hiburan, seperti genre yang sedang populer atau strategi pemasaran yang berhasil di pasar streaming. Analisis kompetitor ini dapat memberikan insight untuk inovasi bisnis dan meningkatkan daya saing.
5. Evaluasi Berkelanjutan: Menerapkan proses evaluasi berkelanjutan dengan membandingkan kinerja sebelum dan sesudah penerapan strategi berbasis data. Melalui evaluasi ini, bisnis dapat mengidentifikasi area yang perlu diperbaiki, mengukur keberhasilan strategi yang sudah dijalankan, dan mengarahkan fokus pengembangan di masa mendatang.

## DAFTAR PUSTAKA

PostgreSQL Global Development Group. PostgreSQL: The world's most advanced open source database. PostgreSQL Global Development Group. Retrieved from <https://www.postgresql.org>

DBeaver Community. DBeaver: Universal database tool. DBeaver Community. Retrieved from <https://dbeaver.io>

Hitachi Vantara. Pentaho Data Integration. Hitachi Vantara. Retrieved from <https://www.pentaho.com>

Tableau Software. Tableau: Visual analytics platform. Salesforce. Retrieved from <https://www.tableau.com>

Bharadwaj, A., El Sawy, O. A., Pavlou, P. A., & Venkatraman, N. (2013). Digital business strategy: Toward a next generation of insights. *MIS Quarterly*, 37(2), 471–482. Retrieved from [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2742300](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2742300)

Rumahweb. *DBeaver adalah? Pengertian, fitur, dan cara install DBeaver*. Retrieved December 19, 2024, from <https://www.rumahweb.com/journal/dbeaver-adalah/>

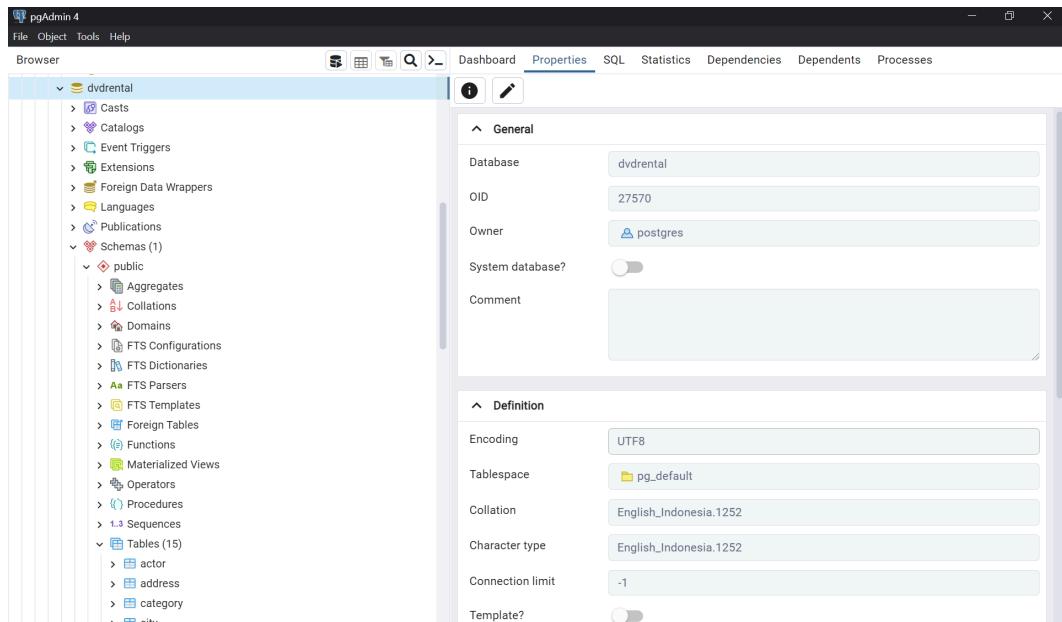
S. M. H. & S., A. (2020). *Analisis dan Implementasi Model Data Mining dalam Sistem Pendukung Keputusan*. *Transformatika*, Volume(Issue), pages. Retrieved December 19, 2024, from <https://journals.usm.ac.id/index.php/transformatika/article/viewFile/62/62>

Yudi, W. (2015). *Modul praktikum Pentaho Kettle*. Retrieved December 19, 2024, [http://file.upi.edu/Direktori/FPMIPA/PRODI\\_ILMU\\_KOMPUTER/Yudi%20Wibisono/data\\_mining/Modul\\_Praktikum\\_Pentaho\\_Kettle.pdf](http://file.upi.edu/Direktori/FPMIPA/PRODI_ILMU_KOMPUTER/Yudi%20Wibisono/data_mining/Modul_Praktikum_Pentaho_Kettle.pdf)

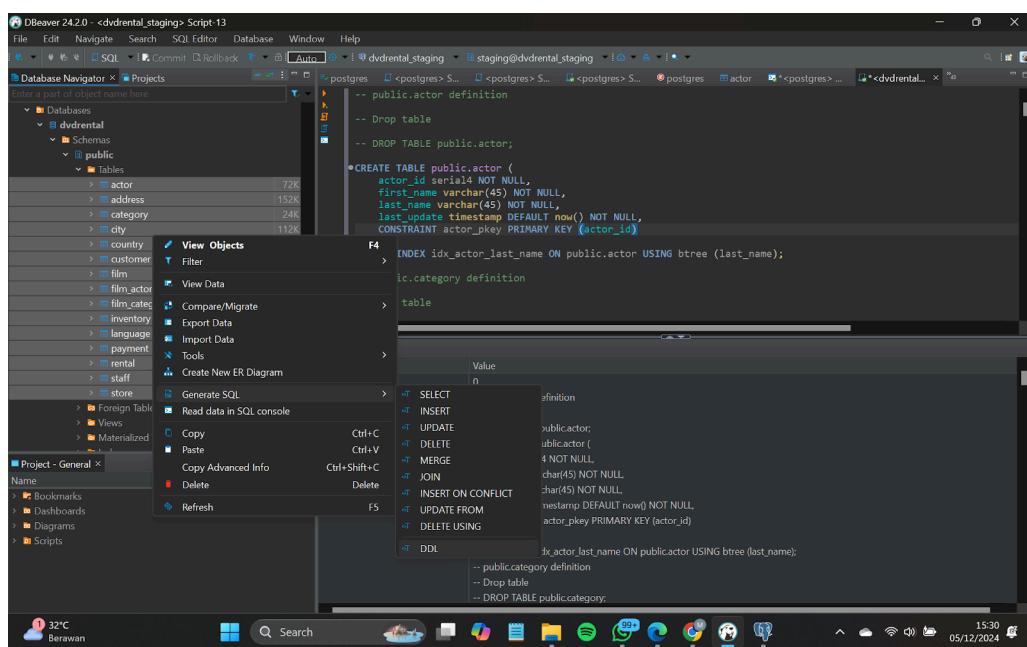
## LAMPIRAN

Berikut adalah dokumentasi berupa tangkapan layar dari langkah-langkah penggerjaan kelompok kami:

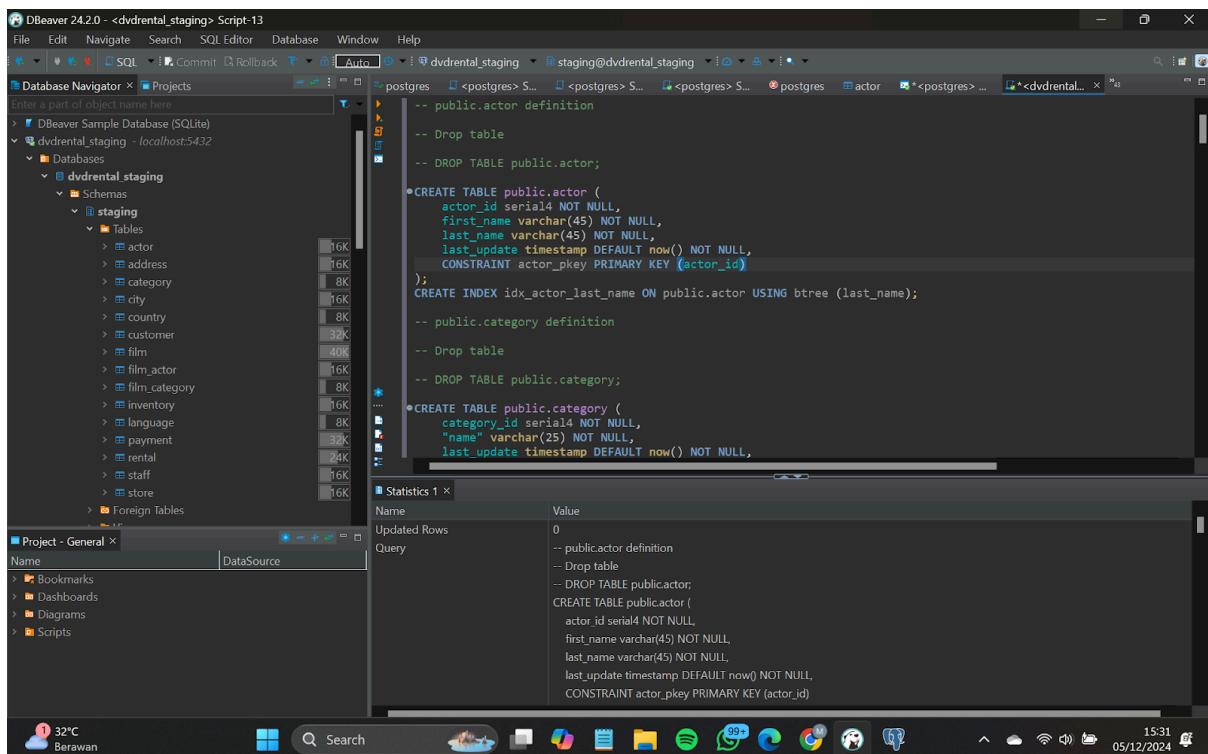
### 1. Proses Staging Area



*Gambar 1.1. Restore database*



**Gambar 1.2. Copy tabel di database**



DBeaver 24.2.0 - <dvrental\_staging> Script-13

File Edit Navigate Search SQL Editor Database Window Help

Database Navigator Projects

Enter a part of object name here

Databases dvrental\_staging - localhost:5432

Schemas dvrental\_staging

Tables staging

- actor (16K)
- address (16K)
- category (8K)
- city (16K)
- country (8K)
- customer (32K)
- film (40K)
- film\_actor (16K)
- film\_category (8K)
- inventory (16K)
- language (8K)
- payment (32K)
- rental (24K)
- staff (16K)
- store (16K)

Foreign Tables

Project - General

Name	DataSource
Query	

Statistics 1

Name	Value
Updated Rows	0
Query	-- public.actor definition -- Drop table -- DROP TABLE public.actor; CREATE TABLE public.actor ( actor_id serial4 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); -- public.category definition -- Drop table -- DROP TABLE public.category; CREATE TABLE public.category ( category_id serial4 NOT NULL, "name" varchar(25) NOT NULL, last_update timestamp DEFAULT now() NOT NULL,

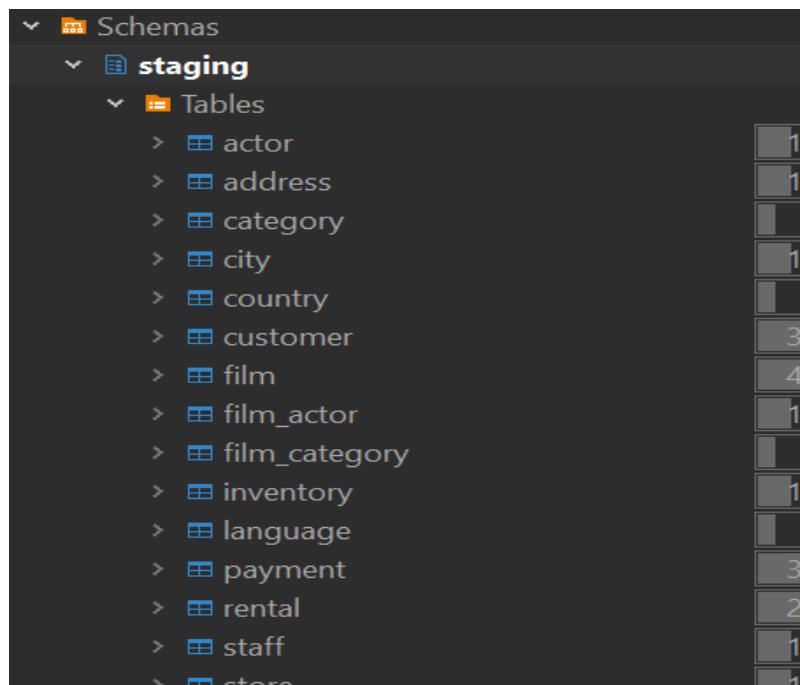
32°C Berawan

Search

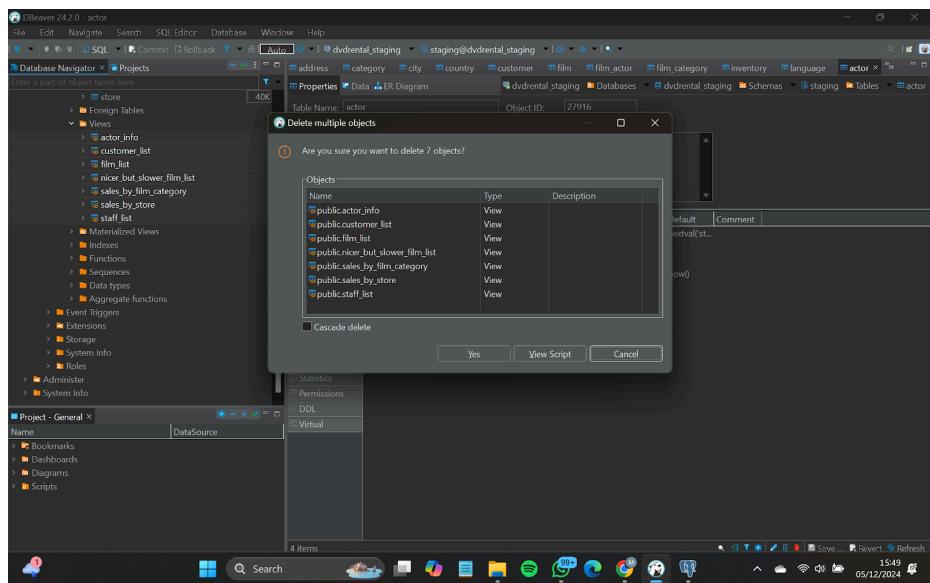
15:31 05/12/2024

This screenshot shows the DBeaver interface. The top navigation bar includes File, Edit, Navigate, Search, SQL Editor, Database, Window, and Help. The Database Navigator pane on the left lists databases, schemas, and tables. The SQL Editor pane at the bottom contains a script for creating the 'actor' table in the 'public' schema. The Project - General pane shows a single entry for 'Query'. The Statistics 1 pane displays the creation script for the 'actor' table.

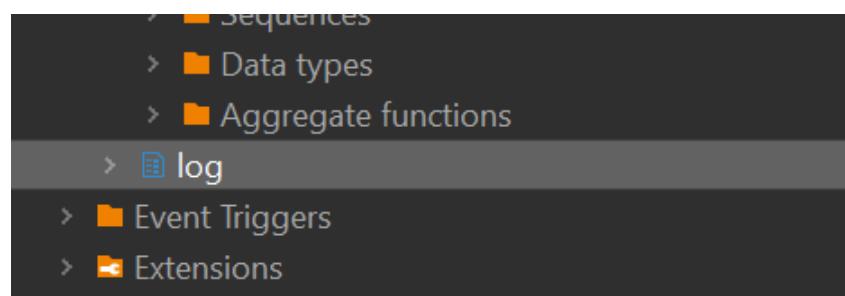
**Gambar 1.3. Pembuatan Struktur Tabel**



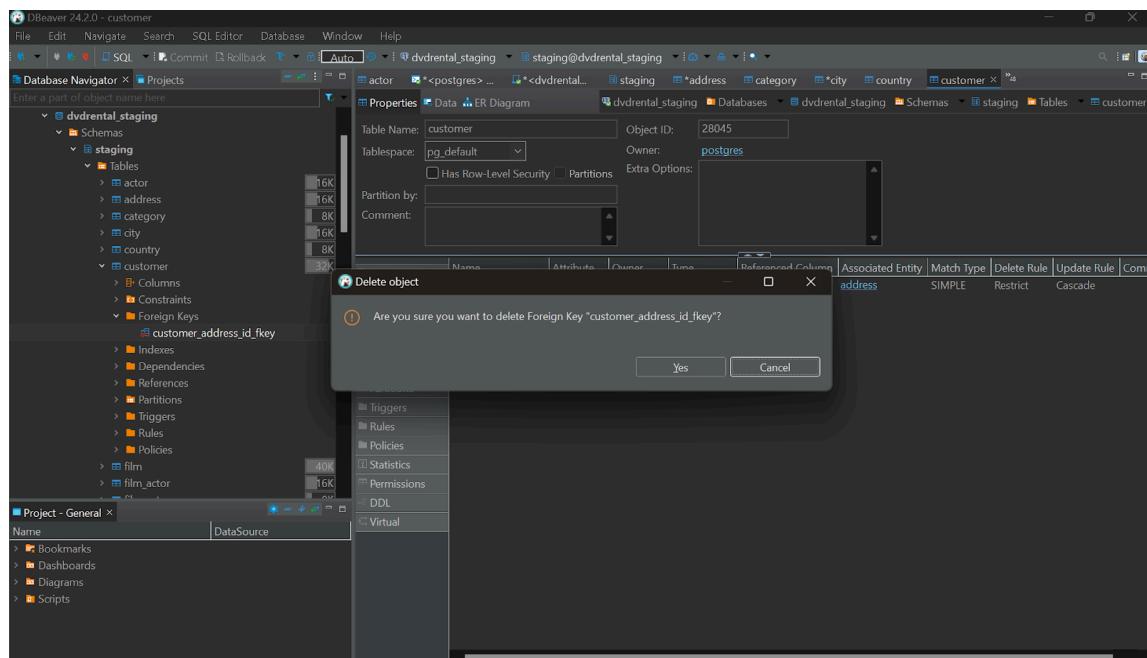
**Gambar 1.4. Rename nama schema**



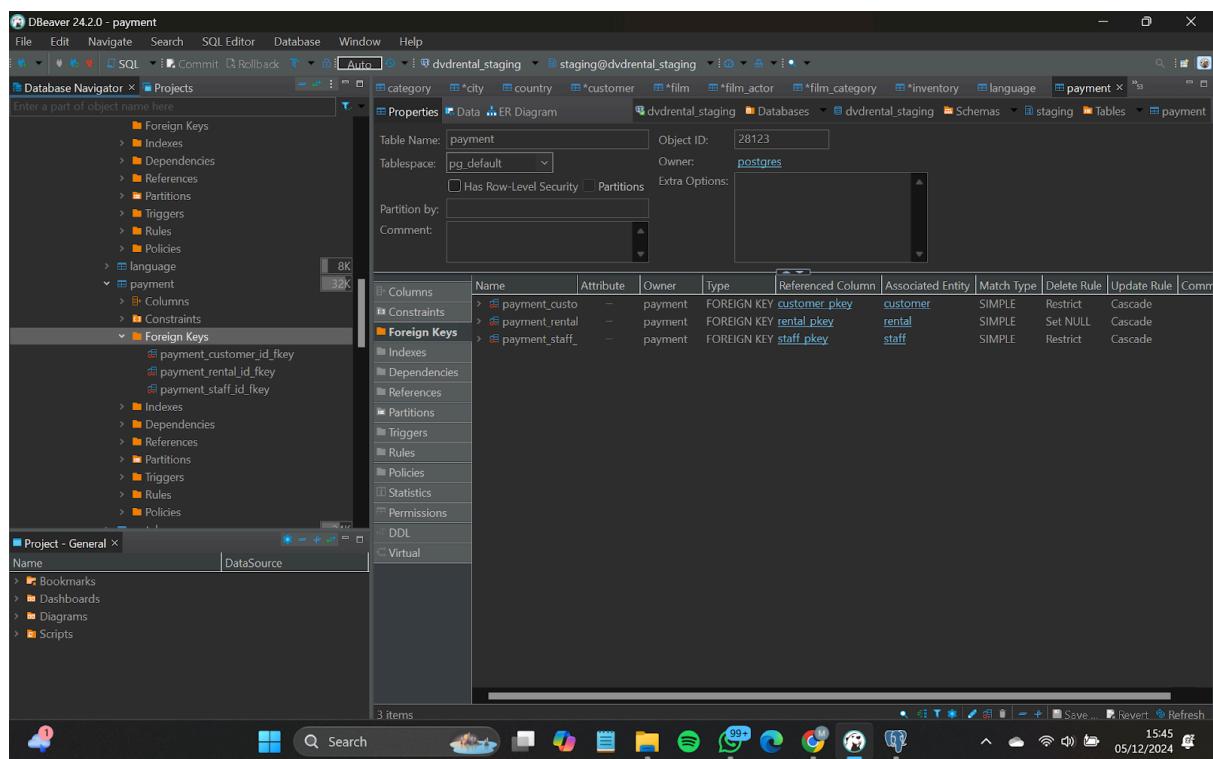
**Gambar 1.5. Menghapus View yang Tidak Diperlukan**



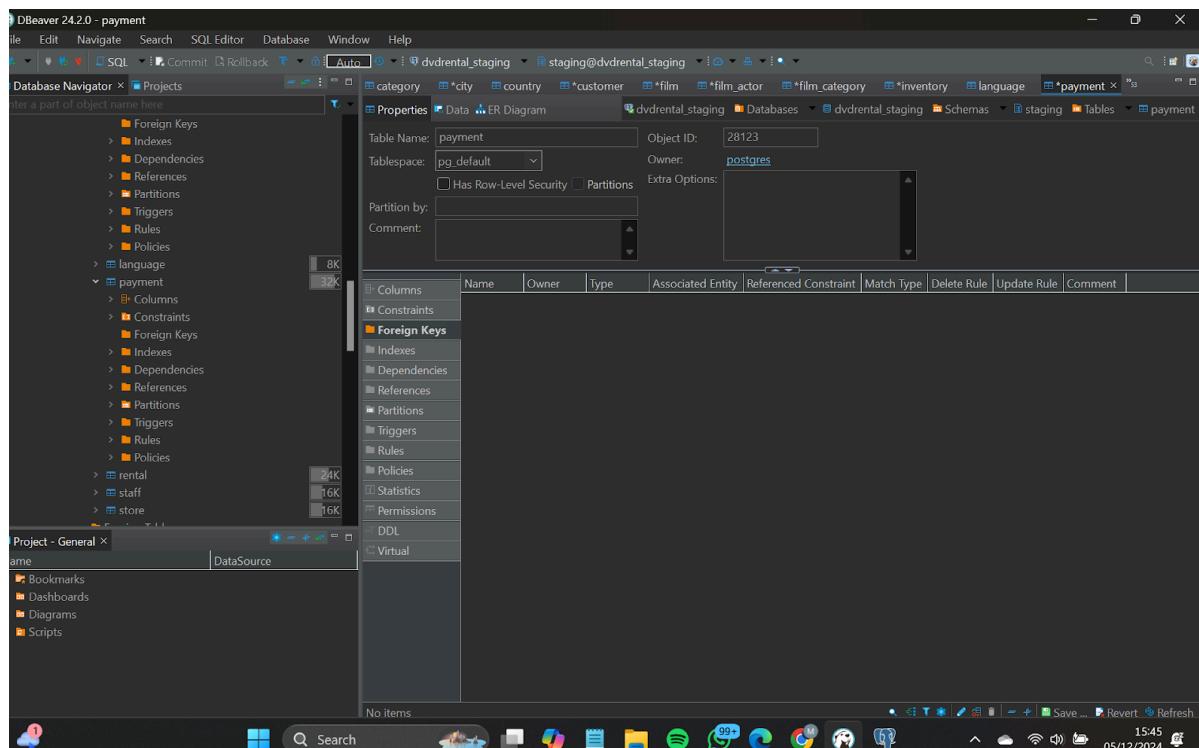
**Gambar 1.6. Membuat Schema Baru untuk Log**



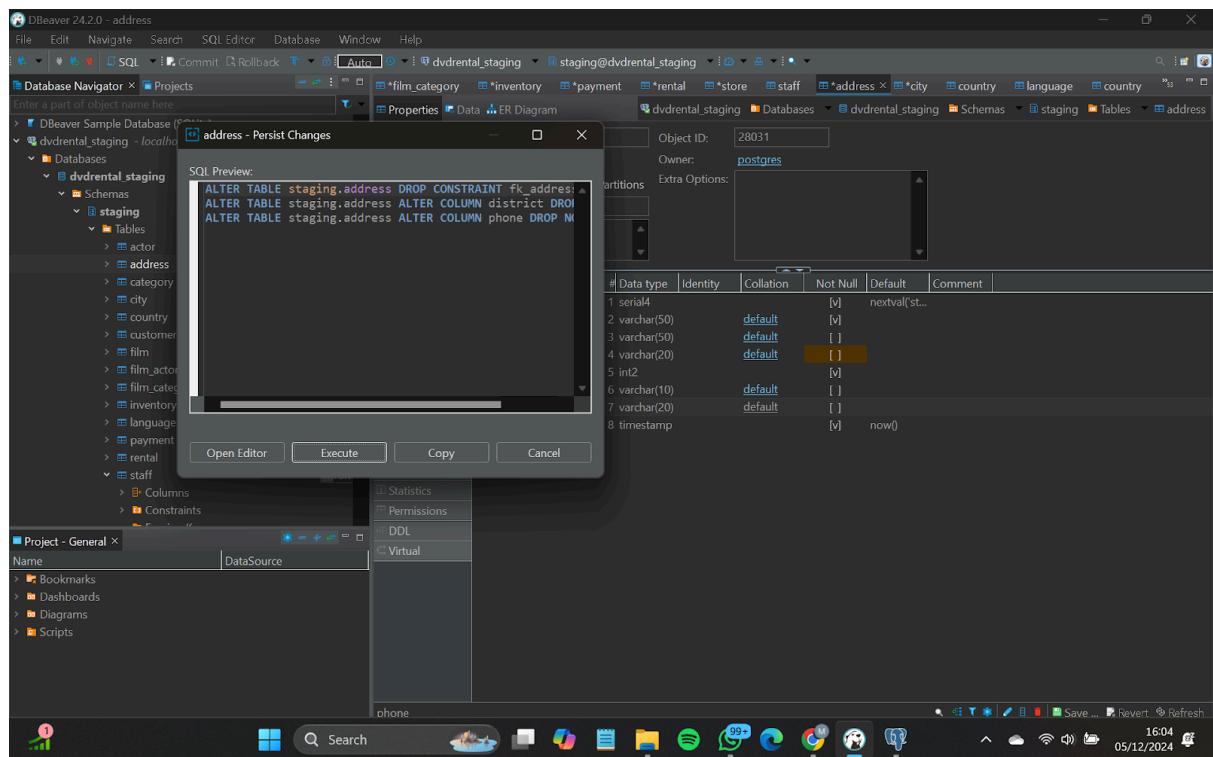
*Gambar 1.7. Hapus foreign key Setiap Tabel Schema*



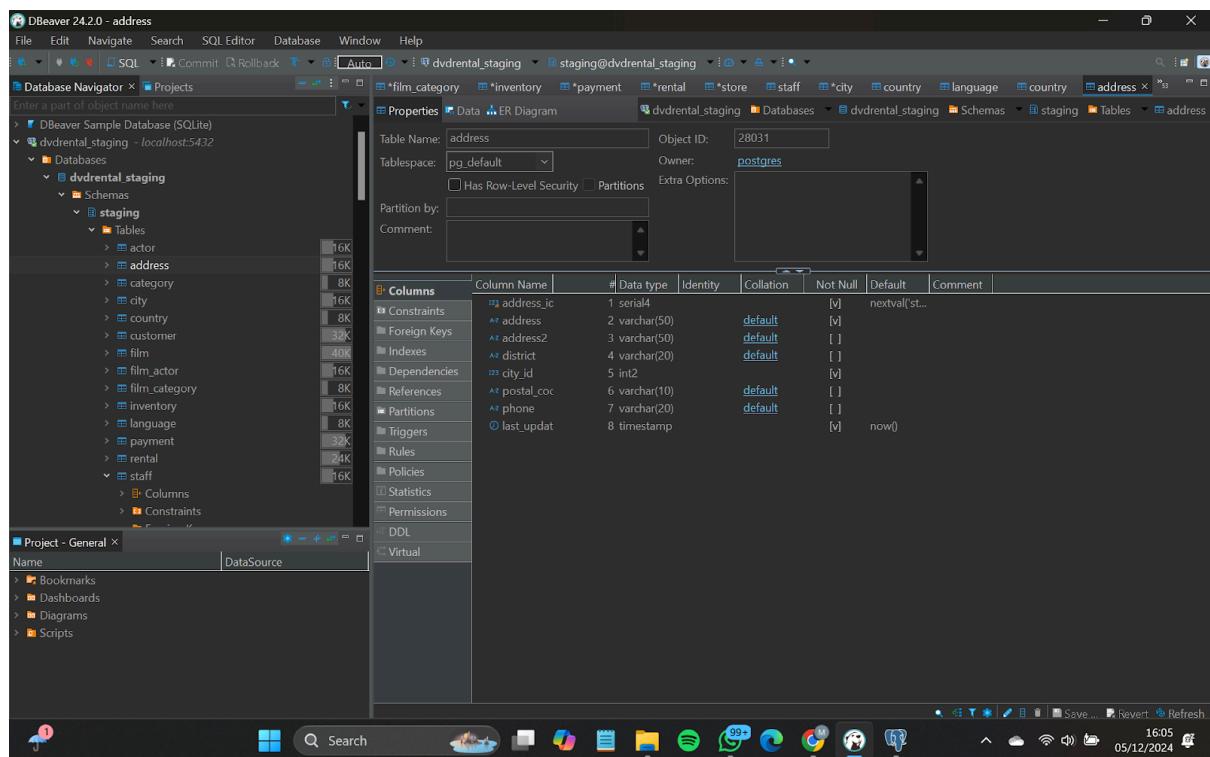
**Gambar 1.8.** Hapus foreign key Setiap Tabel Schema



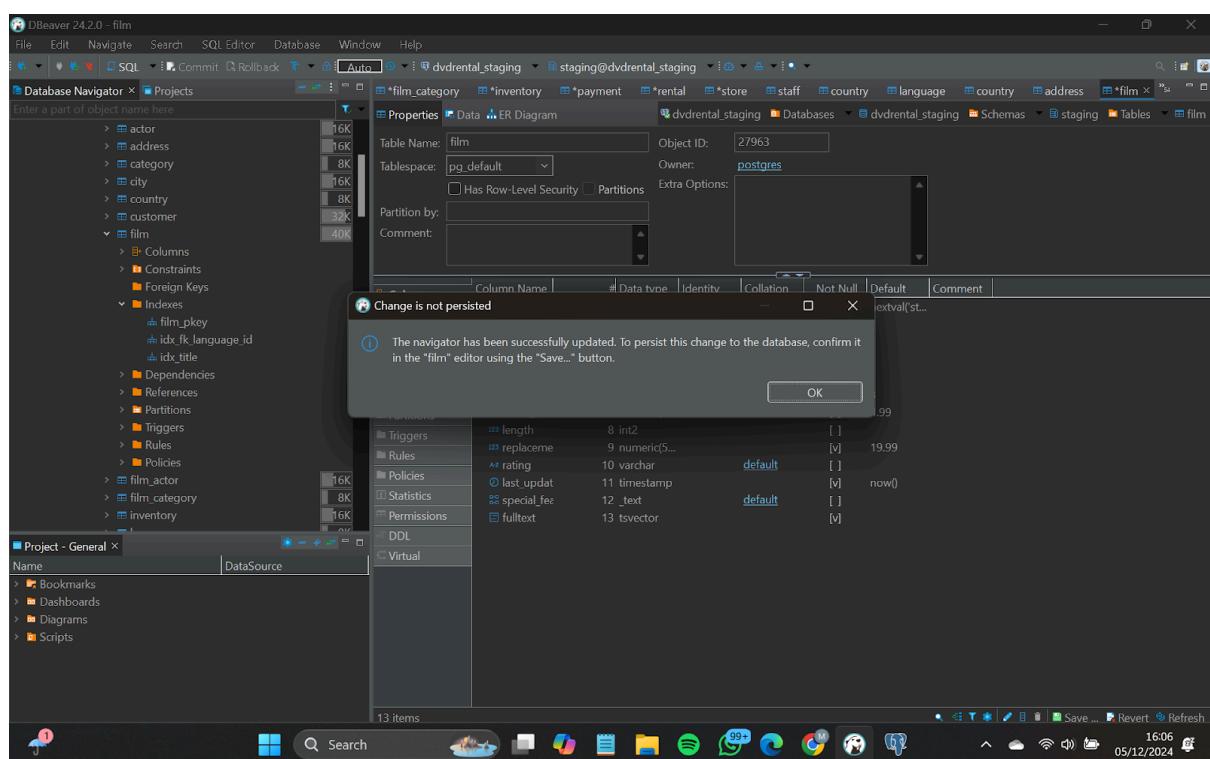
**Gambar 1.9.** Hasil hapus foreign key Setiap Tabel Schema



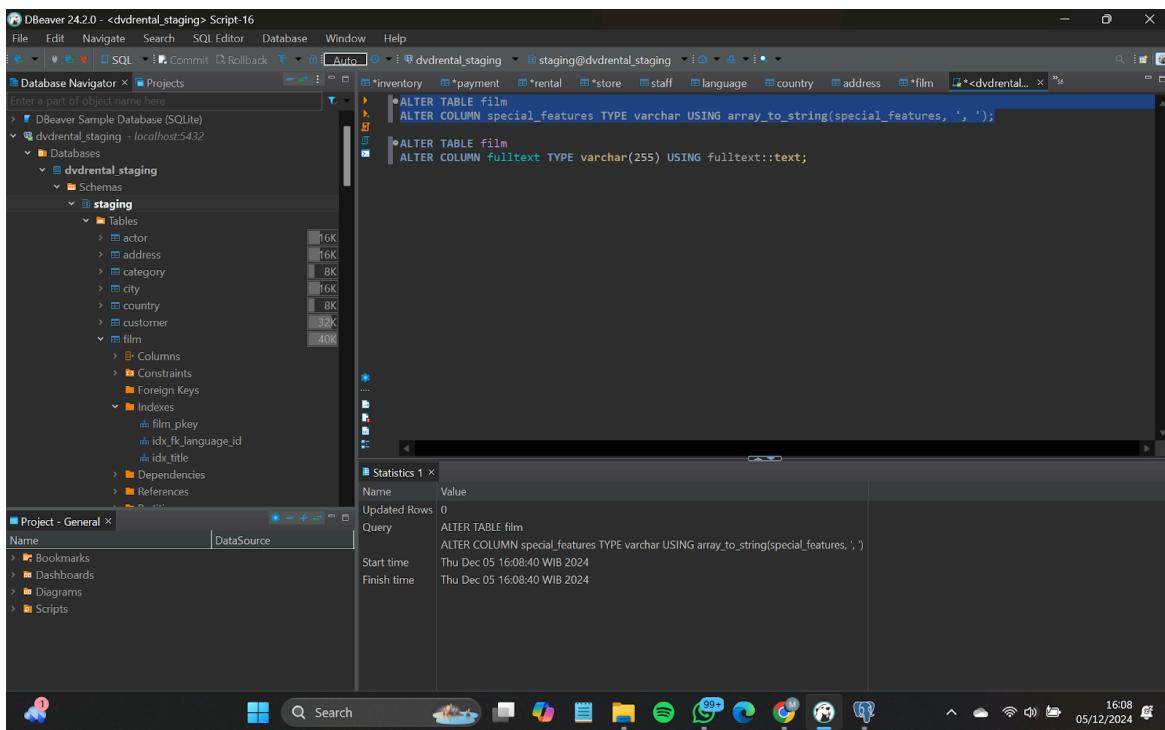
**Gambar 1.10.** Ubah Properti Kolom Menjadi Nullable di Tabel Address



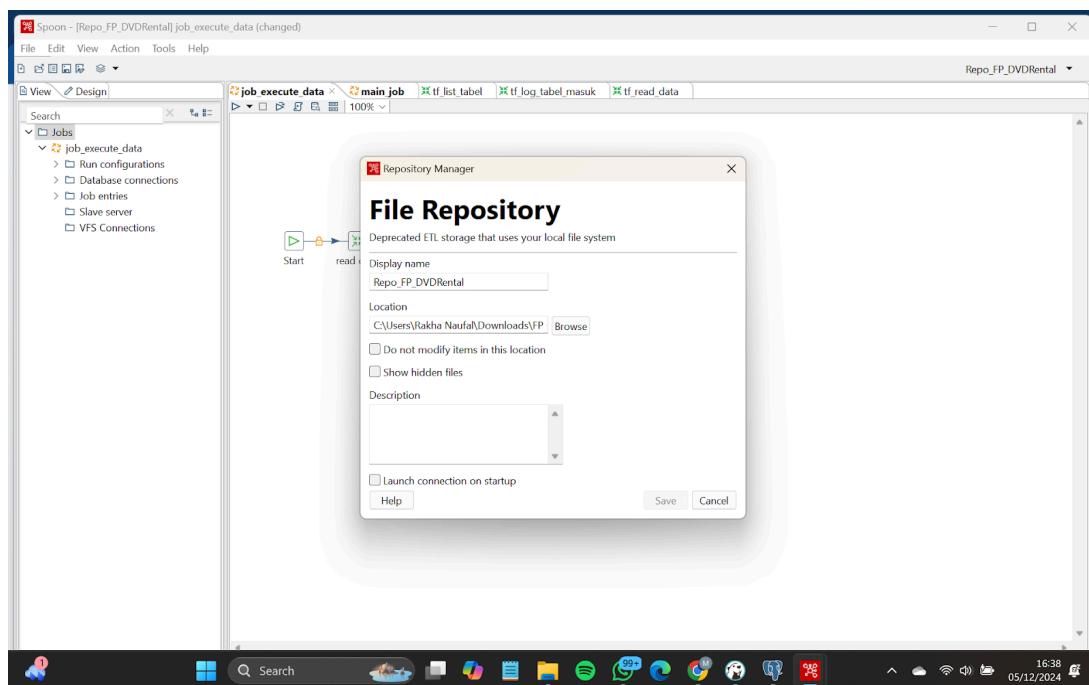
**Gambar 1.11. Hasil Mengubah Kolom di Tabel Address**



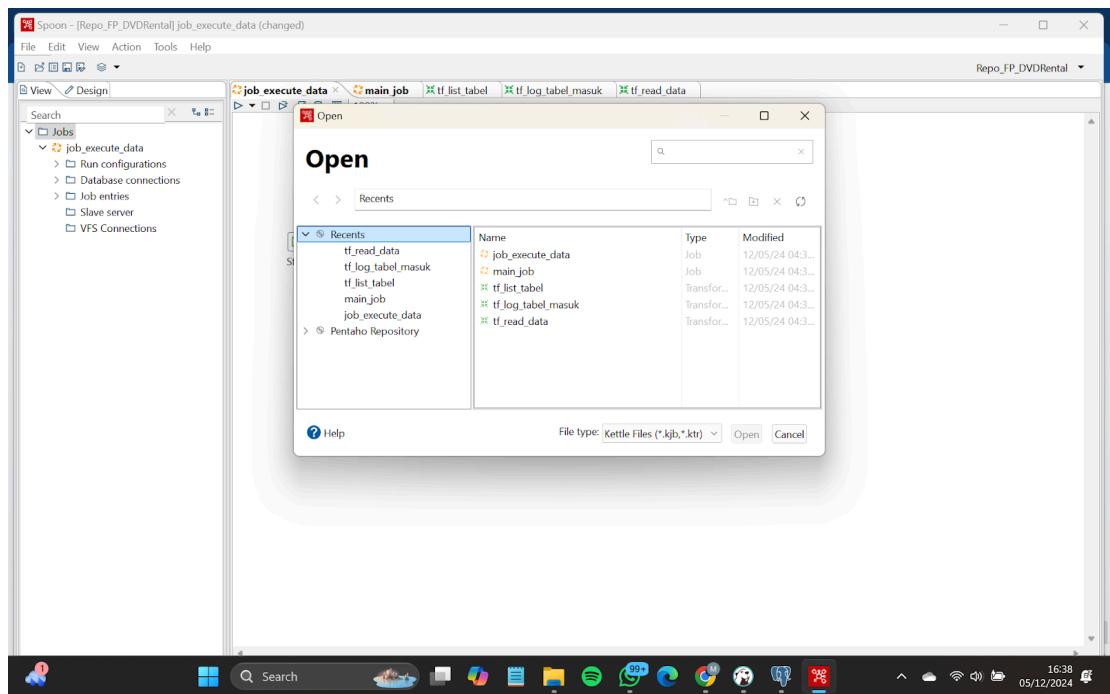
**Gambar 1.12.** Hapus Indeks film\_fulltext.idx pada Tabel Staging



**Gambar 1.13.** Mengubah Data Type pada Tabel Film



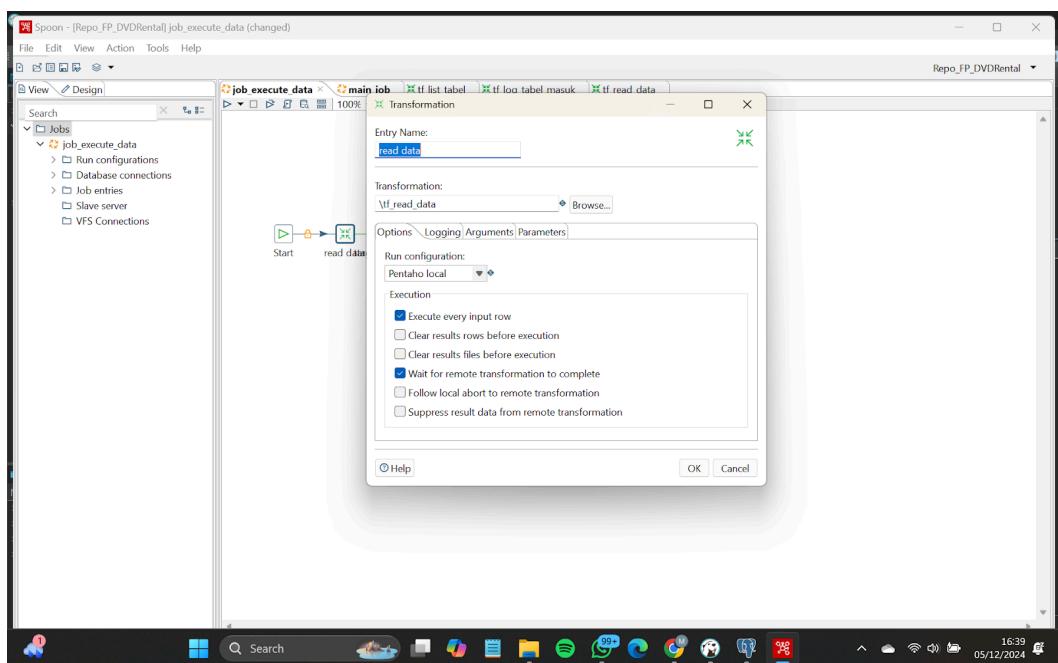
**Gambar 1.14.** Membuat Repository Baru Di Pentaho



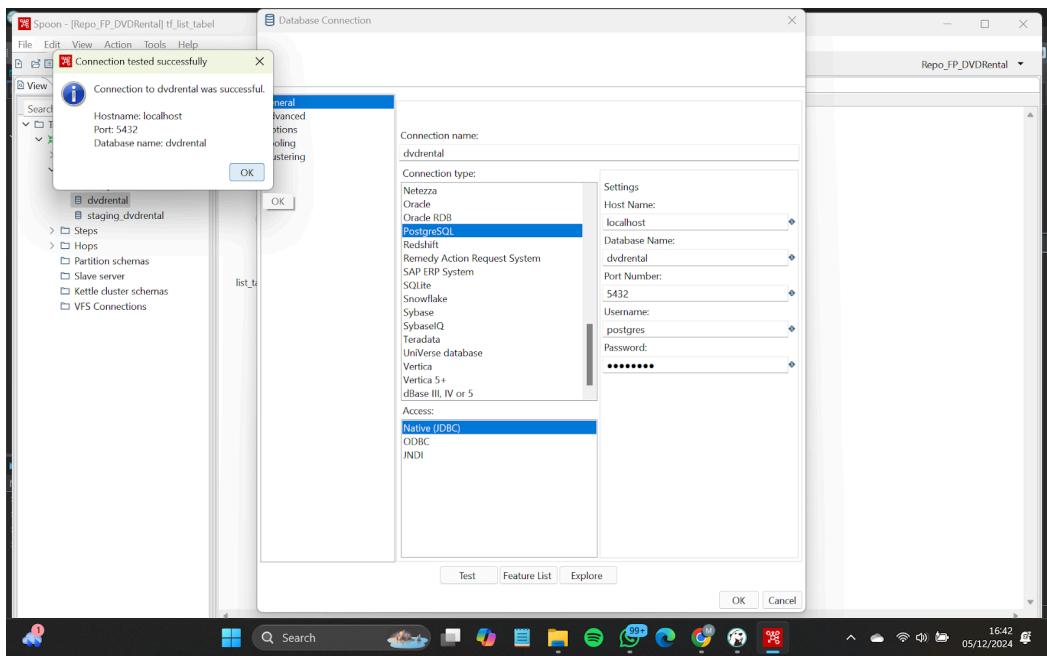
Name	Type	Modified
tf_read_data	Job	12/05/24 04:3...
tf_log_tabel_masuk	Job	12/05/24 04:3...
tf_list_tabel	Transformation	12/05/24 04:3...
main.job	Transformation	12/05/24 04:3...
job_execute_data	Transformation	12/05/24 04:3...
Pentaho Repository		

File type: Kettle Files (\*.kjb,\*ktr) Open Cancel

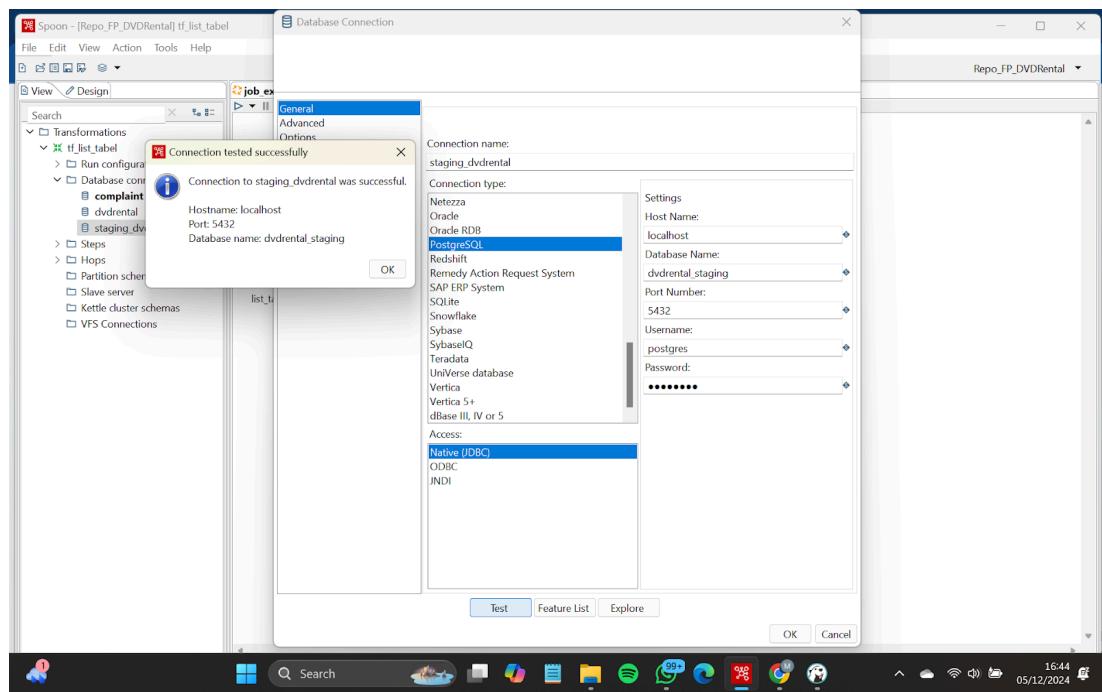
**Gambar 1.15.** Buka File Berisi File Transformasi



**Gambar 1.16.** Memastikan Direktori File Transformasi

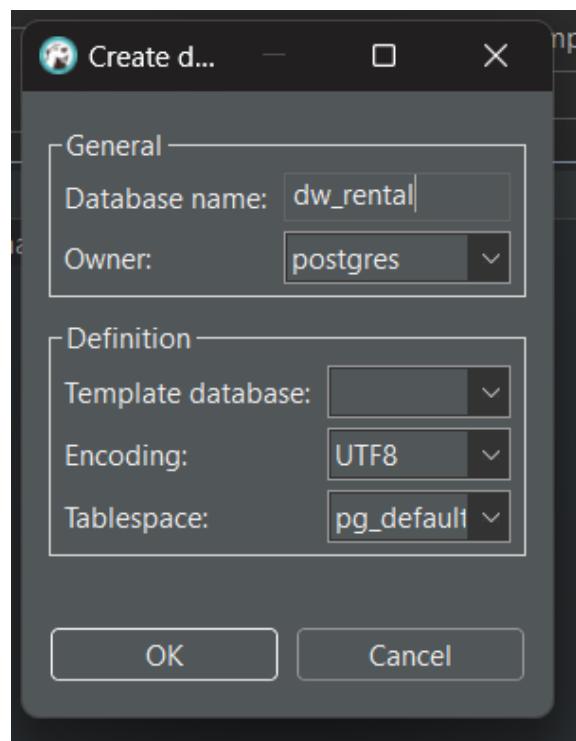


**Gambar 1.17. Membuat Koneksi Baru ke Database dvrental**

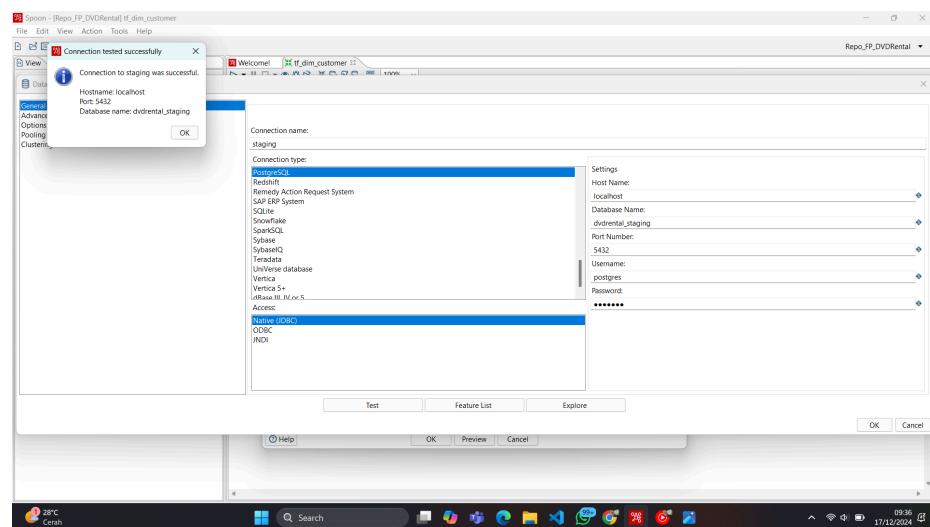


**Gambar 1.18. Membuat Koneksi Baru di Database dvrental\_staging**

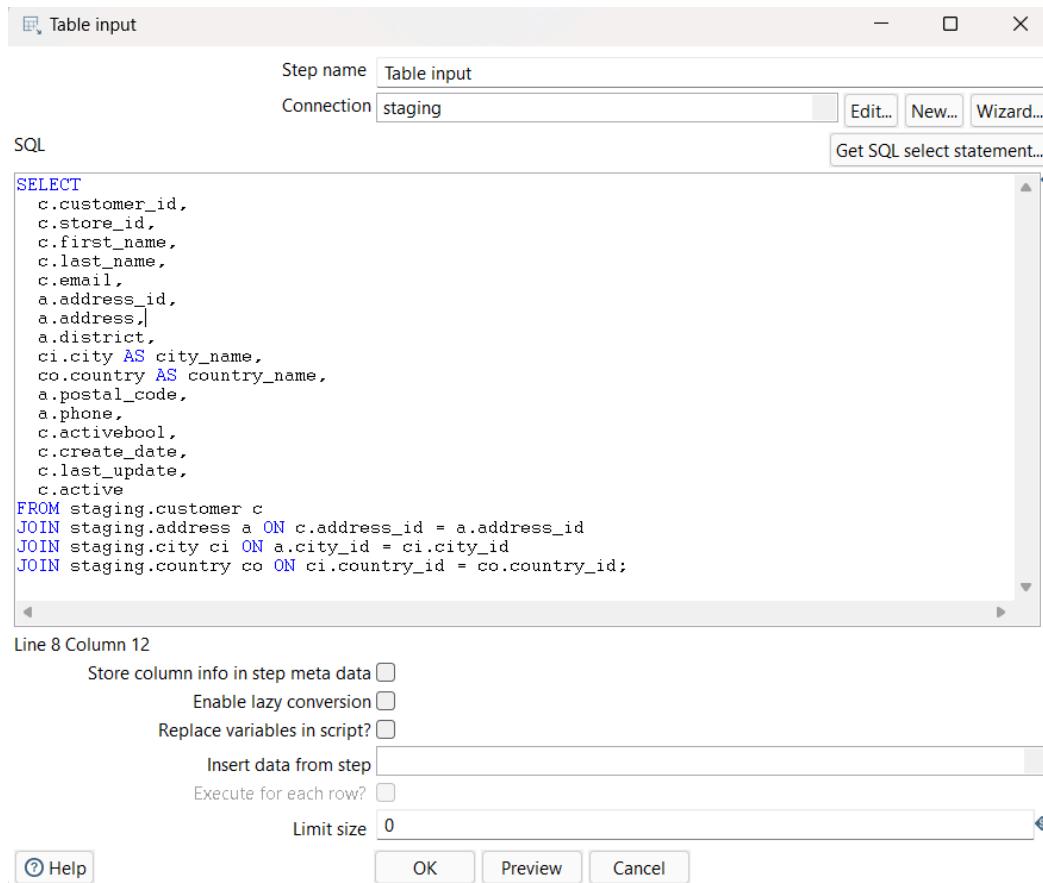
## 2. Pembuatan Data Warehouse



**Gambar 2.1** Membuat Database dw\_rental



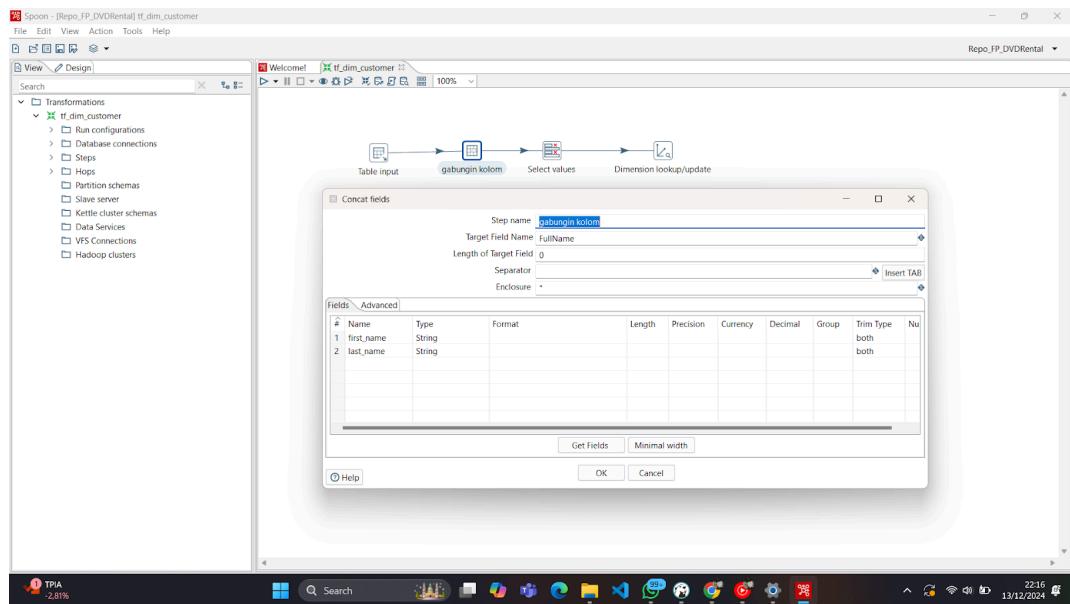
**Gambar 2.2. Menghubungkan Target Schema**



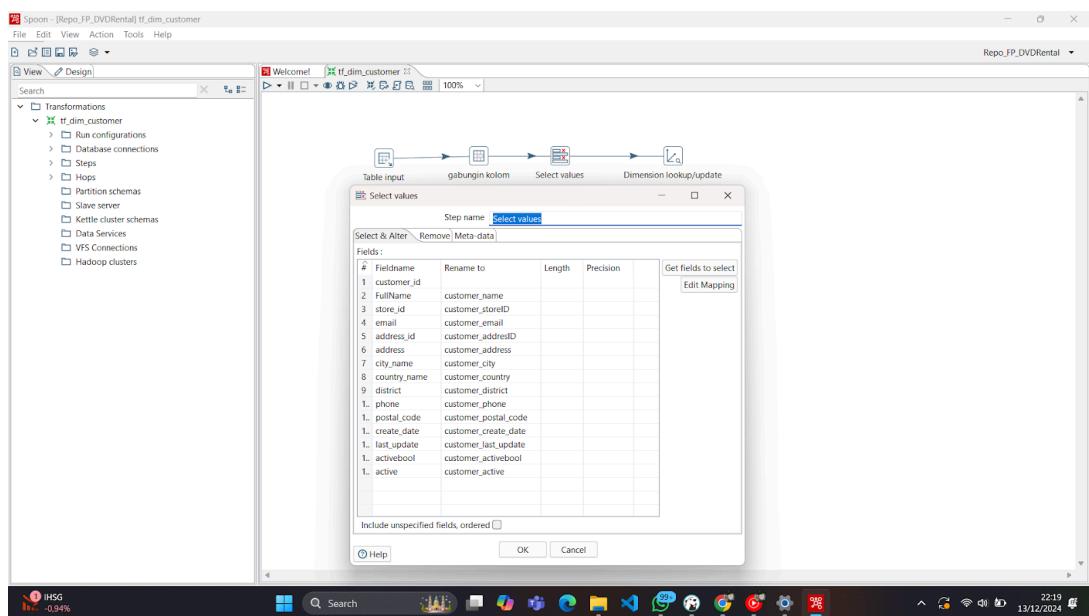
**Gambar 2.3 Isi dari Tabel Input Customer**

#	customer_id	store_id	first_name	last_name	email	address_id	address	district	city_name	country_name
1	524	1	Jared	Ely	jared.ely@sakilacustomer.org	530	1003 Qinhuangdao Street	West Java	Purwakarta	Indonesia
2	1	1	Mary	Smith	mary.smith@sakilacustomer.org	5	1913 Hanoi Way	Nagasaki	Sasebo	Japan
3	2	1	Patricia	Johnson	patricia.johnson@sakilacustomer.org	6	1121 Loja Avenue	California	San Bernardino	United States
4	3	1	Linda	Williams	linda.williams@sakilacustomer.org	7	692 Joliet Street	Attika	Athenai	Greece
5	4	2	Barbara	Jones	barbara.jones@sakilacustomer.org	8	1566 Ingle Manor	Mandalay	Myinyan	Myanmar
6	5	1	Elizabeth	Brown	elizabeth.brown@sakilacustomer.org	9	53 Idfu Parkway	Nantou	Nantou	Taiwan
7	6	2	Jennifer	Davis	jennifer.davis@sakilacustomer.org	10	1795 Santiago de Compostela Way	Texas	Laredo	United States
8	7	1	Maria	Miller	maria.miller@sakilacustomer.org	11	900 Santiago de Compostela Parkway	Central Serbia	Kragujevac	Yugoslavia
9	8	2	Susan	Wilson	susan.wilson@sakilacustomer.org	12	478 Joslin Way	Hamilton	Hamilton	New Zealand
10	9	2	Margaret	Moore	margaret.moore@sakilacustomer.org	13	613 Korley Drive	Masqat	Masqat	Oman
11	10	1	Dorothy	Taylor	dorothy.taylor@sakilacustomer.org	14	1531 Sal Drive	Esfahan	Esfahan	Iran
12	11	2	Lisa	Anderson	lisa.anderson@sakilacustomer.org	15	1542 Tarlac Parkway	Kanagawa	Sagamihara	Japan
13	12	1	Nancy	Thomas	nancy.thomas@sakilacustomer.org	16	808 Rhapsopal Manor	Haryana	Yamuna Nagar	India
14	13	2	Karen	Jackson	karen.jackson@sakilacustomer.org	17	270 Amroha Parkway	Osmaniye	Osmaniye	Turkey
15	14	2	Betty	White	betty.white@sakilacustomer.org	18	770 Bydgoszcz Avenue	California	Citrus Heights	United States
16	15	1	Helen	Harris	helen.harris@sakilacustomer.org	19	419 Iligan Lane	Madhya Pradesh	Bhopal	India
17	16	2	Sandra	Martin	sandra.martin@sakilacustomer.org	20	360 Toulose Parkway	England	Southend-on-Sea	United Kingdom
18	17	1	Donna	Thompson	donna.thompson@sakilacustomer.org	21	270 Toulon Boulevard	Kalmikia	Elista	Russian Federator
19	18	2	Carol	Garcia	carol.garcia@sakilacustomer.org	22	320 Brest Avenue	Kaduna	Kaduna	Nigeria
20	19	1	Ruth	Martinez	ruth.martinez@sakilacustomer.org	23	1417 Lancaster Avenue	Northern Cape	Kimberley	South Africa
21	20	2	Sharon	Robinson	sharon.robinson@sakilacustomer.org	24	1668 Okara Way	Northwest Border Prov	Mardan	Pakistan
22	21	1	Michelle	Clark	micelle.clark@sakilacustomer.org	25	262 A Corua (La Corua) Parkway	Dhaka	Tangail	Bangladesh
23	22	1	Laura	Rodriguez	laura.rodriguez@sakilacustomer.org	26	29 Charlotte Amalie Street	Rabat-Sal-Zemmour-Z	Sal	Morocco
24	23	2	Sarah	Lewis	sarah.lewis@sakilacustomer.org	27	1780 Hirao Boulevard	Liepaja	Liepaja	Latvia
25	24	2	Kimberly	Lee	kimberly.lee@sakilacustomer.org	28	96 Tafuna Way	Crdoba	Crdoba	Argentina
26	25	1	Deborah	Walker	deborah.walker@sakilacustomer.org	29	934 San Felipe de Puerto Plata Street	Sind	Shikarpur	Pakistan
27	26	2	Jessica	Hall	jessica.hall@sakilacustomer.org	30	18 Duisburg Boulevard	<null>	Citt del Vaticano	Holy See (Vatican)

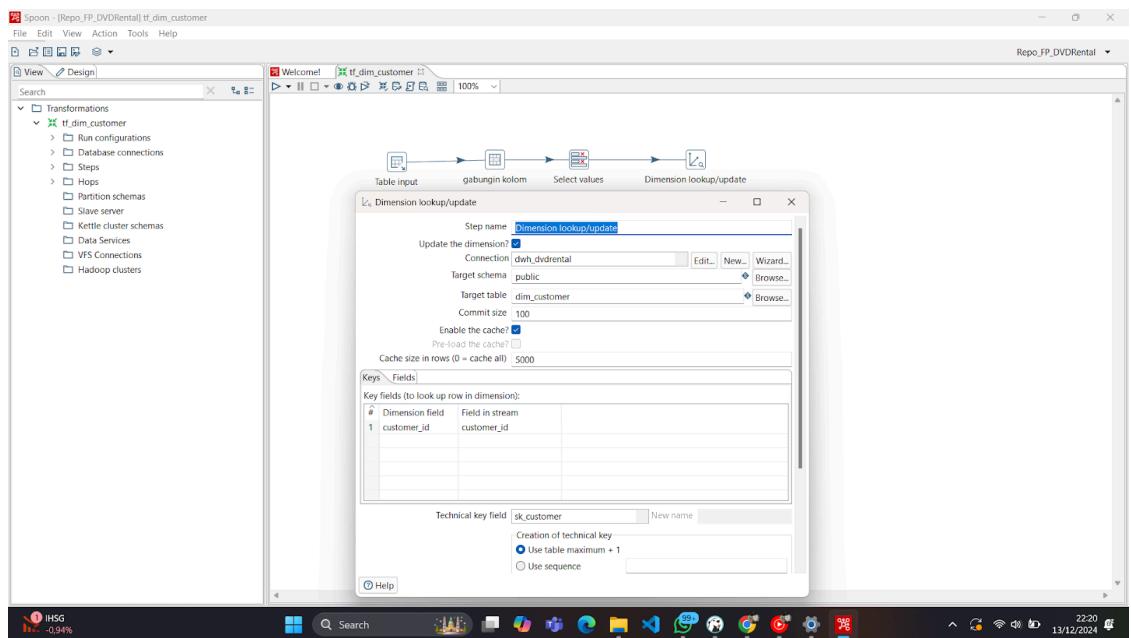
**Gambar 2.4 Preview Data**



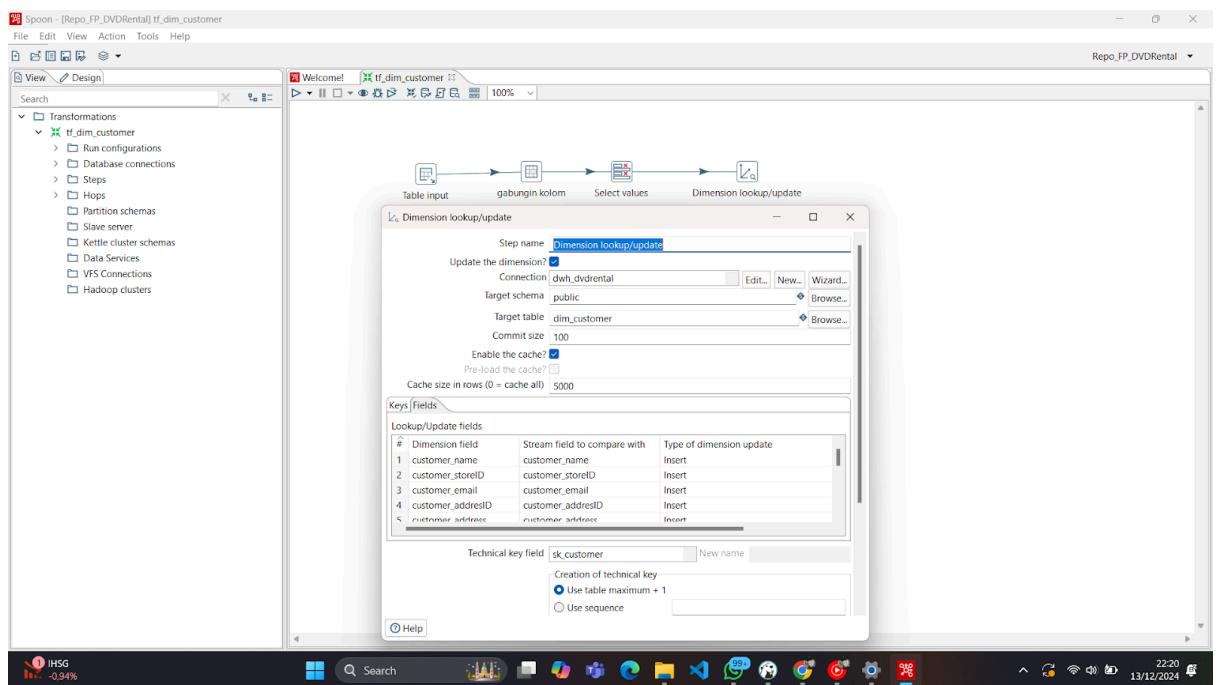
**Gambar 2.5 Penggabungan Kolom**



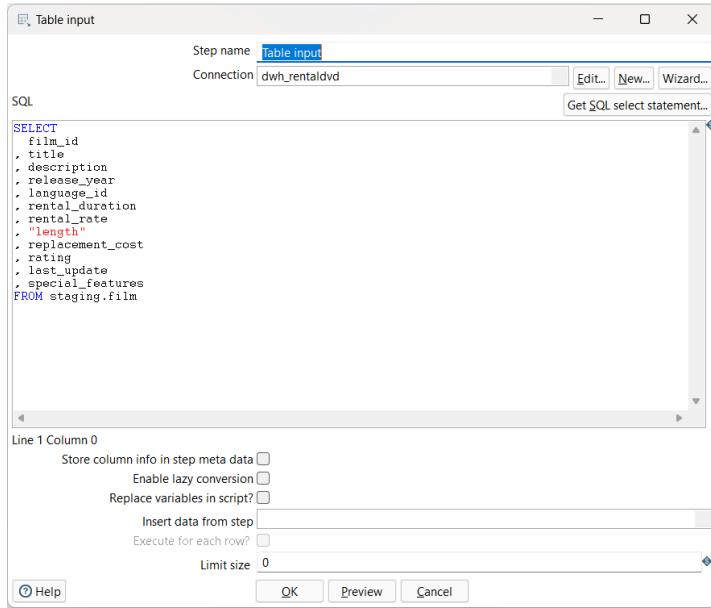
**Gambar 2.6 Penggunaan Select Values**



**Gambar 2.7. Isi dari Dimension Lookup**



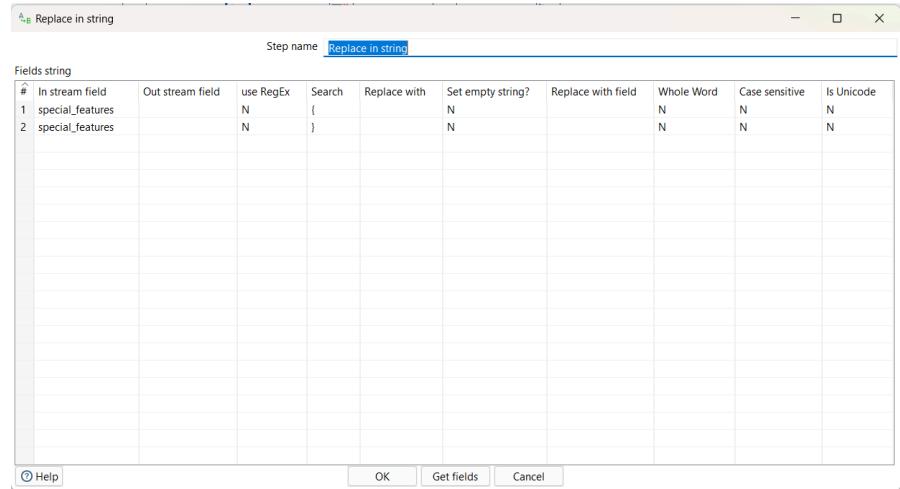
**Gambar 2.8. Isi dari Dimension Lookup**



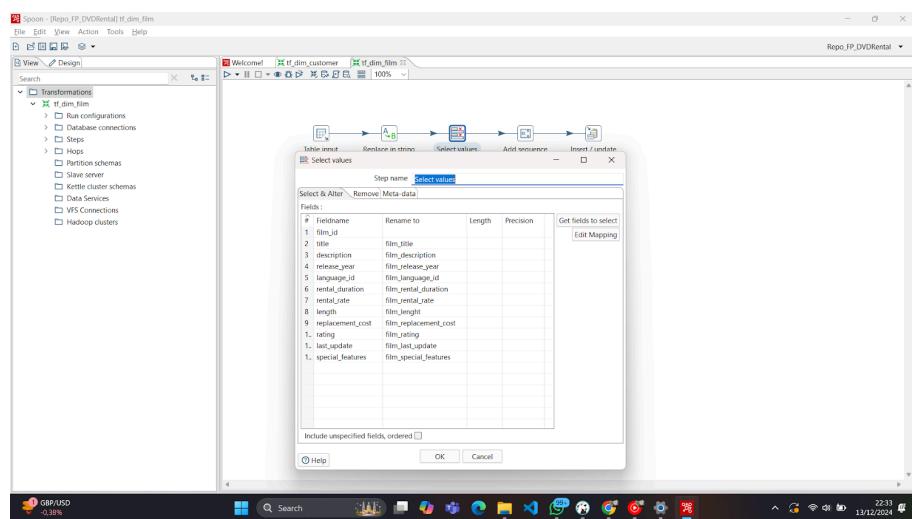
**Gambar 2.9.** Isi dari Tabel Input Film

	film_id	title	description	release_year	language_id	rental_duration	rent
1.	133	Chamber Italian	A Fateful Reflection of a Moose And a Husband who must Overcome a Monkey in Nigeria	2006	1	7	
2.	384	Grosse Wonderful	A Epic Drama of a Cat And a Explorer who must Redeem a Moose in Australia	2006	1	6	
3.	8	Airport Pollock	A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India	2006	1	4	
4.	98	Bright Encounters	A Fateful Year of a Monkey And a Fellow who must Conquer a Student in A Jet Boat	2006	1	6	
5.	1	Adventure Phrasaur	A Epic Drama of a Lumberjack And a Scientist who must Discover a Dentist in The Canadian Rockies	2006	1	3	
6.	2	Axe Goldtripe	A Astounding Escapade of a Database Administrator And a Explorer who must Find a Car in Ancient China	2006	1	7	
7.	3	Adaptation Holes	A Ascending Reflection of a Lumberjack And a Car who must Sive a Lumberjack in A Balcon Factory	2006	1	5	
8.	4	Affair Prejudice	A Fanciful Documentary of a Fribee And a Lumberjack who must Chase a Monkey in A Shark Tank	2006	1	6	
9.	5	African Egg	A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico	2006	1	3	
10.	6	Agent Truman	A Intrepid Panorama of a Robot And a Boy who must Escape a Sumo Wrestler in Ancient China	2006	1	6	
11.	7	Airplane Sierra	A Touching Saga of a Horse And a Butler who must Face a Bear in A Jet Boat	2006	1	6	
12.	9	Aladdin's Lamp	A Inspiring Drama of a Dentist And a Thief who must Meet a Mad Scientist in The Canadian Rockies	2006	1	3	
13.	10	Aladdin's Calendar	A Action-Packed Tale of a Horse And a Lumberjack who must Teach a Feminist in Ancient China	2006	1	6	
14.	11	Alamo Videotape	A Boring Epitome of a Butler And a Cat who must Fight a Pastry Chef in A MySQL Convention	2006	1	6	
15.	12	Alaska Phantom	A Fanciful Saga of a Hunter And a Pastry Chef who must Vanquish a Boy in Australia	2006	1	4	
16.	213	Date Speed	A Touching Saga of a Composer And a Moose who must Discover a Dentist in A MySQL Convention	2006	1	4	
17.	13	All Forever	A Action-Packed Drama of a Dentist And a Crocodile who must Battle a Dentist in The Canadian Rockies	2006	1	4	
18.	14	Alice Fantasia	A Emotional Drama of a Girl And a Datacenter Administrator who must Rescues a Pioneer in Soviet Georgia	2006	1	6	
19.	15	Amelia's Lover	A Boring Drama of a Cat And a Squirrel who must Battle a Feminist in A Jet Boat	2006	1	5	
20.	16	Alley Evolution	A Fast-Paced Drama of a Robot And a Composer who must Teach a Astronaut in New Orleans	2006	1	6	
21.	17	Above Trip	A Fast-Paced Character Study of a Composer And a Dog who must Outrun a Boat in An Abandoned Fun House	2006	1	3	
22.	18	After Victory	A Thoughtful Drama of a Composer And a Feminist who must Meet a Secret Agent in The Canadian Rockies	2006	1	6	
23.	19	Amadeus' Holy	A Emotional Display of a Painter And a Technical Writer who must Face a Man in A Baloon	2006	1	6	
24.	20	Amelie Hellfingers	A Boring Drama of a Woman And a Squirrel who must Conquer a Student in A Baloon	2006	1	4	
25.	21	American Circus	A Insightful Drama of a Girl And a Astronaut who must Face a Database Administrator in A Shark Tank	2006	1	3	
26.	22	Amistad Midsummer	A Emotional Character Study of a Dentist And a Crocodile who must Meet a Sumo Wrestler in California	2006	1	6	
27.	23	Anacrona Confessions	A Lackluster Display of a Dentist And a Dentist who must Fight a Girl in Australia	2006	1	3	

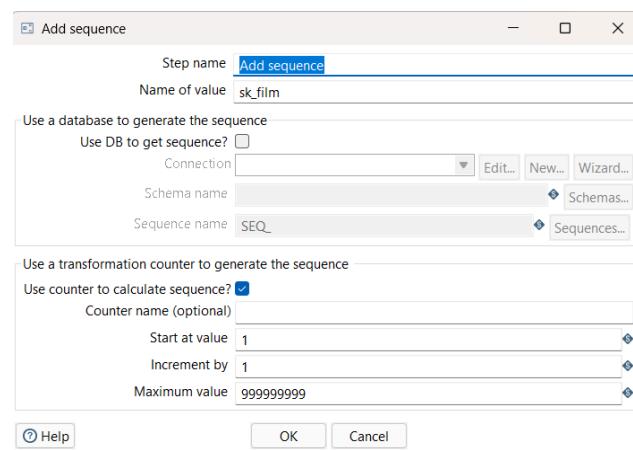
**Gambar 2.10.** Preview Data



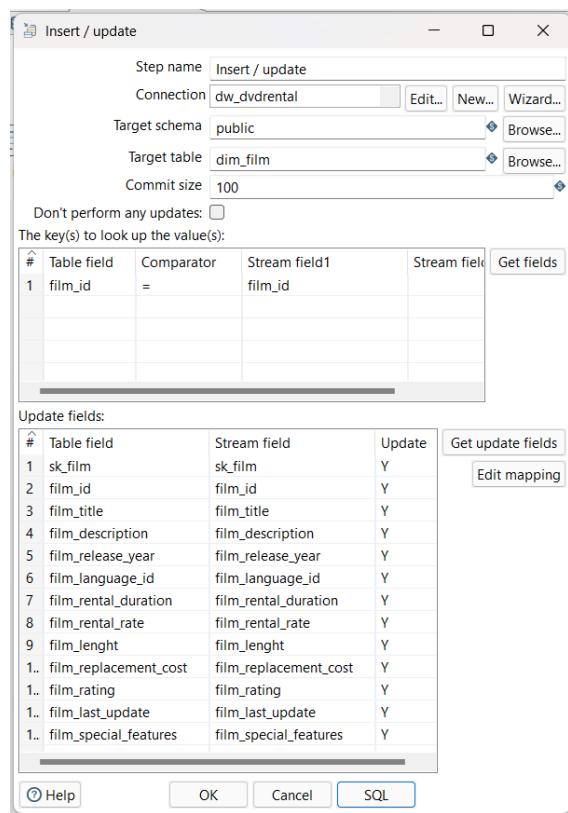
**Gambar 2.11. Pembersihan Simbol pada Data**



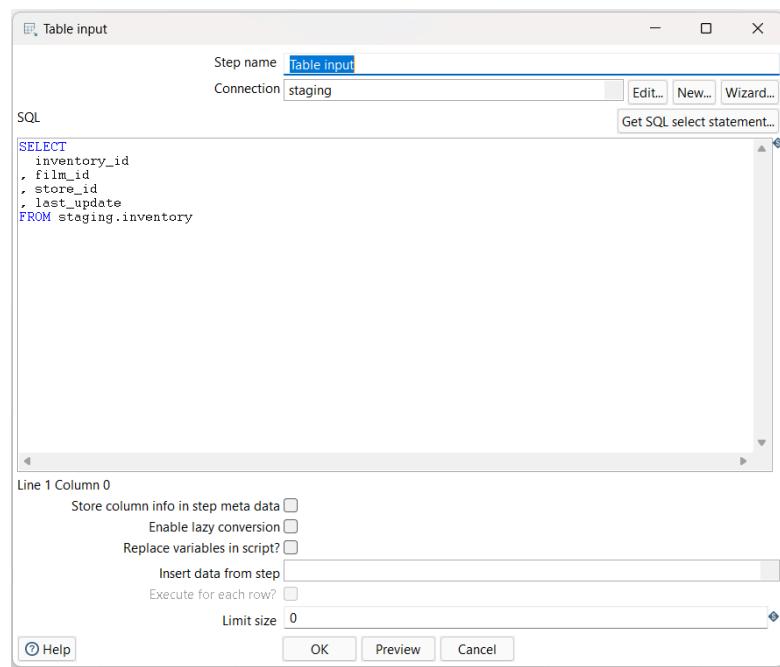
**Gambar 2.11. Rename Nama Tabel**



**Gambar 2.12.** Menambahkan SK baru



**Gambar 2.13.** Isi dari Insert/Update



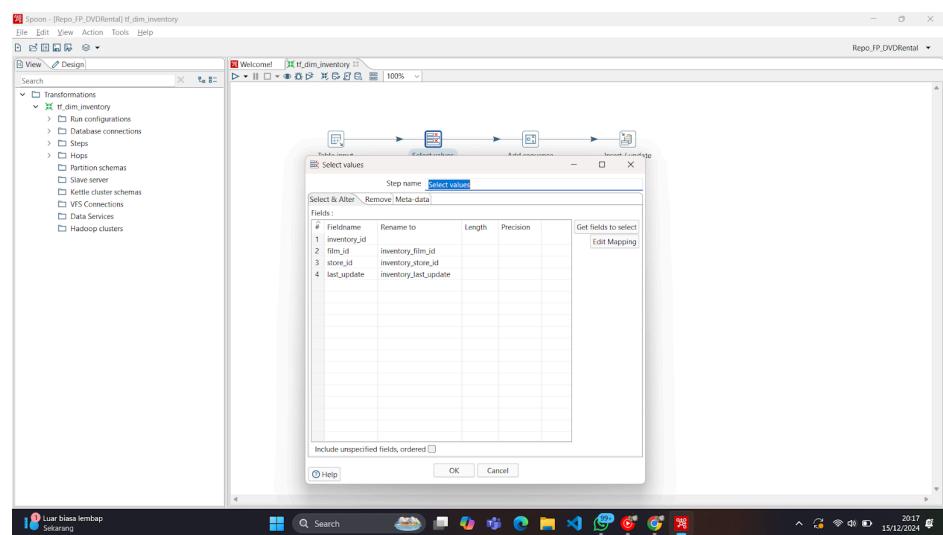
**Gambar 2.14.** Isi dari Tabel Input Inventory

Examine preview data

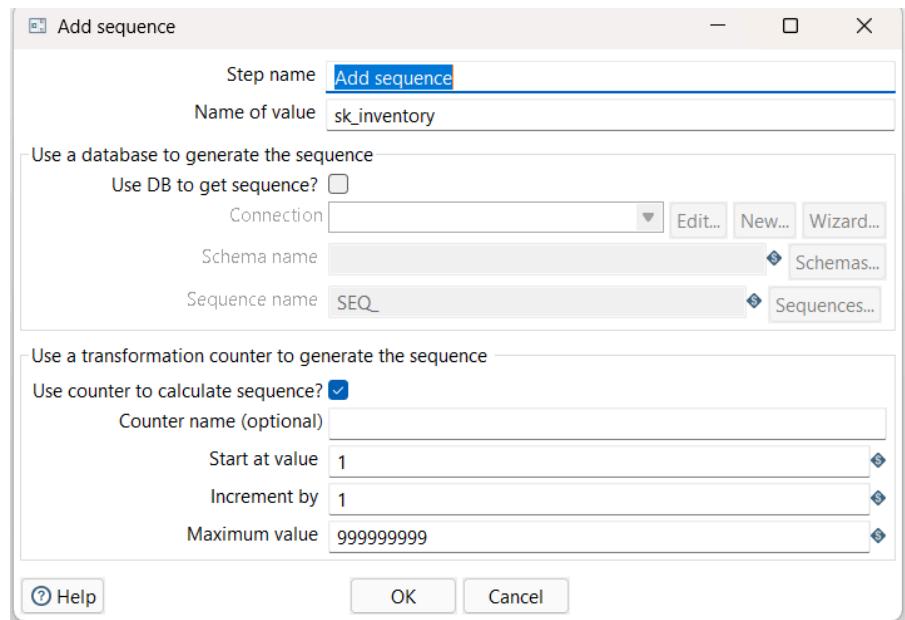
Rows of step: Table input (100 rows)

#	inventory_id	film_id	store_id	last_update
1	1	1	1	2006/02/15 10:09:17.000000000
2	2	1	1	2006/02/15 10:09:17.000000000
3	3	1	1	2006/02/15 10:09:17.000000000
4	4	1	1	2006/02/15 10:09:17.000000000
5	5	1	2	2006/02/15 10:09:17.000000000
6	6	1	2	2006/02/15 10:09:17.000000000
7	7	1	2	2006/02/15 10:09:17.000000000
8	8	1	2	2006/02/15 10:09:17.000000000
9	9	2	2	2006/02/15 10:09:17.000000000
1..	10	2	2	2006/02/15 10:09:17.000000000
1..	11	2	2	2006/02/15 10:09:17.000000000
1..	12	3	2	2006/02/15 10:09:17.000000000
1..	13	3	2	2006/02/15 10:09:17.000000000
1..	14	3	2	2006/02/15 10:09:17.000000000

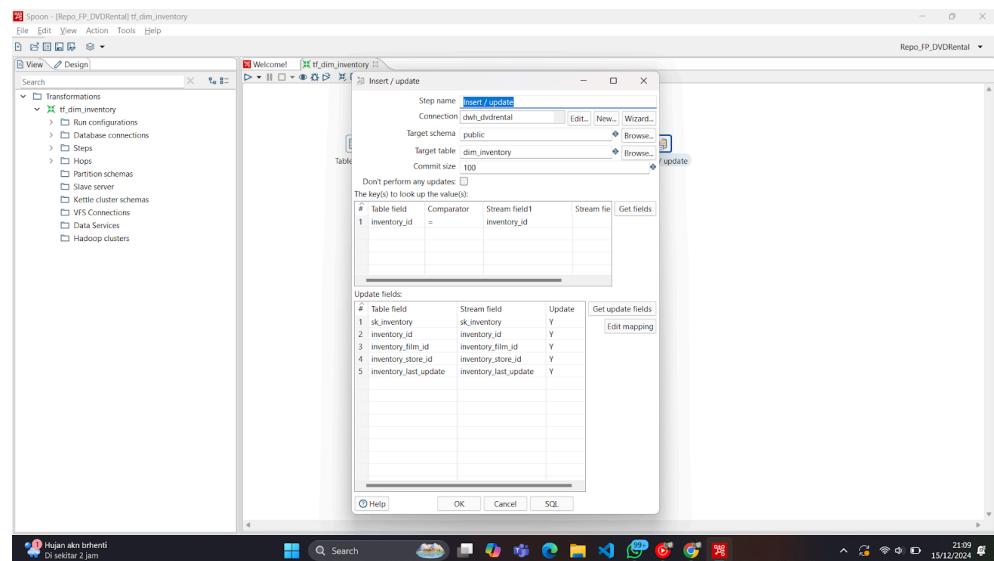
**Gambar 2.15.** Preview Data



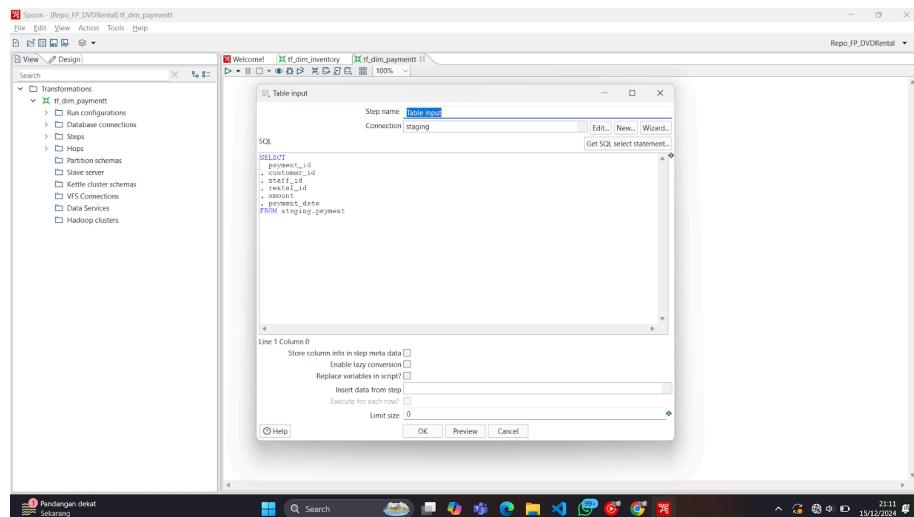
**Gambar 2.16.** Rename Nama Kolom



**Gambar 2.17. Membuat SK Baru**



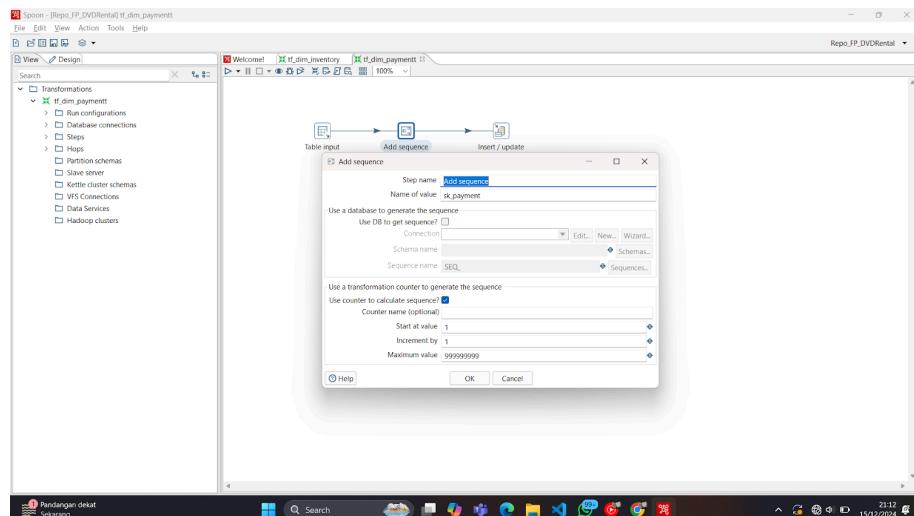
**Gambar 2.18. Isi dari Insert/Update**



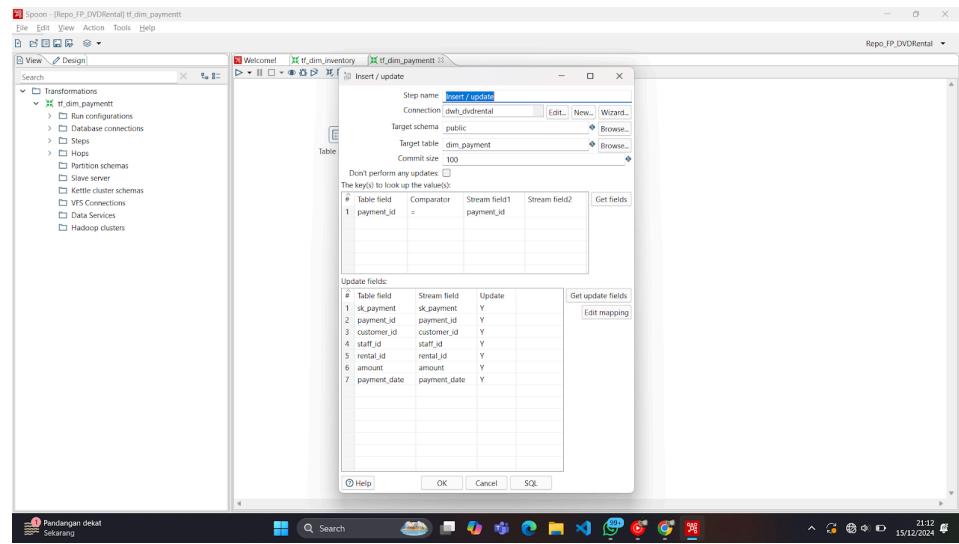
**Gambar 2.19.** Tabel Input Payment

	payment_id	customer_id	staff_id	rental_id	amount	payment_date
1.	17503	341	2	1520	7.99	2007/02/12 22:25:45.99577000
2.	17504	341	1	1778	1.99	2007/02/12 22:25:45.99577000
3.	17505	341	1	1881	7.99	2007/02/12 22:41:45.99577000
4.	17506	341	2	2829	2.99	2007/02/12 19:35:45.99577000
5.	17507	341	2	3130	7.99	2007/02/12 19:31:45.99577000
6.	17508	341	1	3382	5.99	2007/02/12 12:33:45.99577000
7.	17509	342	2	2190	5.99	2007/02/12 23:58:15.99577000
8.	17510	342	1	2314	1.99	2007/02/12 23:58:15.99577000
9.	17511	342	1	3881	2.99	2007/02/12 03:57:30.99577000
10.	17512	343	2	1547	4.99	2007/02/12 03:10:50.99577000
11.	17513	343	1	1564	6.99	2007/02/12 01:53:30.99577000
12.	17514	343	2	1879	0.99	2007/02/12 01:24:37.99577000
13.	17515	343	2	1584	0.99	2007/02/12 01:24:37.99577000
14.	17516	343	2	2461	6.99	2007/02/12 18:26:30.99577000
15.	17517	343	1	2980	8.99	2007/02/12 07:03:20.99577000
16.	17518	343	1	3407	0.99	2007/02/12 14:42:38.99577000
17.	17519	344	1	1547	6.99	2007/02/12 15:54:40.99577000
18.	17520	344	2	1475	4.99	2007/02/12 14:03:45.99577000
19.	17521	344	1	1731	0.99	2007/02/12 14:03:45.99577000
20.	17522	345	2	1210	0.99	2007/02/12 01:26:17.99577000
21.	17523	345	1	1457	4.99	2007/02/12 18:34:15.99577000
22.	17524	345	2	1556	0.99	2007/02/12 18:34:15.99577000
23.	17525	346	2	2366	4.99	2007/02/12 16:12:41.99577000
24.	17526	346	1	1994	5.99	2007/02/12 19:35:32.99577000
25.	17527	346	2	3372	2.99	2007/02/12 12:02:45.99577000
26.	17528	346	1	3421	2.99	2007/02/12 15:51:24.99577000
27.	17529	347	2	1711	8.99	2007/02/12 12:46:18.99577000
28.	17530	347	2	2214	0.99	2007/02/12 04:59:41.99577000

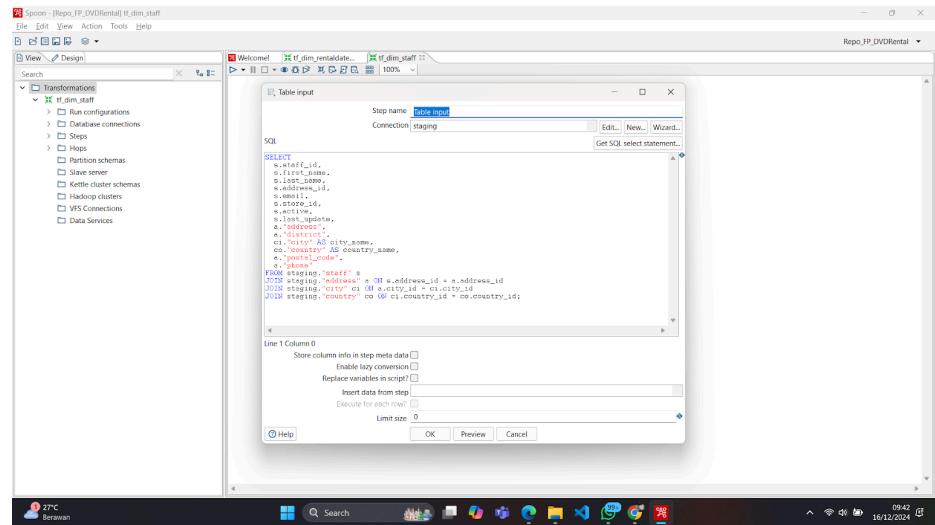
**Gambar 2.20.** Preview Data



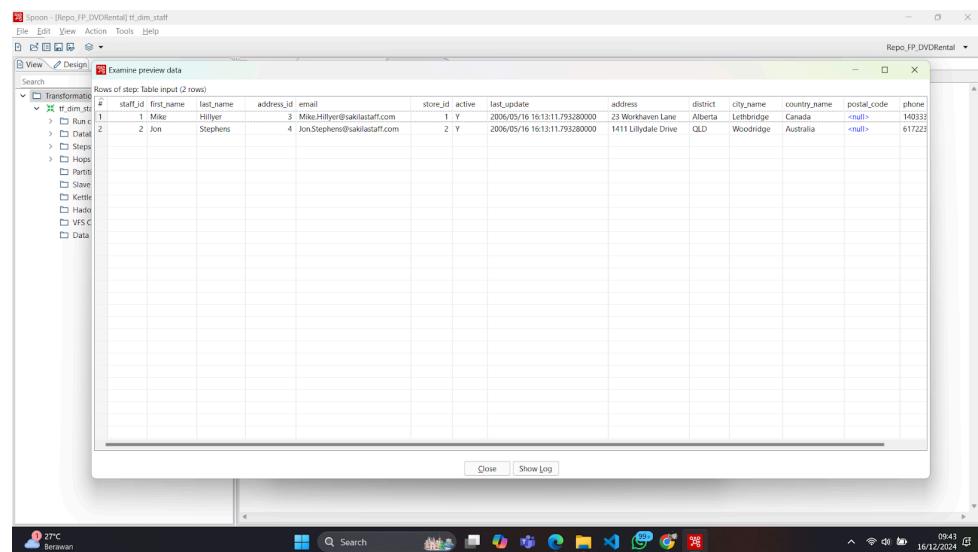
**Gambar 2.21. Membuat SK Baru**



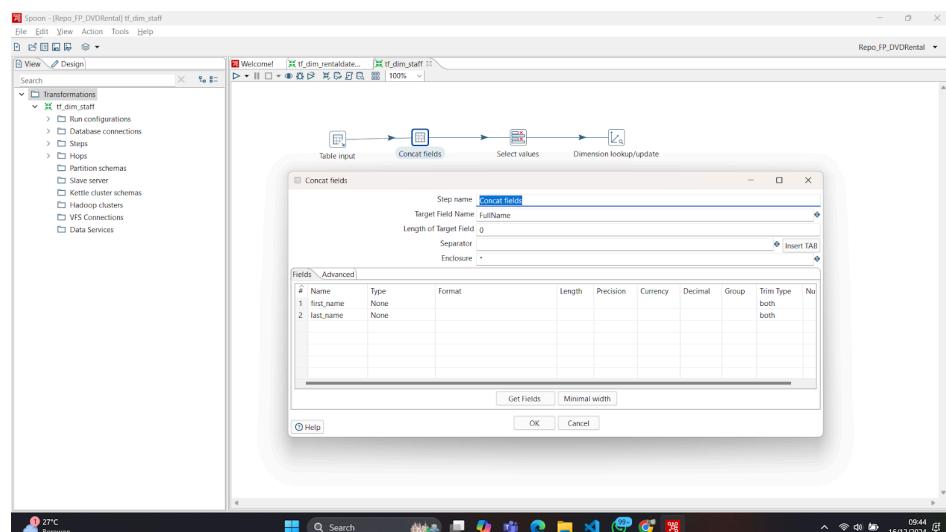
**Gambar 2.22. Isi dari Insert/Update**



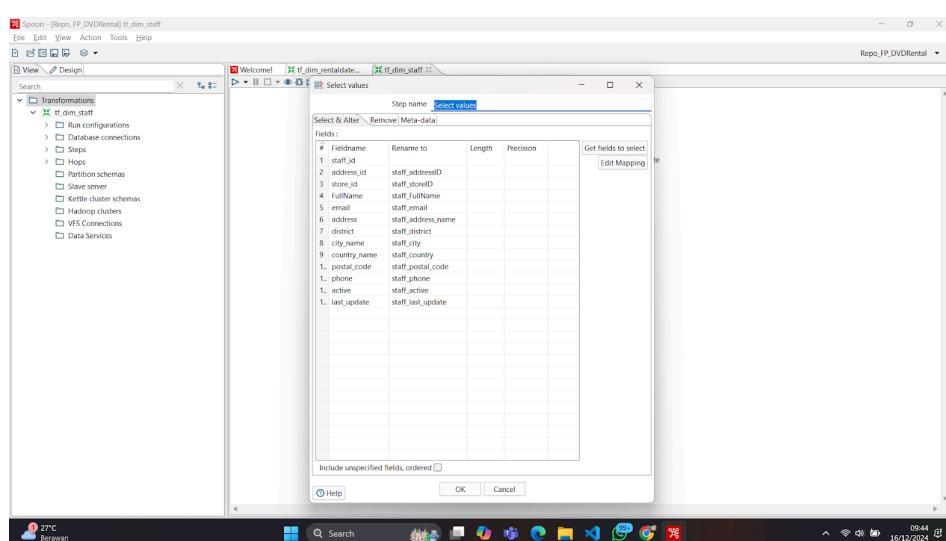
**Gambar 2.23. Tabel Input Dimensi Staff**



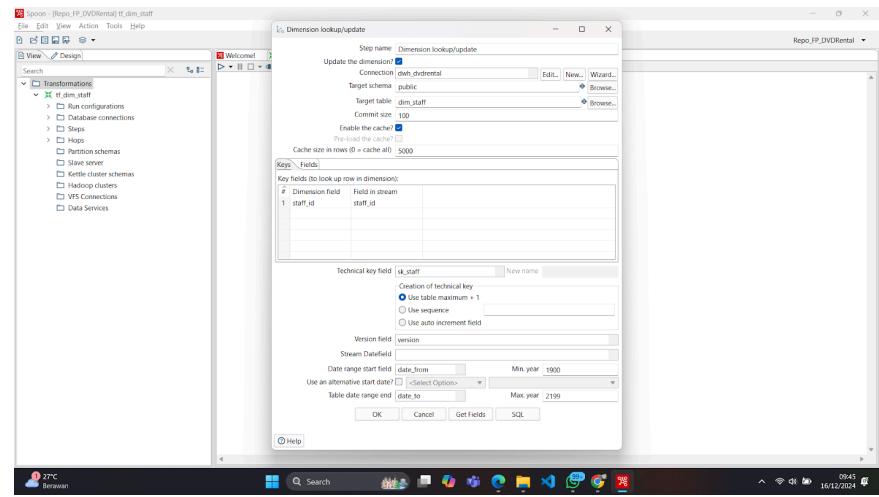
**Gambar 2.24. Preview Data**



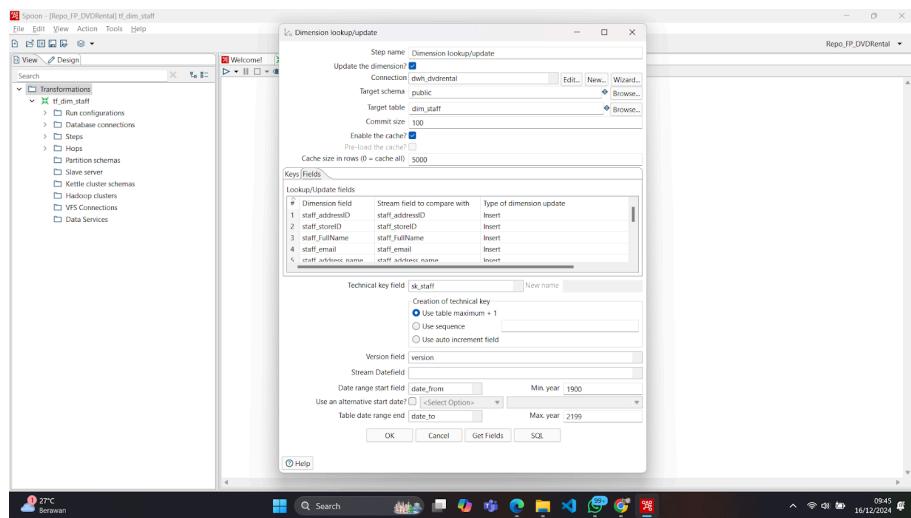
**Gambar 2.25. Penggabungan Kolom**



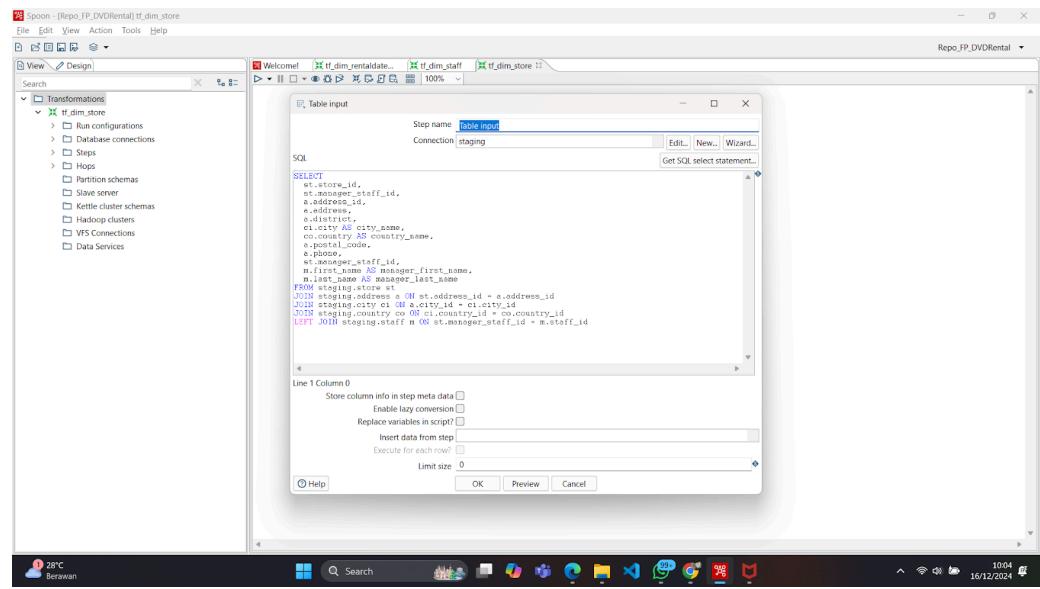
**Gambar 2.26. Rename Kolom**



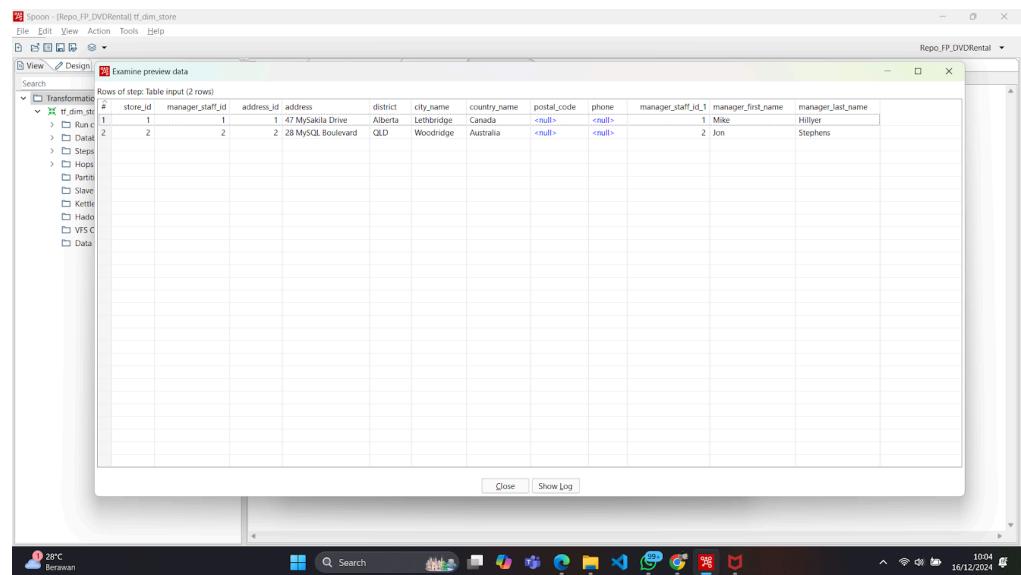
**Gambar 2.27. Isi dari Dimension Lookup**



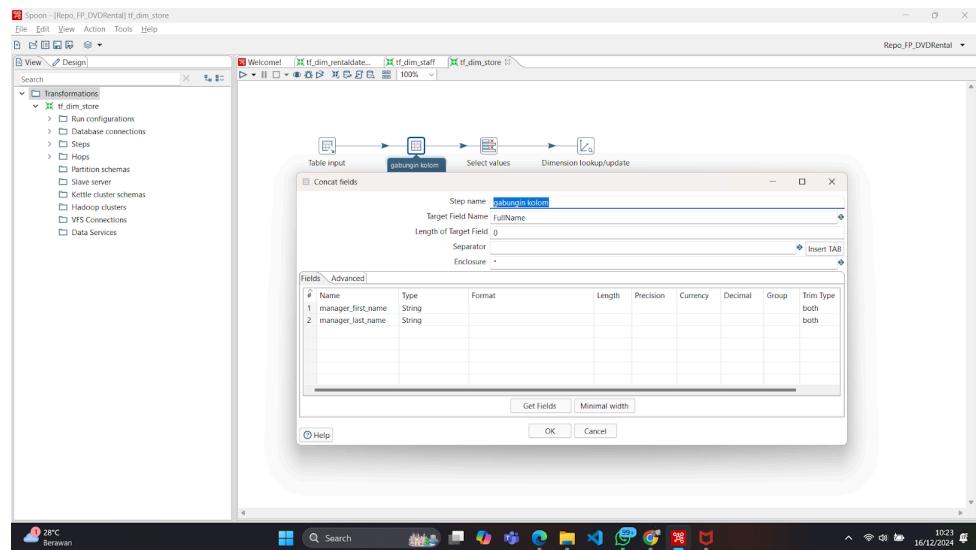
**Gambar 2.28. Isi dari Dimension Lookup**



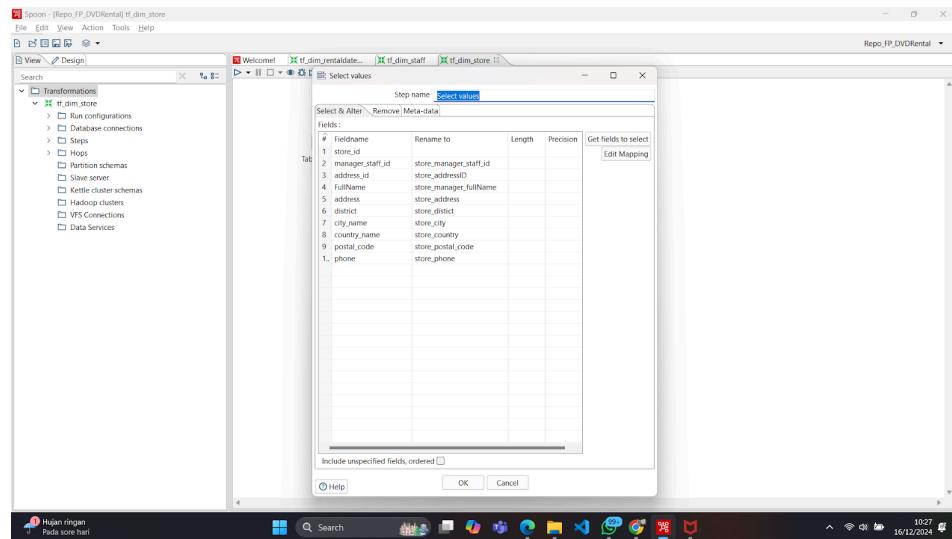
**Gambar 2.29. Tabel Input Dimensi Store**



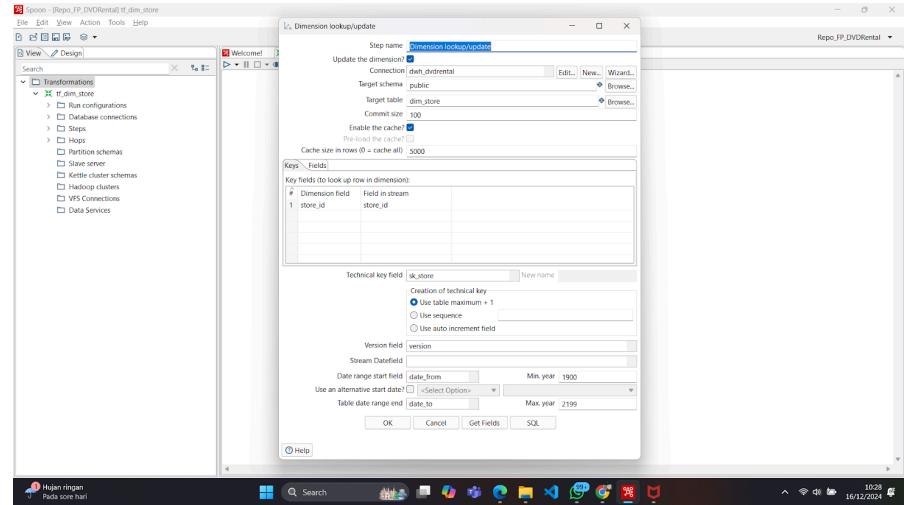
**Gambar 2.30. Preview Data**



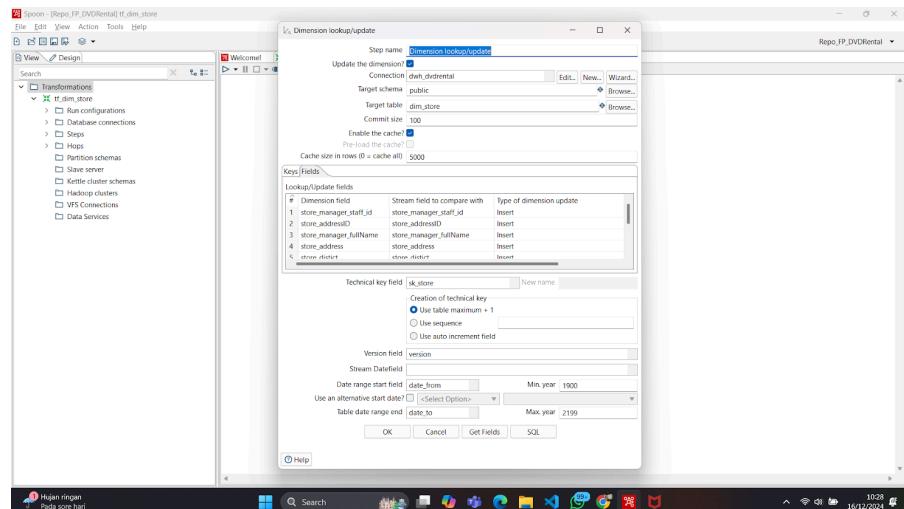
**Gambar 2.31. Penggabungan Kolom**



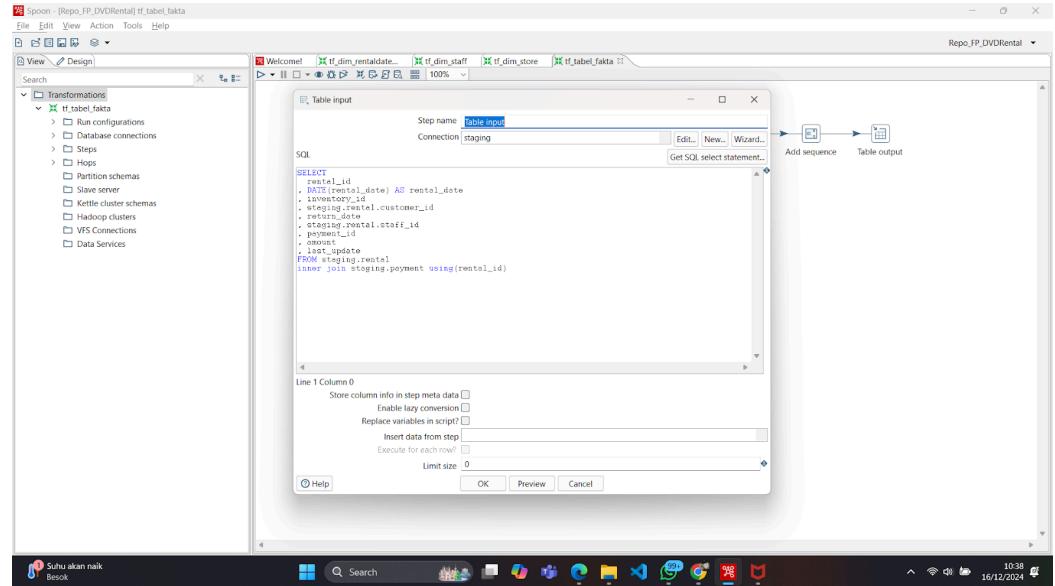
**Gambar 2.32. Rename Kolom**



**Gambar 2.23.** Isi dari Dimension Lookup



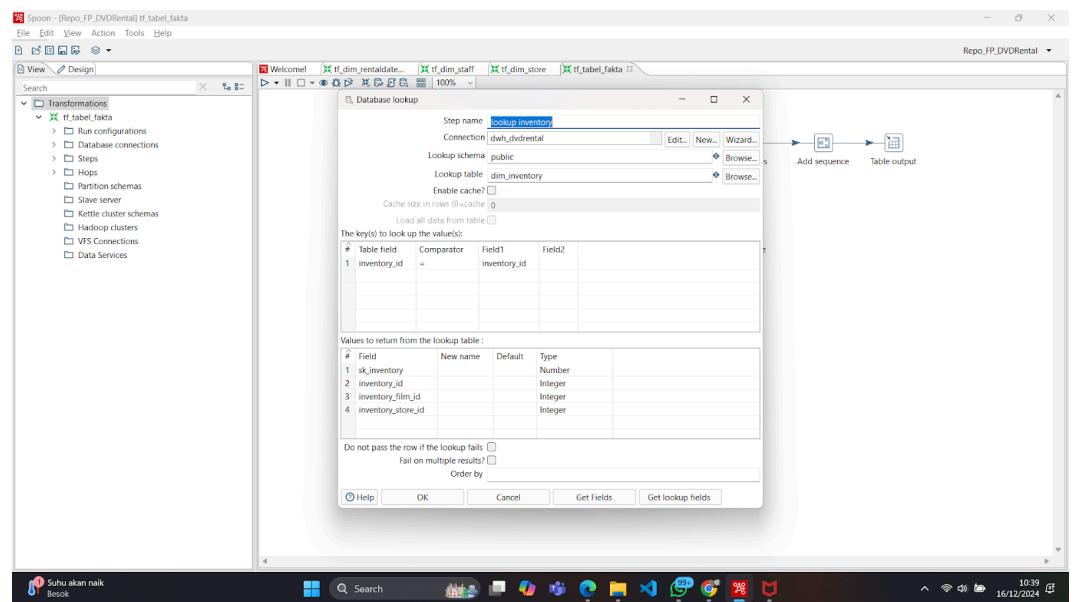
**Gambar 2.24.** Isi dari Dimension Lookup



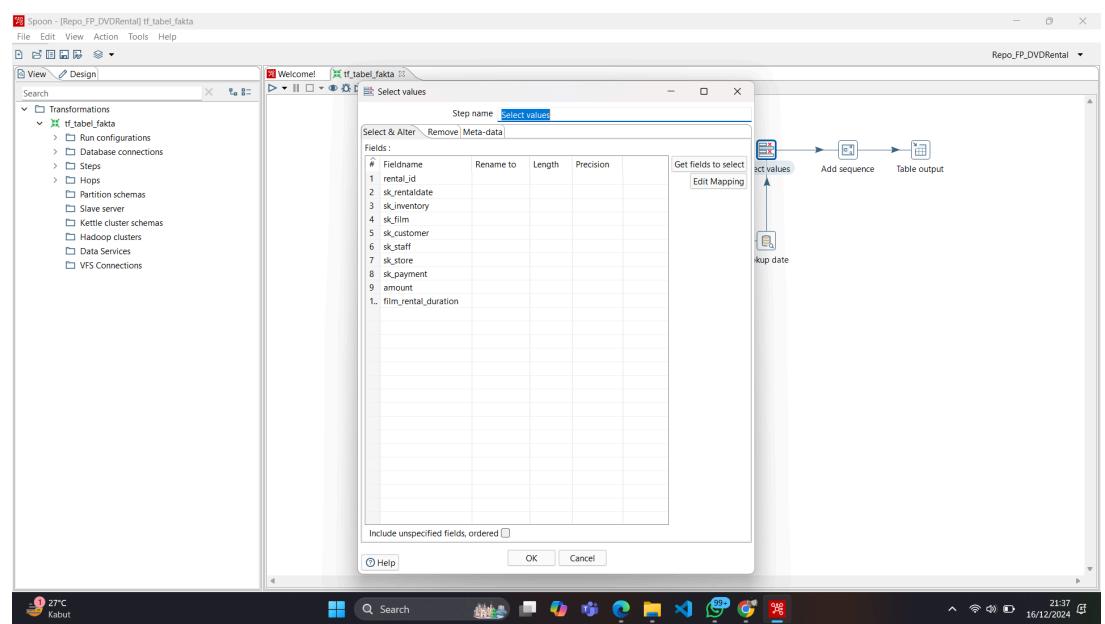
**Gambar 2.25.** Tabel Input Fact Table

#	rental_id	rental_date	inventory_id	customer_id	return_date	staff_id	payment_id	amount	last_update
1	1520	2005/06/15 00:00:00.000	3419	341	2005/06/24 23:46:20.000000000	1	17503	7.99	2006/02/16 02:30:53.000000000
2	1778	2005/06/16 00:00:00.000	2512	341	2005/06/22 16:08:48.000000000	2	17504	1.99	2006/02/16 02:30:53.000000000
3	1840	2005/06/17 00:00:00.000	2507	341	2005/06/23 18:27:19.000000000	2	17505	7.99	2006/02/16 02:30:53.000000000
4	2829	2005/06/19 00:00:00.000	2047	341	2005/06/24 18:10:30.000000000	1	17506	2.99	2006/02/16 02:30:53.000000000
5	3130	2005/06/20 00:00:00.000	2569	341	2005/06/25 18:05:22.000000000	2	17507	7.99	2006/02/16 02:30:53.000000000
6	3382	2005/06/21 00:00:00.000	131	341	2005/06/29 19:13:23.000000000	2	17508	5.99	2006/02/16 02:30:53.000000000
7	3160	2005/06/22 00:00:00.000	4658	342	2005/06/25 18:05:20.000000000	2	17509	5.99	2006/02/16 02:30:53.000000000
8	2914	2005/06/25 00:00:00.000	2419	342	2005/06/25 08:45:10.000000000	2	17510	4.99	2006/02/16 02:30:53.000000000
9	3081	2005/06/20 00:00:00.000	2071	342	2005/06/24 21:05:13.000000000	2	17511	2.99	2006/02/16 02:30:53.000000000
10	1547	2005/06/16 00:00:00.000	2939	343	2005/06/19 01:13:24.000000000	1	17512	4.99	2006/02/16 02:30:53.000000000
11	1544	2005/06/16 00:00:00.000	145	343	2005/06/24 03:12:07.000000000	1	17513	6.99	2006/02/16 02:20:53.000000000
12	1870	2005/06/17 00:00:00.000	3983	343	2005/06/19 00:00:34.000000000	1	17514	0.99	2006/02/16 02:20:53.000000000
13	1932	2005/06/17 00:00:00.000	2484	343	2005/06/18 09:15:25.000000000	2	17515	0.99	2006/02/16 02:30:53.000000000
14	2461	2005/06/18 00:00:00.000	78	343	2005/06/28 01:35:12.000000000	2	17516	6.99	2006/02/16 02:30:53.000000000
15	2980	2005/06/20 00:00:00.000	310	343	2005/06/29 07:57:03.000000000	2	17517	8.99	2006/02/16 02:30:53.000000000
16	3407	2005/06/21 00:00:00.000	3919	343	2005/06/24 19:38:02.000000000	2	17518	0.99	2006/02/16 02:30:53.000000000
17	1341	2005/06/15 00:00:00.000	3363	344	2005/06/21 07:26:18.000000000	2	17519	3.99	2006/02/16 02:30:53.000000000
18	1475	2005/06/16 00:00:00.000	1834	344	2005/06/18 22:33:01.000000000	2	17520	4.99	2006/02/16 02:30:53.000000000
19	1731	2005/06/16 00:00:00.000	3698	344	2005/06/19 18:58:12.000000000	2	17521	0.99	2006/02/16 02:30:53.000000000
20	1210	2005/06/15 00:00:00.000	77	345	2005/06/20 01:41:00.000000000	2	17522	0.99	2006/02/16 02:30:53.000000000
21	1627	2005/06/16 00:00:00.000	739	345	2005/06/20 01:41:00.000000000	1	17523	0.99	2006/02/16 02:30:53.000000000
22	1591	2005/06/16 00:00:00.000	2003	345	2005/06/19 23:56:35.000000000	1	17524	0.99	2006/02/16 02:30:53.000000000
23	2766	2005/06/19 00:00:00.000	3421	345	2005/06/23 20:11:15.000000000	2	17525	4.99	2006/02/16 02:30:53.000000000
24	1994	2005/06/17 00:00:00.000	2203	346	2005/06/25 08:26:06.000000000	2	17526	5.99	2006/02/16 02:30:53.000000000
25	3372	2005/06/21 00:00:00.000	4169	346	2005/06/27 08:41:19.000000000	2	17527	2.99	2006/02/16 02:30:53.000000000
26	3421	2005/06/21 00:00:00.000	3930	346	2005/06/25 18:57:58.000000000	1	17528	2.99	2006/02/16 02:30:53.000000000
27	1711	2005/06/16 00:00:00.000	273	347	2005/06/25 08:49:52.000000000	1	17529	8.99	2006/02/16 02:30:53.000000000
28	2274	2005/06/18 00:00:00.000	3021	347	2005/06/21 09:24:15.000000000	2	17530	0.99	2006/02/16 02:30:53.000000000

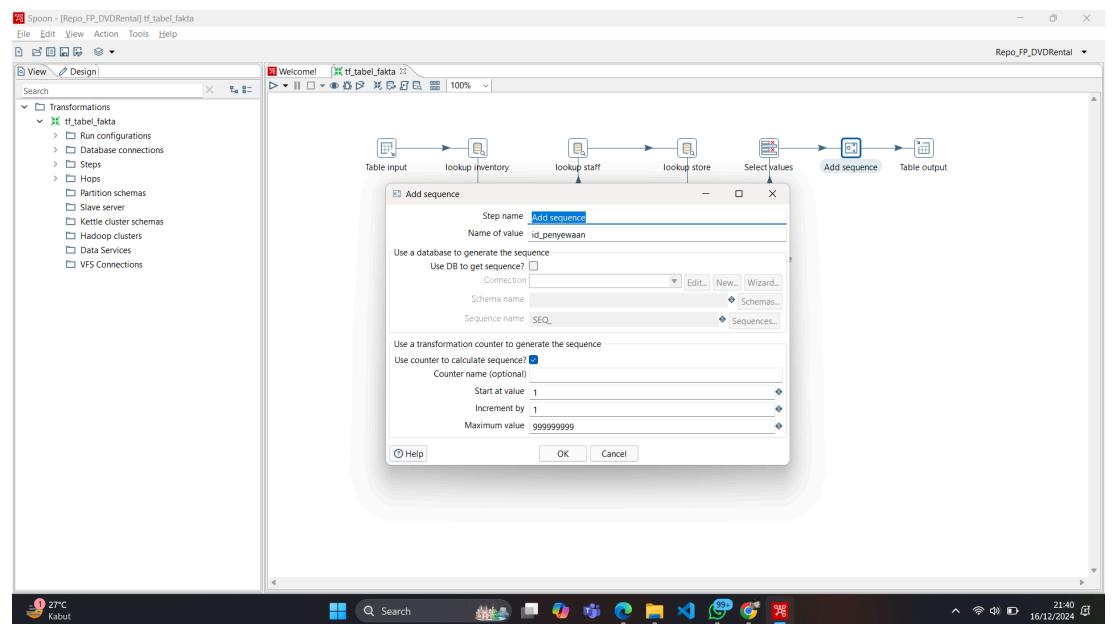
**Gambar 2.26.** Preview Data



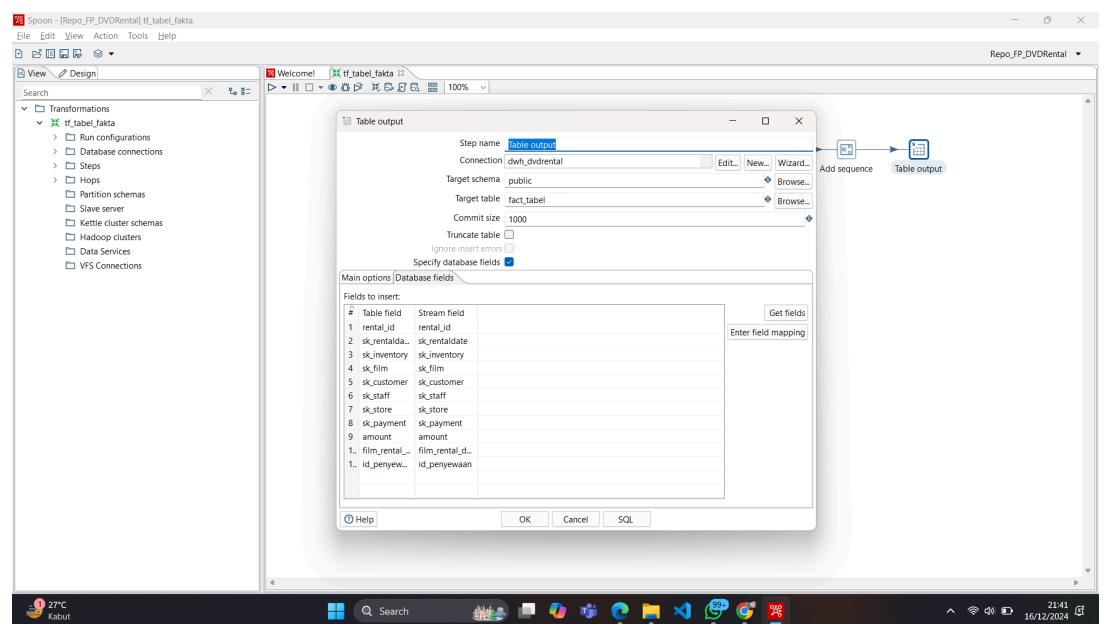
**Gambar 2.28. Database Lookup Fact Table**



**Gambar 2.29. Memilih Kolom Relevan**

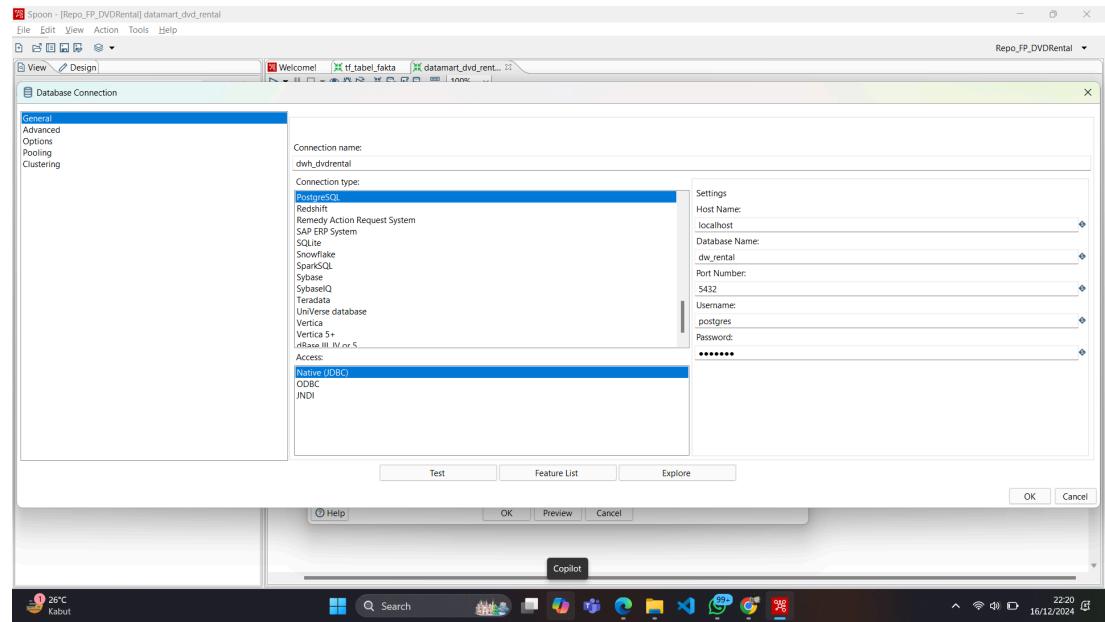


**Gambar 2.30. Membuat SK baru Untuk Fact Table**

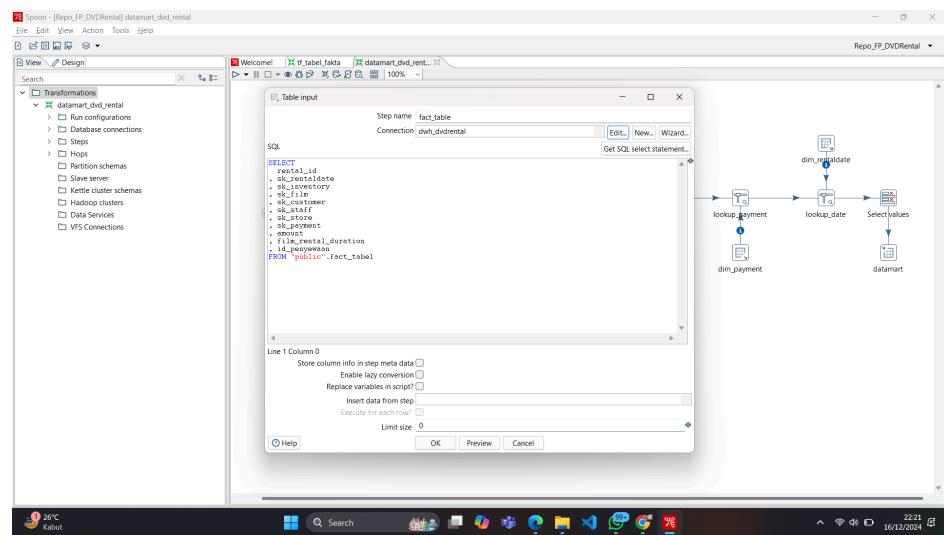


**Gambar 2.31. Eksekusi Tabel Output**

### 3. Pembuatan Datamart



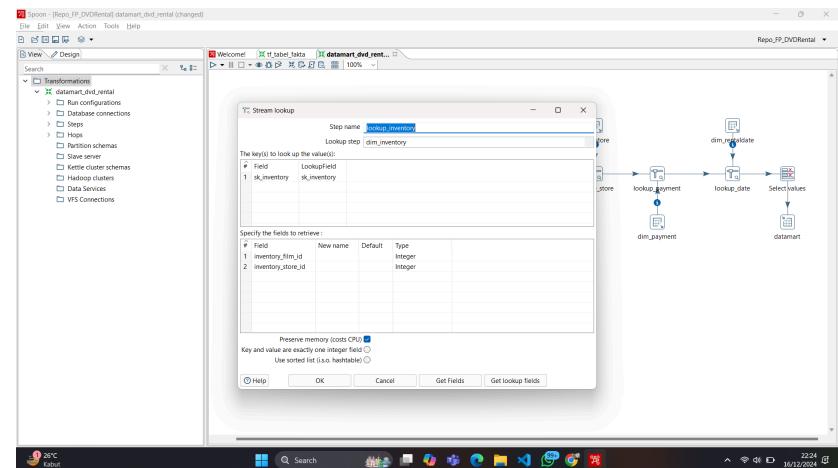
**Gambar 3.1. Buat Koneksi Baru**



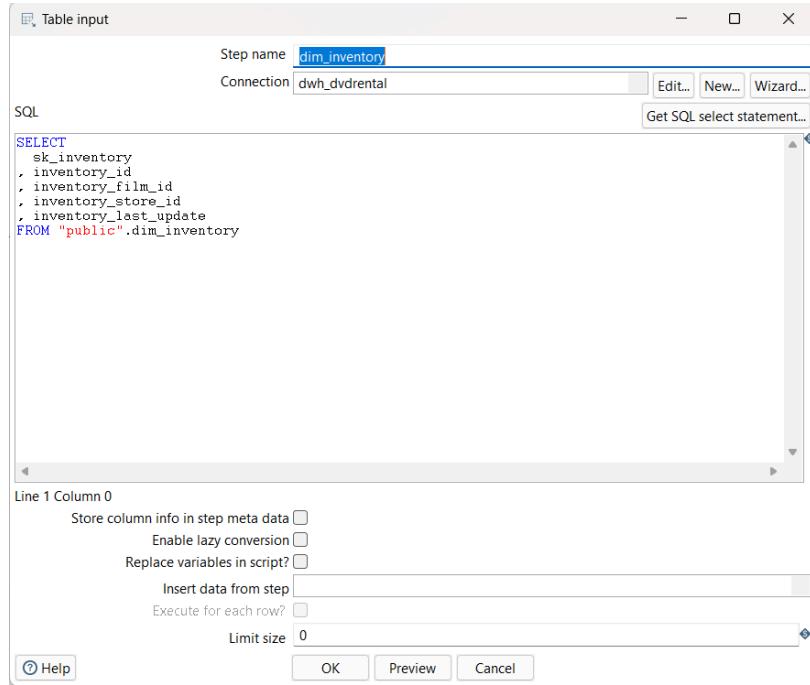
**Gambar 3.2. Tabel Input Datamart**

The screenshot shows a data preview window titled "Examine preview data" for the "fact\_table" table. The table contains 1000 rows and 13 columns. The columns are: rental\_id, sk\_rentaldate, sk\_inventory, sk\_film, sk\_customer, sk\_staff, sk\_store, sk\_payment, amount, film\_rental\_duration, id\_penyewaan, 1, and 10. The data includes various transaction details like rental dates, item IDs, customer IDs, staff IDs, store IDs, payment methods, amounts, and rental durations.

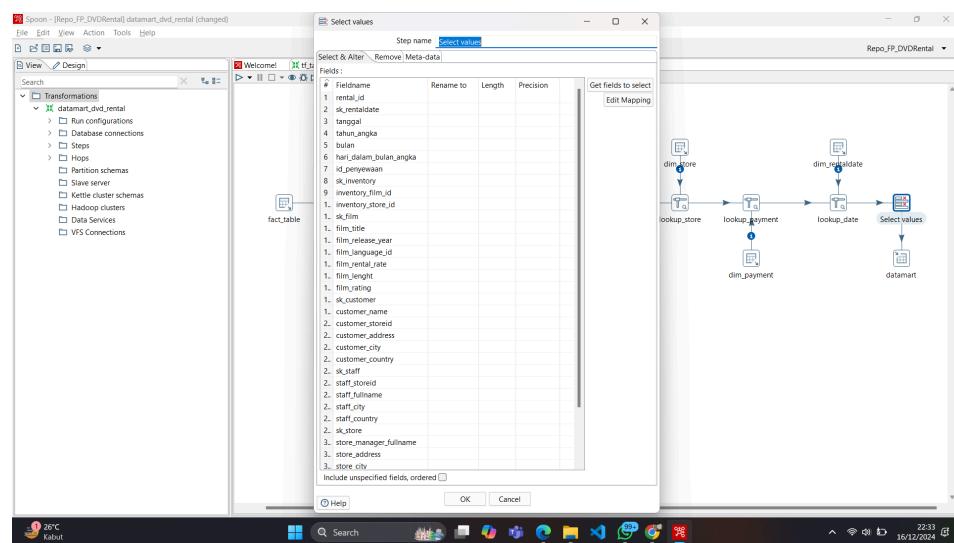
**Gambar 3.3. Preview Data**



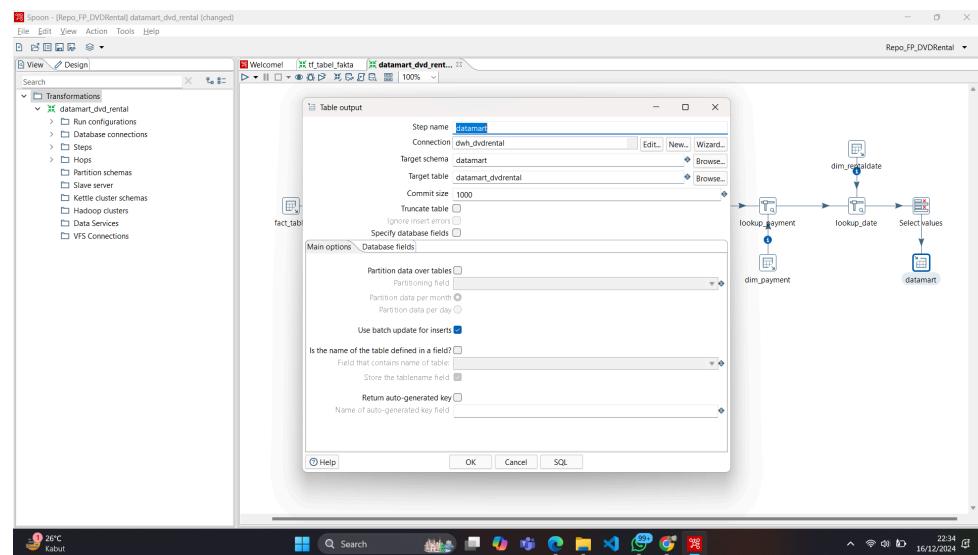
**Gambar 3.4. Stream Lookup Dimensi**



**Gambar 3.5. Tabel Input Inventory**



**Gambar 3.6. Memilih Kolom Relevan**



**Gambar 3.7. Eksekusi Tabel Output Datamart**

#### 4. Pembuatan Visualisasi Data melalui Tableau sudah didokumentasikan pada Bab 3.