

Analisis Operasional Toko DVD Rental dan Preferensi Film Berdasarkan *Genre*



Laporan *Final Project Data Analytics & Business Intelligence*

Mentor : Lisna Sulistiawati

Kelompok : 2

Anggota :

1. Arya Rizky Tri Putra
2. Astifa Deswanti Talosi
3. Enny Ayu Sekar Arum
4. Vinsens Aji Pamungkas

PT Mitra Talenta Grup (Celerates)
Jakarta
2024

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Allah SWT karena atas rahmat dan karunia-Nya, kami dapat menyelesaikan laporan ini tepat pada waktunya. Laporan ini disusun sebagai bagian dari *Final project Data Analytics & Business Intelligence*. Adapun judul dari laporan ini adalah “*Analisis Operasional Toko DVD Rental dan Preferensi Film Berdasarkan Genre*”.

Kami mengucapkan terima kasih yang sebesar-besarnya kepada Kak Putra Gema Nusa selaku *Trainer Data Analytics & Business Intelligence* dan Kak Lisna Sulistiawati selaku *Mentor* Kelompok 2 atas bimbingan, dukungan, serta arahan yang telah diberikan selama program Studi Independen dan dalam proses penyelesaian *final project*. Ucapan terima kasih juga kami sampaikan kepada semua pihak yang telah memberikan bantuan, dukungan, dan kontribusi, baik secara langsung maupun tidak langsung, sehingga laporan ini dapat diselesaikan dengan baik.

Kami menyadari bahwa laporan ini masih memiliki kekurangan karena keterbatasan pengetahuan dan pengalaman kami. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari para pembaca guna menyempurnakan laporan ini di masa mendatang.

Akhir kata, semoga laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pembaca.

Jakarta, 17 Desember 2024

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	iv
DAFTAR TABEL.....	vi
DAFTAR LAMPIRAN.....	vii
ABSTRAK.....	viii
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Tujuan <i>Project</i>	2
1.3. Manfaat <i>Project</i>	2
1.4. Lingkup <i>Project</i>	2
BAB 2 METODOLOGI PROJECT.....	3
2.1. Deskripsi <i>Database</i>	3
2.2. Tahapan <i>Project</i>	5
2.3. <i>Tools</i> dan Teknologi.....	11
BAB 3 IMPLEMENTASI DAN HASIL.....	14
3.1. <i>Input Database</i>	14
3.2. Pembuatan <i>Staging Area</i>	15
3.3. Desain <i>Data Warehouse</i>	18
3.4. Pembuatan <i>Datamart</i>	27
3.5. Visualisasi dan Analisis <i>Dashboard</i>	29
BAB 4 PENUTUP.....	43
4.1. Kesimpulan.....	43
4.2. Saran.....	43
DAFTAR PUSTAKA.....	45
LAMPIRAN.....	46

DAFTAR GAMBAR

Gambar 2. 1 Desain Layout <i>Dashboard</i> 1	9
Gambar 2. 2 Desain <i>Layout Dashboard</i> 2.....	10
Gambar 3. 1 <i>Input Database</i> (1)	14
Gambar 3. 2 <i>Input Database</i> (2)	14
Gambar 3. 3 <i>Input Database</i> (3)	14
Gambar 3. 4 <i>Input Database</i> (4)	15
Gambar 3. 5 <i>Input Database</i> (5)	15
Gambar 3. 6 <i>Database staging_dvdrental</i>	15
Gambar 3. 7 <i>Schemas staging</i>	16
Gambar 3. 8 <i>Schemas log</i>	16
Gambar 3. 9 <i>tf_list_tabel</i>	16
Gambar 3. 10 <i>Preview Data tf_list_tabel</i>	17
Gambar 3. 11 <i>tf_read_data</i>	17
Gambar 3. 12 <i>tf_log_tabel_masuk</i>	17
Gambar 3. 13 <i>job_execute_data</i>	18
Gambar 3. 14 <i>main_job</i>	18
Gambar 3. 15 <i>Database dwh_dvdrental</i>	18
Gambar 3. 16 <i>Schemas dwh</i>	19
Gambar 3. 17 <i>dim_category_film</i>	19
Gambar 3. 18 <i>Preview Data dim_category_film</i>	20
Gambar 3. 19 <i>dim_customer</i>	20
Gambar 3. 20 <i>Preview Data dim_customer</i>	21
Gambar 3. 21 <i>dim_film</i>	21
Gambar 3. 22 <i>Preview Data dim_film</i> (1).....	22
Gambar 3. 23 <i>Preview Data dim_film</i> (2).....	22
Gambar 3. 24 <i>dim_inventory</i>	23
Gambar 3. 25 <i>Preview Data dim_inventory</i>	23
Gambar 3. 26 <i>dim_rental</i>	24
Gambar 3. 27 <i>Preview Data dim_rental</i>	24
Gambar 3. 28 <i>dim_staff</i>	25
Gambar 3. 29 <i>Preview Data dim_staff</i>	25
Gambar 3. 30 <i>dim_store</i>	25
Gambar 3. 31 <i>Preview Data dim_store</i>	26
Gambar 3. 32 <i>fact_payment</i>	26
Gambar 3. 33 <i>Preview Data fact_payment</i>	26
Gambar 3. 34 <i>Schemas datamart</i>	27
Gambar 3. 35 <i>datamart_dvdrental</i>	27
Gambar 3. 36 <i>Preview Data datamart_dvdrental</i> (1)	28
Gambar 3. 37 <i>Preview Data datamart_dvdrental</i> (2)	28
Gambar 3. 38 <i>Preview Data datamart_dvdrental</i> (3)	28
Gambar 3. 39 <i>Preview Data datamart_dvdrental</i> (4)	29
Gambar 3. 40 <i>Filter Date</i>	29

Gambar 3. 41 Parameter Select Store	30
Gambar 3. 42 Calculation Field Store Select.....	30
Gambar 3. 43 Total Revenue Toko 28 MySQL Boulevard.....	31
Gambar 3. 44 Total Revenue Toko 47 MySakila Drive.....	31
Gambar 3. 45 Total Active Customers Toko 28 MySQL Boulevard	32
Gambar 3. 46 Total Active Customers Toko 47 MySakila Drive	32
Gambar 3. 47 Total Staff.....	33
Gambar 3. 48 Total Film Toko 28 MySQL Boulevard	33
Gambar 3. 49 Total Film Toko 47 MySakila Drive	34
Gambar 3. 50 Total Genre.....	34
Gambar 3. 51 Rental Trend Toko 28 MySQL Boulevard	35
Gambar 3. 52 Rental Trend Toko 47 MySakila Drive	35
Gambar 3. 53 Top Customer by Rental Toko 28 MySQL Boulevard.....	36
Gambar 3. 54 Top Customer by Rental Toko 47 MySakila Drive.....	36
Gambar 3. 55 Top Film by Rental Duration Toko 28 MySQL Boulevard.....	37
Gambar 3. 56 Top Film by Rental Duration Toko 47 MySakila Drive.....	37
Gambar 3. 57 Revenue by Genre Toko 28 MySQL Boulevard	38
Gambar 3. 58 Revenue by Genre Toko 47 MySakila Drive	38
Gambar 3. 59 Top Film by Rental Rate Toko 28 MySQL Boulevard	39
Gambar 3. 60 Top Film by Rental Rate Toko 47 MySakila Drive	39
Gambar 3. 61 Revenue by Film Toko 28 MySQL Boulevard	40
Gambar 3. 62 Revenue by Film Toko 47 MySakila Drive	40
Gambar 3. 63 <i>Dashboard</i> DVD Rental Summary Toko 28 MySQL Boulevard	41
Gambar 3. 64 <i>Dashboard</i> DVD Rental Summary Toko 47 MySakila Drive	41
Gambar 3. 65 <i>Dashboard</i> Film Details by Genre	42

DAFTAR TABEL

Tabel 2. 1 Struktur <i>Database</i> yang Digunakan.....	3
Tabel 2. 2 Hubungan Kardinalitas Antar Tabel.....	5
Tabel 2. 3 Tabel Dimensi yang Digunakan	6
Tabel 2. 4 Tabel Fakta yang Digunakan.....	7
Tabel 2. 5 Tabel <i>Datamart</i> yang Digunakan	8
Tabel 2. 6 Informasi yang Ditampilkan pada <i>Dashboard</i>	10

DAFTAR LAMPIRAN

Lampiran 1 <i>Script Query Database staging_dvdrental</i>	46
Lampiran 2 <i>Script Query Schemas log</i>	54

ABSTRAK

Arya Rizky Tri Putra, Astifa Deswanti Talosi, Enny Ayu Sekar Arum, Vinsens Aji Pamungkas

ANALISIS OPERASIONAL TOKO DVDRENTAL DAN PREFERENSI FILM BERDASARKAN *GENRE*

Kata Kunci: Analitik data berbasis *database*, preferensi pelanggan, efisiensi operasional
(viii + 45 + Lampiran)

Industri persewaan DVD terus memberikan pengalaman menonton yang unik, terutama bagi pelanggan yang mencari koleksi film klasik atau *genre* tertentu yang sulit ditemukan di *platform digital*. Toko DVD Rental menghadapi tantangan dalam mengelola operasional dua toko yang berlokasi di negara berbeda, masing-masing dengan preferensi pelanggan yang beragam. Untuk menjawab tantangan ini, *project* ini melakukan analisis data berbasis *database* dvdrental dengan tujuan mengevaluasi kinerja operasional toko dan memahami pola preferensi pelanggan. Proses analisis melibatkan tahapan yang sistematis, mulai dari pengolahan data melalui *staging area*, pembangunan *data warehouse*, hingga perancangan *dashboard* visualisasi yang intuitif. Data yang dianalisis mencakup informasi terkait *genre* film populer, pola perilaku pelanggan, dan efisiensi operasional toko. Hasil analisis ini memberikan wawasan strategis untuk mendukung pengelolaan stok yang lebih efektif, penyusunan koleksi film yang relevan, dan peningkatan layanan berbasis kebutuhan lokal. Dari sisi teknis, *project* ini berhasil mengintegrasikan data menggunakan berbagai *tools*, seperti PostgreSQL, DBeaver, Pentaho, dan Tableau, untuk menciptakan solusi analitik data *end-to-end*. Dari sisi bisnis, wawasan yang dihasilkan mendukung perencanaan strategi pemasaran berbasis data, optimalisasi inventaris, dan peningkatan pengalaman pelanggan. Dengan pendekatan berbasis data, toko DVD Rental dapat memanfaatkan hasil analisis ini untuk memperkuat daya saing bisnis di tengah dinamika pasar hiburan modern dan kebutuhan pelanggan yang terus berkembang.

BAB 1 PENDAHULUAN

1.1. Latar Belakang

Industri persewaan DVD telah mengalami perkembangan pesat sejak awal kemunculannya sebagai salah satu bentuk hiburan yang terjangkau dan mudah diakses. Meskipun saat ini *platform streaming digital* semakin mendominasi pasar hiburan, bisnis persewaan DVD tetap memiliki peran dalam menyediakan pengalaman menonton yang berbeda, terutama untuk pelanggan yang mencari koleksi film klasik atau *genre* tertentu yang mungkin tidak tersedia di *platform digital*. Dalam konteks ini, memahami preferensi pelanggan menjadi kunci untuk mempertahankan daya saing di pasar yang semakin ketat. Selain itu, efisiensi operasional menjadi faktor penting dalam memastikan keberlanjutan usaha, terutama bagi toko-toko yang beroperasi di lokasi geografis yang berbeda.

Toko DVD Rental menghadapi tantangan unik dalam mengelola operasional dua toko yang berlokasi di negara berbeda. Kedua toko ini melayani pelanggan dari berbagai latar belakang budaya dan geografis, yang memengaruhi preferensi mereka terhadap *genre* film. Dengan koleksi film yang mencakup berbagai *genre*, perusahaan perlu memahami pola permintaan pelanggan untuk memastikan bahwa inventaris dikelola secara optimal. Hal ini mencakup pengelolaan stok film populer di setiap lokasi, merancang strategi pemasaran yang sesuai, dan meningkatkan layanan untuk memenuhi kebutuhan pasar lokal.

Analitik data berbasis *database* menjadi salah satu pendekatan yang dapat membantu perusahaan dalam menjawab tantangan tersebut. Studi menunjukkan bahwa analitik data mampu memberikan wawasan strategis untuk meningkatkan efisiensi operasional, mengidentifikasi preferensi pelanggan, dan mengoptimalkan proses bisnis (Colosimo, 2018).

Dalam hal ini, *database dvdrental* menjadi alat penting untuk mengevaluasi performa operasional toko dan memahami pola perilaku pelanggan. Setiap tabel dalam *database* ini menyediakan informasi mendalam yang dapat digunakan untuk mendukung keputusan bisnis berbasis data. *Database* ini relevan untuk tujuan analisis karena mencakup data yang penting untuk memahami aspek operasional bisnis secara keseluruhan. Informasi dalam *database* ini memungkinkan analisis terhadap berbagai elemen kunci, seperti pola perilaku pelanggan, preferensi *genre* film, serta kinerja operasional toko di dua lokasi yang berbeda.

Dengan menganalisis data ini, perusahaan dapat memperoleh wawasan untuk meningkatkan kinerja operasional dan memahami pola preferensi pelanggan dengan lebih baik. Hal ini mencakup rekomendasi strategis seperti meningkatkan stok *genre* populer, menyesuaikan layanan dengan kebutuhan pelanggan lokal, serta mengidentifikasi peluang untuk meningkatkan promosi berbasis data.

Berdasarkan latar belakang di atas, *project* ini bertujuan untuk mengevaluasi kinerja operasional kedua toko DVD Rental dan menganalisis preferensi pelanggan terhadap *genre* film. Hasil analisis ini diharapkan dapat membantu perusahaan merancang strategi berbasis data untuk mengoptimalkan operasional, meningkatkan pengalaman pelanggan, dan memperkuat daya saing bisnis di tengah tantangan industri hiburan modern.

1.2. Tujuan *Project*

- **Sisi Teknis**
 1. Menguasai proses analitik data berbasis *database* melalui tahapan yang sistematis.
 2. Mengembangkan kemampuan dalam pengolahan data dan pembuatan visualisasi yang mendukung pengambilan keputusan.
 3. Mengaplikasikan keterampilan teknis dalam menyelesaikan permasalahan nyata menggunakan data.
- **Sisi Bisnis**
 1. Membantu toko DVD Rental mengevaluasi kinerja operasional di dua lokasi yang berbeda.
 2. Memahami pola preferensi pelanggan untuk menyusun strategi penyediaan koleksi film yang relevan.
 3. Memberikan rekomendasi berbasis data untuk meningkatkan efisiensi, layanan, dan strategi pemasaran.

1.3. Manfaat *Project*

- **Sisi Teknis**
 1. Menyelesaikan *final project* sebagai bentuk pembelajaran dan evaluasi kemampuan analisis data.
 2. Memahami proses pengolahan data, termasuk *staging area*, pembuatan *data warehouse*, *datamart*, dan visualisasi data.
 3. Mempelajari dan menggunakan *tools* baru yang mendukung analisis data secara efisien.
- **Sisi Bisnis**
 1. Memberikan wawasan untuk mengoptimalkan operasional toko DVD Rental berdasarkan analisis data.
 2. Memahami preferensi pelanggan untuk meningkatkan pengalaman dan kepuasan mereka.
 3. Merancang strategi pemasaran berbasis data untuk memperkuat daya saing di pasar hiburan.

1.4. Lingkup *Project*

1. *Project* ini terbatas pada analisis data dari *database* dvdrental.
2. *Project* ini berfokus pada analisis toko DVD Rental dan preferensi film berdasarkan *genre*.
3. *Staging area* akan dibuat menggunakan Dbeaver dan Pentaho.
4. *Data warehouse* akan dibangun menggunakan Pentaho.
5. Visualisasi data akan dirancang dalam dua tahap. Pertama, *layout dashboard* akan dirancang menggunakan Figma. Terakhir, implementasi *dashboard* akan dilakukan menggunakan Tableau.

BAB 2 METODOLOGI PROJECT

2.1. Deskripsi *Database*

Database yang digunakan dalam *project* ini adalah dvdrental. Dari 15 tabel yang tersedia dalam *database* ini, dipilih 11 tabel yang relevan dengan kebutuhan analisis dan visualisasi data yang akan dilakukan. Struktur *database* yang digunakan dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Struktur Database yang Digunakan

No	Tabel yang Digunakan	<i>Field</i> Asli	<i>Field</i> yang Digunakan	Alasan Perubahan
1	film	film_id	film_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		title	film_title	Membuat nama lebih deskriptif
		description	film_desc	Membuat nama lebih deskriptif
		release_year	film_release_year	Membuat nama lebih deskriptif
		rental_duration	film_rental_duration	Membuat nama lebih deskriptif
		rental_rate	film_rental_rate	Membuat nama lebih deskriptif
		special_features	film_special_features	Membuat nama lebih deskriptif
2	film_category	category_id	category_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		film_id	film_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>
		name	category_film_name	Membuat nama lebih deskriptif
3	store	store_id	store_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		address_id	address_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>
4	address	address_id	address_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		city_id	city_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>

		address	store_address	Membuat nama lebih deskriptif
5	city	city_id	city_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		country_id	country_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>
		city	store_city	Membuat nama lebih deskriptif
6	country	country_id	country_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		country	store_country	Membuat nama lebih deskriptif
7	staff	staff_id	staff_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		store_id	store_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>
		first_name + last_name	staff_name	Menggabungkan first_name dan last_name menjadi satu <i>field</i> untuk menyederhanakan data
8	customer	customer_id	customer_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		store_id	store_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>
		first_name + last_name	customer_name	Menggabungkan first_name dan last_name menjadi satu <i>field</i> untuk menyederhanakan data
		active	customer_active	Membuat nama lebih deskriptif
9	inventory	inventory_id	inventory_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		film_id	film_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>

		store_id	store_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>
10	rental	rental_id	rental_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		staff_id	staff_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>
		customer_id	customer_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>
		inventory_id	inventory_id	Tidak diubah, digunakan sebagai <i>Foreign Key</i>
		rental_date	rental_date	Tidak diubah
		return_date	rental_return_date	Membuat nama lebih deskriptif
11	payment	payment_id	payment_id	Tidak diubah, digunakan sebagai <i>Primary Key</i>
		amount	payment_amount	Membuat nama lebih deskriptif
		payment_date	payment_date	Tidak diubah

Berikut adalah hubungan kardinalitas antar tabel yang digunakan dalam *project* ini, seperti yang ditampilkan pada Tabel 2.2.

Tabel 2. 2 Hubungan Kardinalitas Antar Tabel

No	Relasi Antar Tabel	Hubungan Kardinalitas
1	film_category → film	1:1
2	store → address	1:1
3	address → city	1:1
4	city → country	1:1
5	staff → store	1:1
6	customer → store	1:M
7	inventory → film	1:M
8	inventory → store	1:M
9	rental → staff	M:1
10	rental → customer	1:M
11	rental → inventory	1:1

2.2. Tahapan *Project*

Dalam *project* ini, terdapat beberapa tahapan *project* yang harus dilakukan secara sistematis untuk memastikan proses pengolahan data berjalan dengan baik. Tahapan

ini mencakup langkah-langkah penting dalam pengelolaan data hingga menghasilkan *output* yang dapat digunakan untuk analisis lebih lanjut.

- **Input Database ke RDBMS**

Langkah awal dimulai dengan mengimpor *database* dvdrental ke dalam PostgreSQL sebagai RDBMS yang digunakan. Proses ini bertujuan untuk menyimpan data mentah dalam sistem basis data yang terstruktur, sehingga memudahkan pengelolaan dan pengolahan data di tahap selanjutnya.

- **Pembuatan Staging Area**

Data dari PostgreSQL dipindahkan ke *staging area*, tempat sementara untuk memproses data sebelum masuk ke *data warehouse*. Di tahap ini, dilakukan transformasi data menggunakan DBeaver dan Pentaho, termasuk langkah pembersihan (*data cleaning*) serta normalisasi data agar sesuai dengan kebutuhan struktur *database*.

- **Desain Data Warehouse**

Data warehouse dirancang menggunakan Pentaho dengan tujuan untuk menyimpan data yang sudah terintegrasi dan siap dianalisis. Dalam Desain ini, tabel dimensi dibuat untuk menyimpan informasi deskriptif yang memberikan konteks pada data, sedangkan tabel fakta digunakan untuk menyimpan data transaksi atau kejadian yang dapat diukur dan dianalisis. Pembuatan tabel dimensi dan fakta ini telah disesuaikan dengan kebutuhan analisis dan visualisasi data pada *project* ini, sehingga dibuat tujuh tabel dimensi serta satu tabel fakta, seperti yang ditampilkan pada Tabel 2.3 dan Tabel 2.4.

Tabel 2. 3 Tabel Dimensi yang Digunakan

No	Tabel Dimensi	Field yang Digunakan	Keterangan
1	dim_category_film	sk_category_film	Digunakan sebagai <i>Primary Key</i>
		film_id	
		category_id	
		category_film_name	
2	dim_customer	sk_customer	Digunakan sebagai <i>Primary Key</i>
		customer_id	
		customer_name	
		store_id	
		customer_active	
		version	
		date_from	
3	dim_film	sk_film	Digunakan sebagai <i>Primary Key</i>
		film_id	
		film_title	
		film_desc	
		film_release_year	

		film_rental_duration	
		film_rental_rate	
		film_special_features	
4	dim_inventory	sk_inventory	Digunakan sebagai <i>Primary Key</i>
		inventory_id	
		film_id	
		store_id	
5	dim_rental	sk_rental	Digunakan sebagai <i>Primary Key</i>
		rental_id	
		customer_id	
		rental_date	
		rental_return_date	
		staff_id	
		inventory_id	
6	dim_staff	sk_staff	Digunakan sebagai <i>Primary Key</i>
		staff_id	
		staff_name	
		store_id	
		version	
		date_from	
		date_to	
7	dim_store	sk_store	Digunakan sebagai <i>Primary Key</i>
		store_id	
		address_id	
		store_address	
		city_id	
		store_city	
		country_id	
		store_country	

Tabel 2. 4 Tabel Fakta yang Digunakan

No	Tabel Fakta	<i>Field</i> yang Digunakan	Keterangan
1	fact_payment	payment_id	Digunakan sebagai <i>Primary Key</i>
		sk_rental	Digunakan sebagai <i>Foreign Key</i>
		sk_inventory	Digunakan sebagai <i>Foreign Key</i>
		sk_film	Digunakan sebagai <i>Foreign Key</i>
		sk_category_film	Digunakan sebagai <i>Foreign Key</i>

		sk_staff	Digunakan sebagai <i>Foreign Key</i>
		sk_store	Digunakan sebagai <i>Foreign Key</i>
		sk_customer	Digunakan sebagai <i>Foreign Key</i>
		payment_amount	
		payment_date	

- **Pembuatan *Datamart***

Setelah *data warehouse* selesai, *datamart* dibuat menggunakan Pentaho dengan menggabungkan tabel fakta dan tabel dimensi. *Field* yang digunakan telah disesuaikan untuk mendukung tujuan utama, yaitu analisis operasional toko DVD Rental dan preferensi film berdasarkan *genre*, seperti yang ditampilkan pada Tabel 2.5.

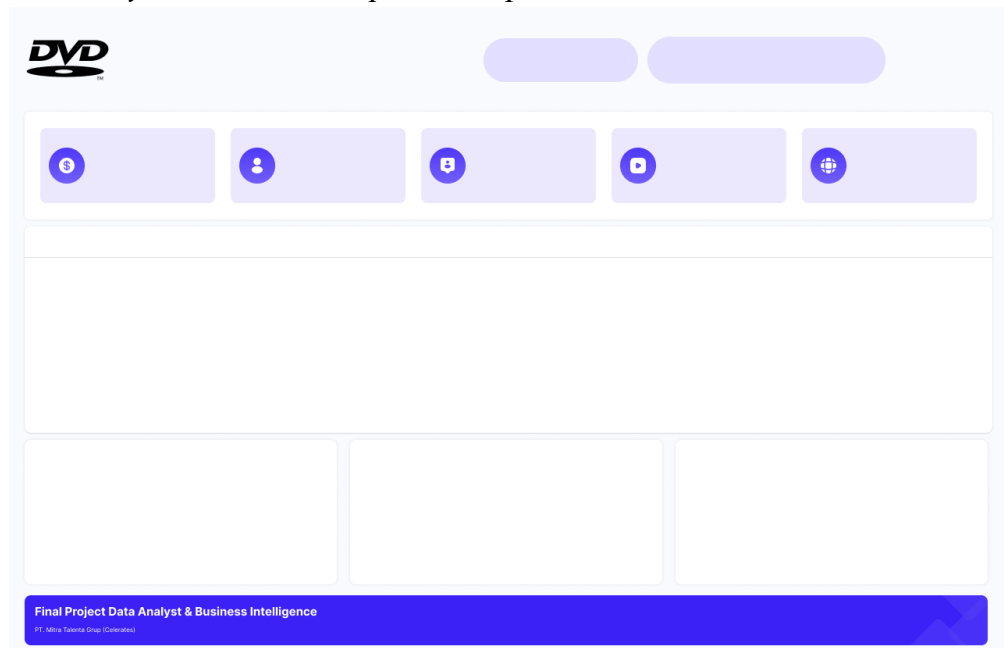
Tabel 2. 5 Tabel *Datamart* yang Digunakan

No	<i>Datamart</i>	<i>Field</i> yang Digunakan	Keterangan
1	datamart_dvdrental	payment_id	Digunakan sebagai <i>Primary Key</i>
		payment_amount	
		payment_date	
		sk_rental	Digunakan sebagai <i>Foreign Key</i>
		rental_date	
		rental_return_date	
		sk_inventory	Digunakan sebagai <i>Foreign Key</i>
		sk_film	Digunakan sebagai <i>Foreign Key</i>
		film_title	
		film_desc	
		film_release_year	
		film_rental_duration	
		film_rental_rate	
		film_special_features	
		sk_category_film	Digunakan sebagai <i>Foreign Key</i>
		category_film_name	
		sk_staff	Digunakan sebagai <i>Foreign Key</i>
		staff_name	
		sk_store	Digunakan sebagai <i>Foreign Key</i>
		store_address	
		store_city	

		store_country	
		sk_customer	Digunakan sebagai <i>Foreign Key</i>
		customer_name	
		customer_active	

- **Pembuatan *Dashboard***

Sebelum menggunakan Tableau, langkah awal dilakukan dengan merancang *layout dashboard* di Figma. Proses ini mencakup penentuan tata letak elemen-elemen seperti grafik, filter, informasi, serta pemilihan skema warna yang konsisten dan profesional. Desain *layout* ini bertujuan untuk memberikan gambaran visual awal *dashboard* yang akan dibuat, sehingga memastikan semua elemen data tersusun secara rapi dan mudah dipahami. Desain *layout dashboard* dapat dilihat pada Gambar 2.1 dan Gambar 2.2.



Gambar 2. 1 Desain *Layout Dashboard 1*



Gambar 2. 2 Desain *Layout Dashboard 2*

Setelah desain selesai, Tableau digunakan untuk membuat *dashboard* interaktif. Akan dibuat 2 *dashboard*, yang akan menyajikan informasi penting mengenai operasional toko DVD Rental dan preferensi film berdasarkan *genre*, seperti yang ditampilkan pada Tabel 2.6.

Tabel 2. 6 Informasi yang Ditampilkan pada *Dashboard*

<i>Dashboard</i>	<i>Judul Dashboard</i>	Informasi yang Ditampilkan	Jenis Visualisasi
1	DVD Rental Summary	Total Revenue	<i>KPI Card</i>
		Total Customer	<i>KPI Card</i>
		Total Staff	<i>KPI Card</i>
		Total Film	<i>KPI Card</i>
		Total Genre	<i>KPI Card</i>
		Rental Trend	<i>Area Chart</i>
		Top Customer by Rental	<i>Vertical Bar Chart</i>
		Top Film by Rental Duration	<i>Vertical Bar Chart</i>
		Revenue by Genre	<i>Donut Chart</i>
2	Film Details by Genre	Total Revenue	<i>KPI Card</i>
		Total Customer	<i>KPI Card</i>
		Total Staff	<i>KPI Card</i>
		Total Film	<i>KPI Card</i>
		Total Genre	<i>KPI Card</i>
		Top Film by Rental Rate	<i>Vertical Bar Chart</i>
		Revenue by Film	<i>Donut Chart</i>

2.3. Tools dan Teknologi

Dalam *project* ini, berbagai *tools* dan teknologi digunakan untuk mendukung setiap tahapan proses. Pemilihan *tools* ini didasarkan pada fungsionalitasnya dalam memproses data secara efisien dan menghasilkan *output* yang sesuai dengan tujuan analisis.

- **PostgreSQL**

PostgreSQL adalah sistem manajemen basis data relasional (RDBMS) sumber terbuka yang kuat dan andal. PostgreSQL dirancang untuk mendukung transaksi yang mematuhi prinsip ACID (*Atomicity, Consistency, Isolation, Durability*), menjadikannya sangat andal untuk aplikasi yang memerlukan integritas data. Selain itu, PostgreSQL memiliki fitur ekstensibilitas yang memungkinkan pengguna menambahkan fungsi khusus, tipe data baru, dan prosedur. PostgreSQL juga mendukung berbagai jenis data, termasuk JSON, XML, dan data spasial, serta mematuhi standar SQL, menjadikannya fleksibel untuk beragam kebutuhan aplikasi modern (Sri Wahyuningsih & Junianto, 2020).

PostgreSQL dipilih untuk digunakan dalam *project* ini karena keandalannya dalam mengelola data kompleks dengan performa tinggi. PostgreSQL memiliki kemampuan transaksi yang mematuhi ACID, yang memastikan integritas data selama proses ETL (*Extract, Transform, Load*) dari *staging* ke *data warehouse*. Selain itu, PostgreSQL mendukung ekstensibilitas dan berbagai jenis data, sehingga cocok untuk mengelola *database* dengan struktur yang beragam seperti pada *database* dvdrental.

- **Dbeaver**

DBeaver adalah perangkat lunak antarmuka basis data lintas platform bersifat *open-source* yang dirancang untuk mempermudah pengelolaan, analisis, dan visualisasi berbagai sistem basis data, termasuk PostgreSQL, MySQL, Oracle, dan lainnya. Dengan antarmuka yang intuitif, DBeaver menyediakan fitur seperti *editor* SQL, visualisasi data, pembuatan diagram basis data, serta kemampuan untuk bekerja dengan berbagai jenis data secara efisien. Selain itu, DBeaver mendukung integrasi dengan sistem kontrol versi dan memiliki ekosistem ekstensi untuk mendukung kebutuhan manajemen data yang kompleks (Kamaliya et al., 2024).

Dbeaver dipilih untuk digunakan dalam *project* ini karena fleksibilitas dan kemudahan penggunaannya dalam mengelola *database*, terutama PostgreSQL yang digunakan dalam proyek ini. Antarmuka intuitif DBeaver memungkinkan untuk menulis, menguji, dan menjalankan *query* SQL dengan efisien selama proses ETL dan analisis data. Fitur visualisasi bawaan sangat membantu dalam memahami struktur data serta memvalidasi hasil transformasi sebelum dimasukkan ke dalam *data warehouse*.

- **Pentaho Data Integration (PDI)**

Pentaho adalah platform perangkat lunak sumber terbuka yang dirancang untuk mendukung solusi *Business Intelligence* (BI) dan integrasi

data. Salah satu komponen utamanya, Pentaho Data Integration (PDI), menyediakan alat untuk proses ETL (*Extract, Transform, Load*) yang memungkinkan pengguna memindahkan, mengolah, dan mengorganisasi data dari berbagai sumber ke dalam format yang siap dianalisis. Dengan kemampuan untuk menangani data terstruktur hingga semi-terstruktur, Pentaho menjadi alat yang fleksibel untuk membangun *data warehouse*, *datamart*, hingga visualisasi data yang interaktif (Saraswati & Martarini, 2020).

Pentaho dipilih untuk digunakan dalam *project* ini karena Pentaho memiliki kemampuan yang sangat cocok untuk kebutuhan *project*, yaitu membangun proses *staging area*, *data warehouse*, dan *datamart*. Pentaho menawarkan fleksibilitas dengan antarmuka *drag-and-drop* yang mempermudah perancangan alur ETL tanpa harus bergantung sepenuhnya pada kode. Selain itu, kemampuan Pentaho untuk menangani data dalam jumlah besar dan mendukung berbagai format memastikan proses integrasi data berjalan lancar.

- **Figma**

Figma adalah alat desain berbasis web yang dirancang untuk kolaborasi *real-time* dalam merancang antarmuka pengguna (UI) dan prototipe interaktif. Dengan fitur-fitur seperti *drag-and-drop*, pengelolaan komponen, serta integrasi dengan berbagai alat desain lainnya, Figma memungkinkan *desainer* untuk bekerja secara efisien dalam lingkungan yang terpusat. Karena berbasis cloud, Figma mempermudah kolaborasi lintas tim tanpa perlu menginstal perangkat lunak tambahan, sekaligus memastikan semua anggota tim selalu bekerja dengan versi desain terbaru (Sudjiran et al., 2023).

Figma dipilih untuk digunakan dalam *project* ini karena Figma menawarkan fleksibilitas tinggi dan efisiensi dalam mendesain *layout* interaktif. Dengan Figma, elemen visual dapat diatur dengan mudah untuk menciptakan desain yang rapi dan intuitif.

- **Tableau**

Tableau adalah perangkat lunak visualisasi data yang dirancang untuk membantu pengguna mengolah dan menyajikan data dalam bentuk laporan, dasbor, serta analisis interaktif yang intuitif. Dengan antarmuka berbasis *drag-and-drop*, Tableau memungkinkan pengguna untuk mengeksplorasi data dari berbagai sumber, baik itu *database*, *file*, maupun layanan cloud, dan mengubahnya menjadi visualisasi yang informatif. Kemampuan Tableau untuk menangani data besar, mendukung kolaborasi tim, serta integrasinya dengan alat analitik lain menjadikannya pilihan populer di dunia *Business Intelligence* (BI) untuk mempercepat pengambilan keputusan berbasis data (Marvaro & Sefina Samosir, 2021).

Tableau dipilih untuk digunakan dalam *project* ini karena kemampuannya yang sangat baik dalam mengubah data kompleks menjadi visualisasi yang intuitif, terutama untuk kebutuhan spesifik *project* ini, yaitu menyoroti ringkasan kinerja toko DVD Rental dan menganalisis detail film

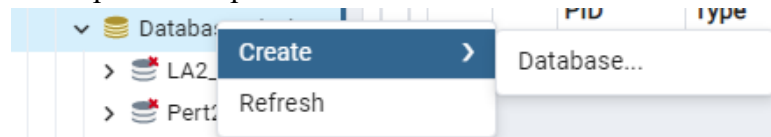
berdasarkan *genre*. Dengan fitur *drag-and-drop*, dapat dengan mudah membuat grafik interaktif yang menampilkan metrik penting.

BAB 3

IMPLEMENTASI DAN HASIL

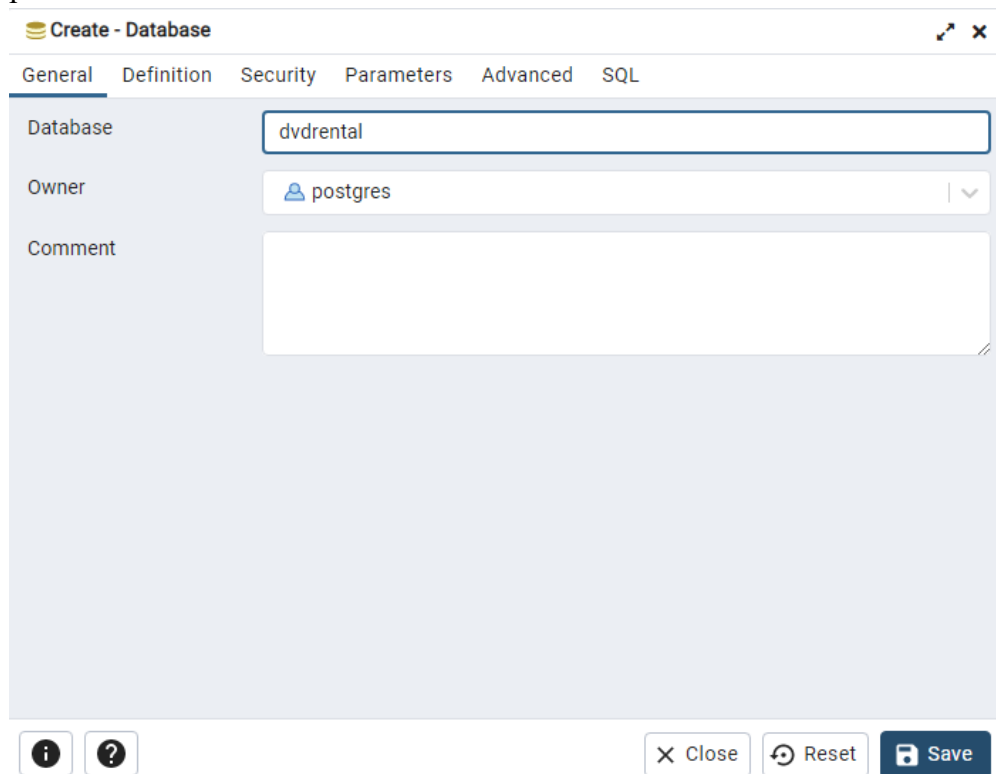
3.1. *Input Database*

1. Pertama, pada PostgreSQL klik kanan "Database", lalu pilih "Create" dan klik "Database". Dapat dilihat pada Gambar 3.1.



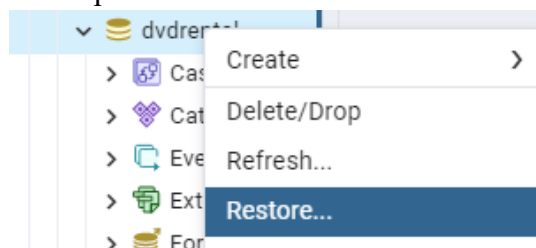
Gambar 3. 1 *Input Database* (1)

2. Selanjutnya, *input* dvdrental pada "Database" dan klik "Save". Dapat dilihat pada Gambar 3.2.



Gambar 3. 2 *Input Database* (2)

3. Setelah itu, klik kanan *database* dvdrental yang telah dibuat, lalu klik "Restore". Dapat dilihat pada Gambar 3.3.



Gambar 3. 3 *Input Database* (3)

4. Kemudian, jika sudah memasukkan path tempat dimana *database* dvdrental diunduh, klik "Restore". Dapat dilihat pada Gambar 3.5.

Restore (Database: dvdrental)

General Data/Objects Options

Format: Custom or tar

Filename: C:\Users\USER\Downloads\dvdrental.tar

Number of jobs:

Role name: Select an item...

Close Reset Restore

Gambar 3. 4 Input Database (4)

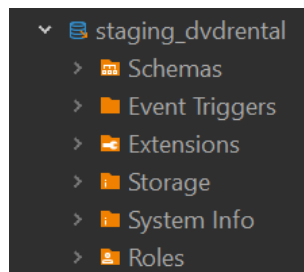
5. Terakhir, *database* dvdrental telah berhasil diimpor ke dalam PostgreSQL. Dapat dilihat pada Gambar 3.6.

		PID	Type	Server	Object	Start Time	Status	Time Taken (sec)
<input type="checkbox"/>		3904	Restore	PostgreSQL 15 (localhost:543...	dvdrental	14/12/2024, 14:42:20	Finished	8.7

Gambar 3. 5 Input Database (5)

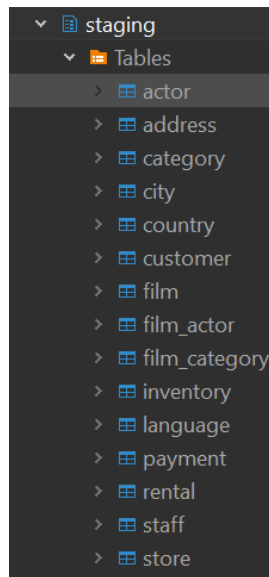
3.2. Pembuatan *Staging Area*

1. Pertama, buat *database* baru pada Dbeaver dengan nama *staging_dvdrental*. Dapat dilihat pada Gambar 3.6.



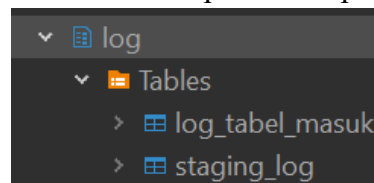
Gambar 3. 6 Database staging_dvdrental

2. Selanjutnya, buat *schemas* staging dan pindahkan kerangka data dari *database* dvdrental ke *database* staging_dvdrental. Dapat dilihat pada Gambar 3.7.



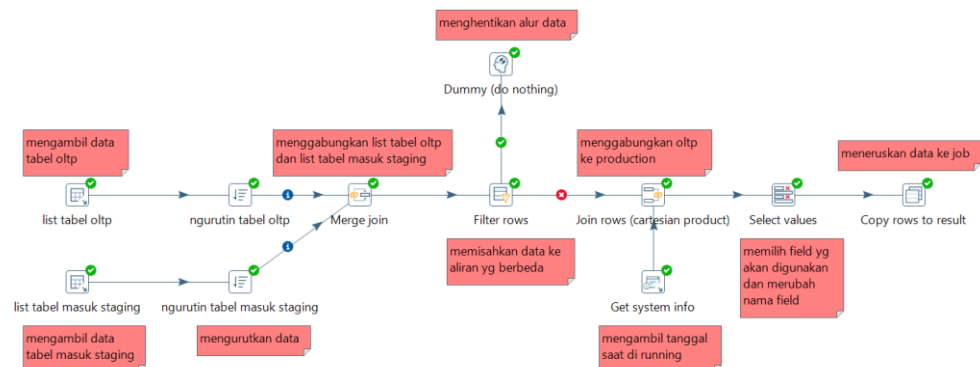
Gambar 3. 7 Schemas staging

3. Lalu, buat *schemas* log dan buat tabel staging_log dan log_tabel_masuk dengan *field* yang sudah ditentukan. Dapat dilihat pada Gambar 3.8.



Gambar 3. 8 Schemas log

4. Kemudian, pada Pentaho buat transformasi dengan nama tf_list_tabel. Proses dimulai dengan mengambil dan mengurutkan data dari tabel OLTP dan tabel masuk staging, lalu digabungkan menggunakan "Merge Join". Data difilter berdasarkan kondisi "table_name = table_name_1". Jika benar (true), data diteruskan ke "Dummy" dan dihentikan. Jika salah (false), data dilanjutkan ke "Join Rows" dan ditambahkan informasi tanggal saat proses berjalan. Selanjutnya, *field* dipilih dan namanya disesuaikan di "Select Values", dan hasilnya diteruskan ke job berikutnya. Dapat dilihat pada Gambar 3.9, dengan *Preview* data pada Gambar 3.10.

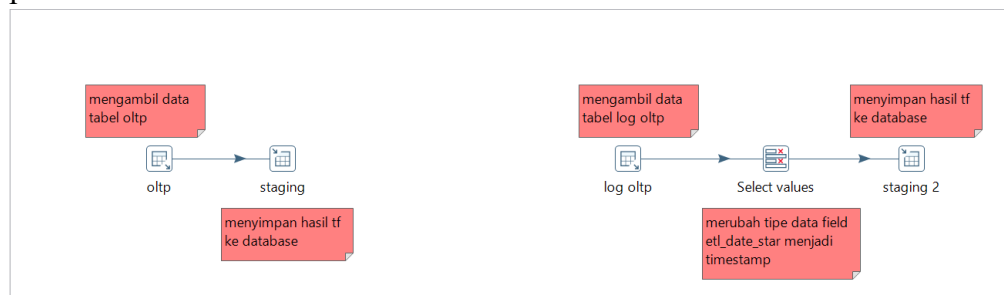


Gambar 3. 9 tf_list_tabel

Logging			Execution His
<input checked="" type="radio"/> First rows			<input type="radio"/> Last rows
#	mulai	table_name	
1			

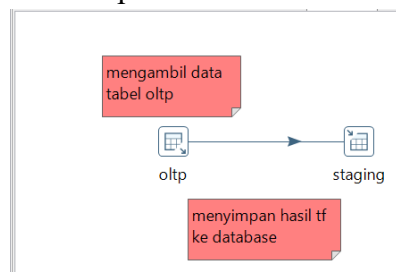
Gambar 3. 10 Preview Data tf_list_tabel

- Setelah itu, buat transformasi dengan nama tf_read_data. Proses dimulai dengan mengambil data dari tabel OLTP dan menyimpannya ke dalam tabel di *schema staging*. Selanjutnya, data dari tabel log OLTP diambil, diproses dengan "Select Values" untuk mengubah tipe data *field etl_date_start* menjadi timestamp, dan hasilnya disimpan ke dalam tabel staging_log. Dapat dilihat pada Gambar 3.11.



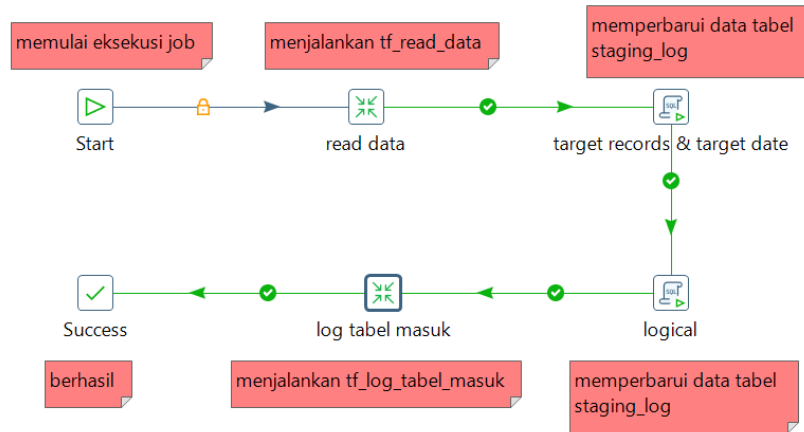
Gambar 3. 11 tf_read_data

- Lalu, buat transformasi dengan nama tf_log_tabel_masuk. Proses dimulai dengan mengambil data dari tabel OLTP dan menyimpannya ke dalam tabel log_tabel_masuk. Dapat dilihat pada Gambar 3.12.



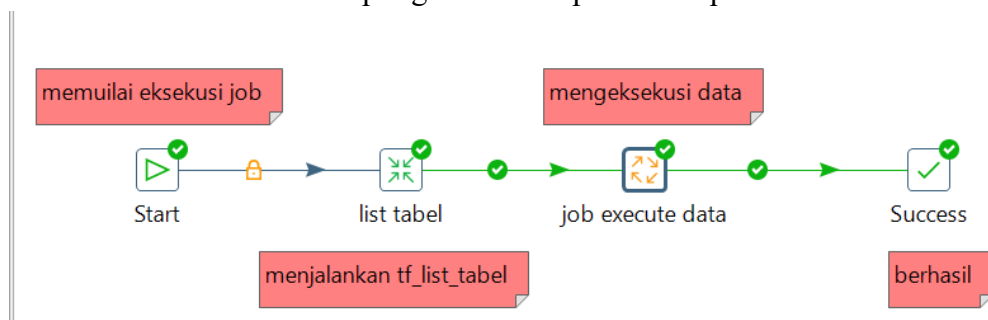
Gambar 3. 12 tf_log_tabel_masuk

- Selanjutnya, buat job dengan nama job_execute_data. Proses dimulai dari Start, diikuti dengan langkah "Transformasi" untuk membaca data tf_read_data. Data kemudian diproses pada langkah "Target Records & Target Date", yang berfungsi untuk menyaring data, memperbarui tabel log dengan jumlah data yang dihitung dari tabel *staging*, dan mencatat waktu eksekusi proses. Selanjutnya, transformasi logis dilakukan di langkah "Logical" untuk memvalidasi apakah jumlah data sumber sama dengan jumlah data target. Jika sama, status akan diisi dengan "Success", dan jika tidak, akan diisi dengan "To Be Evaluated". Hasilnya dicatat ke tabel log melalui langkah tf_log_tabel_masuk untuk keperluan pelacakan dan *debugging*. "Success" menandakan proses selesai. Dapat dilihat pada Gambar 3.13.



Gambar 3. 13 job_execute_data

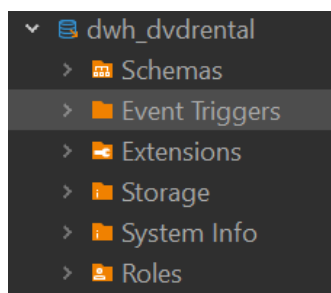
- Terakhir, buat job dengan nama main_job. Proses dimulai dari langkah “Start”, diikuti dengan langkah “Transformation” untuk membaca data dari tf_list_tabel. Selanjutnya, langkah “Job” akan menjalankan sub-job job_execute_data untuk memproses data. Proses diakhiri dengan ”Success”, yang menandakan bahwa semua langkah telah berhasil dijalankan. *Staging area* berhasil dibuat dan siap digunakan. Dapat dilihat pada Gambar 3.14.



Gambar 3. 14 main_job

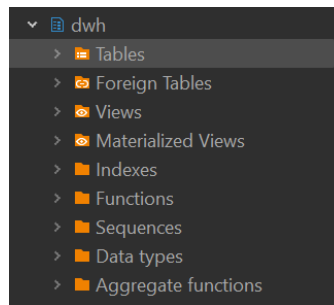
3.3. Desain Data Warehouse

- Pertama, buat database baru pada Dbeaver dengan nama dwh_dvdrental. Dapat dilihat pada Gambar 3.15.



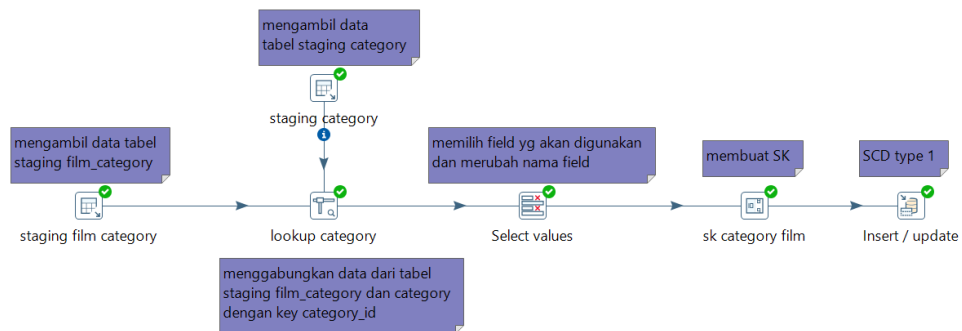
Gambar 3. 15 Database dwh_dvdrental

- Kemudian, buat *schemas* dengan nama dwh. Dapat dilihat pada Gambar 3.16.



Gambar 3. 16 *Schemas dwh*

3. Berikutnya, pada Pentaho buat transformasi tabel dimensi dengan nama `dim_category_film`. Proses ini dimulai dengan mengambil data dari tabel *staging* `film_category` dan `category`. Data dari kedua tabel digabungkan menggunakan *key* `category_id`. Setelah proses penggabungan, dilakukan seleksi *field* yang relevan serta penyesuaian nama *field* agar lebih deskriptif dan mudah dipahami. Selanjutnya, buat *Surrogate Key* sebagai *Primary Key* untuk tabel dimensi. Hasil akhir disimpan ke dalam tabel `dim_category_film` menggunakan metode SCD (*Slowly Changing Dimension*) *Type* 1. Dapat dilihat pada Gambar 3.17, dengan *preview* data pada Gambar 3.18.

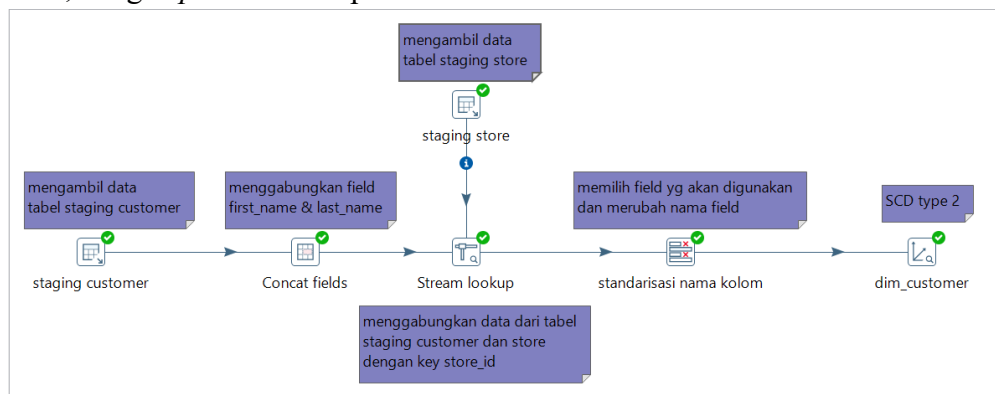


Gambar 3. 17 `dim_category_film`

Logging Execution History Step Metrics Performance Graph Metrics Preview data				
First rows Last rows Off				
#	film_id	category_id	category_film_name	sk_category_film
1	1	6	Documentary	1
2	2	11	Horror	2
3	3	6	Documentary	3
4	4	11	Horror	4
5	5	8	Family	5
6	6	9	Foreign	6
7	7	5	Comedy	7
8	8	11	Horror	8
9	9	11	Horror	9
10	10	15	Sports	10
11	11	9	Foreign	11
12	12	12	Music	12
13	13	11	Horror	13
14	14	4	Classics	14
15	15	9	Foreign	15
16	16	9	Foreign	16
17	17	12	Music	17
18	18	2	Animation	18
19	19	1	Action	19
20	20	12	Music	20
21	21	1	Action	21
22	22	13	New	22
23	23	2	Animation	23
24	24	11	Horror	24

Gambar 3. 18 Preview Data dim_category_film

- Setelah itu, buat transformasi tabel dimensi dengan nama dim_customer. Proses ini dimulai dengan mengambil data dari tabel *staging* customer dan store. *Field* first_name dan last_name pada data tabel *staging* customer digabungkan menjadi satu *field*. Data dari kedua tabel digabungkan menggunakan *key* store_id. Setelah proses penggabungan, dilakukan seleksi *field* yang relevan serta penyesuaian nama *field* agar lebih deskriptif dan mudah dipahami. Selanjutnya, buat *Surrogate Key* sebagai *Primary Key* untuk tabel dimensi. Hasil akhir disimpan ke dalam tabel dim_customer menggunakan metode SCD (*Slowly Changing Dimension*) Type 2. Dapat dilihat pada Gambar 3.19, dengan *preview data* pada Gambar 3.20.

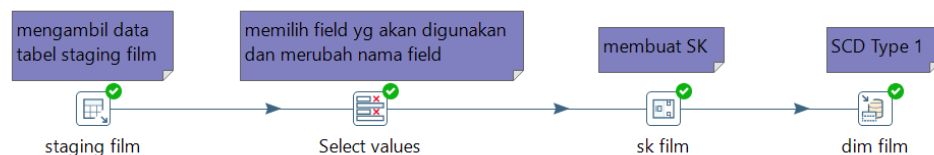


Gambar 3. 19 dim_customer

Logging Execution History Step Metrics Performance Graph Metrics Preview data					
First rows Last rows Off					
#	customer_id	customer_active	customer_name	store_id	sk_customer
1	524	1	Jared Ely	1	1
2	1	1	Mary Smith	1	2
3	2	1	Patricia Johnson	1	3
4	3	1	Linda Williams	1	4
5	4	1	Barbara Jones	2	5
6	5	1	Elizabeth Brown	1	6
7	6	1	Jennifer Davis	2	7
8	7	1	Maria Miller	1	8
9	8	1	Susan Wilson	2	9
10	9	1	Margaret Moore	2	10
11	10	1	Dorothy Taylor	1	11
12	11	1	Lisa Anderson	2	12
13	12	1	Nancy Thomas	1	13
14	13	1	Karen Jackson	2	14
15	14	1	Betty White	2	15
16	15	1	Helen Harris	1	16
17	16	0	Sandra Martin	2	17
18	17	1	Donna Thompson	1	18
19	18	1	Carol Garcia	2	19
20	19	1	Ruth Martinez	1	20
21	20	1	Sharon Robinson	2	21
22	21	1	Michelle Clark	1	22
23	22	1	Laura Rodriguez	1	23
24	23	1	Sarah Lewis	2	24

Gambar 3. 20 Preview Data dim_customer

- Kemudian, buat transformasi tabel dimensi dengan nama dim_ film. Proses ini dimulai dengan mengambil data dari tabel *staging* film. Setelah itu, dilakukan seleksi *field* yang relevan serta penyesuaian nama *field* agar lebih deskriptif dan mudah dipahami. Selanjutnya, buat *Surrogate Key* sebagai *Primary Key* untuk tabel dimensi. Hasil akhir disimpan ke dalam tabel dim_film menggunakan metode SCD (*Slowly Changing Dimension*) Type 1. Dapat dilihat pada Gambar 3.21, dengan *preview* data pada Gambar 3.22 dan Gambar 3.23.



Gambar 3. 21 dim_film

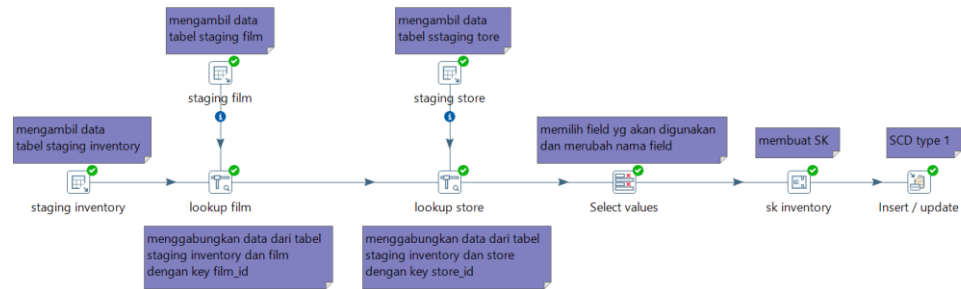
Logging Execution History Step Metrics Performance Graph Metrics Preview data			
First rows Last rows Off			
#	film_id	film_title	film_desc
1	133	Chamber Italian	A Fateful Reflection of a Moose And a Husband who must Overcome a Monkey in Nigeria
2	384	Grosse Wonderful	A Epic Drama of a Cat And a Explorer who must Redeem a Moose in Australia
3	8	Airport Pollock	A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India
4	98	Bright Encounters	A Fateful Yarn of a Lumberjack And a Feminist who must Conquer a Student in A Jet Boat
5	1	Academy Dinosaur	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies
6	2	Ace Goldfinger	A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China
7	3	Adaptation Holes	A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory
8	4	Affair Prejudice	A Fanciful Documentary of a Frisbee And a Lumberjack who must Chase a Monkey in A Shark Tank
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
10	6	Agent Truman	A Intrepid Panorama of a Robot And a Boy who must Escape a Sumo Wrestler in Ancient China
11	7	Airplane Sierra	A Touching Saga of a Hunter And a Butler who must Discover a Butler in A Jet Boat
12	9	Alabama Devil	A Thoughtful Panorama of a Database Administrator And a Mad Scientist who must Outgun a Mad Scientist in A Jet Boat
13	10	Aladdin Calendar	A Action-Packed Tale of a Man And a Lumberjack who must Reach a Feminist in Ancient China
14	11	Alamo Videotape	A Boring Epistle of a Butler And a Cat who must Fight a Pastry Chef in A MySQL Convention
15	12	Alaska Phantom	A Fanciful Saga of a Hunter And a Pastry Chef who must Vanquish a Boy in Australia
16	213	Date Speed	A Touching Saga of a Composer And a Moose who must Discover a Dentist in A MySQL Convention
17	13	Ali Forever	A Action-Packed Drama of a Dentist And a Crocodile who must Battle a Feminist in The Canadian Rockies
18	14	Alice Fantasia	A Emotional Drama of a A Shark And a Database Administrator who must Vanquish a Pioneer in Soviet Georgia
19	15	Alien Center	A Brilliant Drama of a Cat And a Mad Scientist who must Battle a Feminist in A MySQL Convention
20	16	Alley Evolution	A Fast-Paced Drama of a Robot And a Composer who must Battle a Astronaut in New Orleans
21	17	Alone Trip	A Fast-Paced Character Study of a Composer And a Dog who must Outgun a Boat in An Abandoned Fun House
22	18	Alter Victory	A Thoughtful Drama of a Composer And a Feminist who must Meet a Secret Agent in The Canadian Rockies
23	19	Amadeus Holy	A Emotional Display of a Pioneer And a Technical Writer who must Battle a Man in A Baloon

Gambar 3. 22 Preview Data dim_film (1)

ance Graph Metrics Preview data				
film_release_year	film_rental_duration	film_rental_rate	film_special_features	sk_film
2006	7	4.99	(Trailers)	1
2006	5	4.99	("Behind the Scenes")	2
2006	6	4.99	(Trailers)	3
2006	4	4.99	(Trailers)	4
2006	6	0.99	("Deleted Scenes","Behind the Scenes")	5
2006	3	4.99	(Trailers,"Deleted Scenes")	6
2006	7	2.99	(Trailers,"Deleted Scenes")	7
2006	5	2.99	(Commentaries,"Behind the Scenes")	8
2006	6	2.99	("Deleted Scenes")	9
2006	3	2.99	("Deleted Scenes")	10
2006	6	4.99	(Trailers,"Deleted Scenes")	11
2006	3	2.99	(Trailers,"Deleted Scenes")	12
2006	6	4.99	(Trailers,"Deleted Scenes")	13
2006	6	0.99	(Commentaries,"Behind the Scenes")	14
2006	6	0.99	(Commentaries,"Deleted Scenes")	15
2006	4	0.99	(Commentaries)	16
2006	4	4.99	("Deleted Scenes","Behind the Scenes")	17
2006	6	0.99	(Trailers,"Deleted Scenes","Behind the Scenes")	18
2006	5	2.99	(Trailers,Commentaries,"Behind the Scenes")	19
2006	6	2.99	(Trailers,Commentaries)	20
2006	3	0.99	(Trailers,"Behind the Scenes")	21
2006	6	0.99	(Trailers,"Behind the Scenes")	22
2006	6	0.99	(Commentaries,"Deleted Scenes","Behind the Scenes")	23

Gambar 3. 23 Preview Data dim_film (2)

- Lalu, buat transformasi tabel dimensi dengan nama dim_inventory. Proses ini dimulai dengan mengambil data dari tabel *staging* inventory, film, dan store. Data dari ketiga tabel digabungkan menggunakan *key* seperti film_id dan store_id. Setelah proses penggabungan, dilakukan seleksi *field* yang relevan serta penyesuaian nama *field* agar lebih deskriptif dan mudah dipahami. Selanjutnya, buat *Surrogate Key* sebagai *Primary Key* untuk tabel dimensi. Hasil akhir disimpan ke dalam tabel dim_inventory menggunakan metode SCD (*Slowly Changing Dimension*) Type 1. Dapat dilihat pada Gambar 3.24, dengan *preview data* pada Gambar 3.25.



Gambar 3. 24 dim_inventory

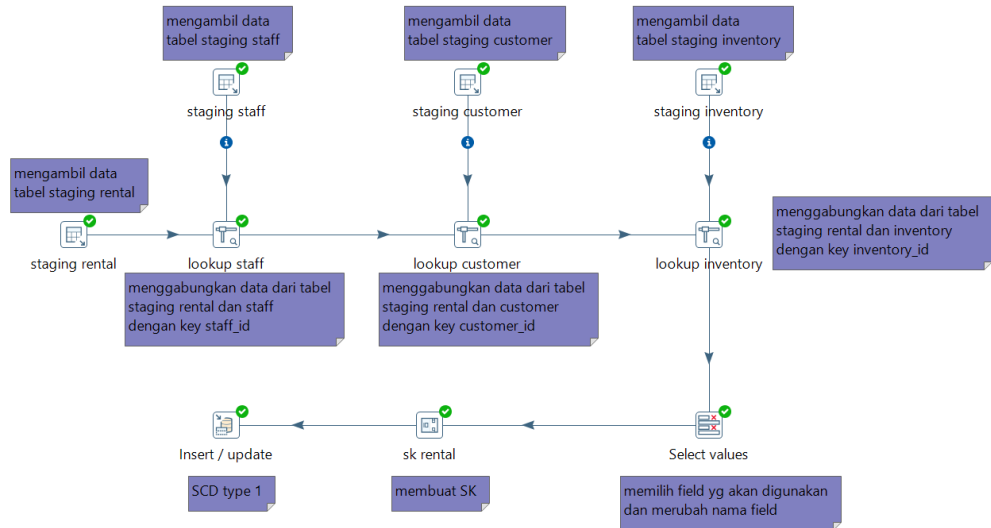
Logging Execution History Step Metrics Performance

First rows Last rows Off

#	inventory_id	film_id	store_id	sk_inventory
1	1	1	1	1
2	2	1	1	2
3	3	1	1	3
4	4	1	1	4
5	5	1	2	5
6	6	1	2	6
7	7	1	2	7
8	8	1	2	8
9	9	2	2	9
10	10	2	2	10
11	11	2	2	11
12	12	3	2	12
13	13	3	2	13
14	14	3	2	14
15	15	3	2	15
16	16	4	1	16
17	17	4	1	17
18	18	4	1	18
19	19	4	1	19
20	20	4	2	20
21	21	4	2	21
22	22	4	2	22
23	23	5	2	23
24	24	5	2	24

Gambar 3. 25 Preview Data dim_inventory

- Berikutnya, buat transformasi tabel dimensi dengan nama dim_rental. Proses ini dimulai dengan mengambil data dari tabel *staging* rental, staff, customer, dan inventory. Data dari keempat tabel digabungkan menggunakan *key* seperti *staff_id*, *customer_id*, dan *inventory_id*. Setelah proses penggabungan, dilakukan seleksi *field* yang relevan serta penyesuaian nama *field* agar lebih deskriptif dan mudah dipahami. Selanjutnya, buat *Surrogate Key* sebagai *Primary Key* untuk tabel dimensi. Hasil akhir disimpan ke dalam tabel dim_rental menggunakan metode SCD (*Slowly Changing Dimension*) Type 1. Dapat dilihat pada Gambar 3.26, dengan *preview* data pada Gambar 3.27.

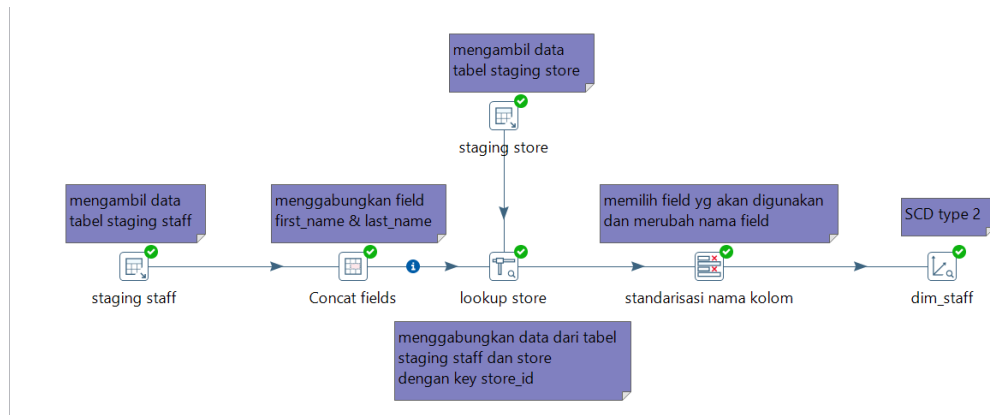


Gambar 3. 26 dim_rental

Logging Execution History Step Metrics Performance Graph Metrics Preview data							
First rows Last rows Off							
#	rental_id	rental_date	rental_return_date	staff_id	customer_id	inventory_id	sk_rental
1	2	2005/05/24 22:54:33.000000000	2005/05/28 19:40:33.000000000	1	459	1525	1
2	3	2005/05/24 23:03:39.000000000	2005/06/01 22:12:39.000000000	1	408	1711	2
3	4	2005/05/24 23:04:41.000000000	2005/06/03 01:43:41.000000000	2	333	2452	3
4	5	2005/05/24 23:05:21.000000000	2005/06/02 04:33:21.000000000	1	222	2079	4
5	6	2005/05/24 23:08:07.000000000	2005/05/27 01:32:07.000000000	1	549	2792	5
6	7	2005/05/24 23:11:53.000000000	2005/05/29 20:34:53.000000000	2	269	3995	6
7	8	2005/05/24 23:31:46.000000000	2005/05/27 23:33:46.000000000	2	239	2346	7
8	9	2005/05/25 00:00:40.000000000	2005/05/28 00:22:40.000000000	1	126	2580	8
9	10	2005/05/25 00:02:21.000000000	2005/05/31 22:44:21.000000000	2	399	1824	9
10	11	2005/05/25 00:09:02.000000000	2005/06/02 20:56:02.000000000	2	142	4443	10
11	12	2005/05/25 00:19:27.000000000	2005/05/30 05:44:27.000000000	2	261	1584	11
12	13	2005/05/25 00:22:55.000000000	2005/05/30 04:28:55.000000000	1	334	2294	12
13	14	2005/05/25 00:31:15.000000000	2005/05/26 02:56:15.000000000	1	446	2701	13
14	15	2005/05/25 00:39:22.000000000	2005/06/03 03:30:22.000000000	1	319	3049	14
15	16	2005/05/25 00:43:11.000000000	2005/05/26 04:42:11.000000000	2	316	389	15
16	17	2005/05/25 01:06:36.000000000	2005/05/27 00:43:36.000000000	1	575	830	16
17	18	2005/05/25 01:10:47.000000000	2005/05/31 06:35:47.000000000	2	19	3376	17
18	19	2005/05/25 01:17:24.000000000	2005/05/31 06:00:24.000000000	1	456	1941	18
19	20	2005/05/25 01:48:41.000000000	2005/05/27 02:20:41.000000000	2	185	3517	19
20	21	2005/05/25 01:59:46.000000000	2005/05/26 01:01:46.000000000	2	388	146	20
21	22	2005/05/25 02:19:23.000000000	2005/05/26 04:52:23.000000000	2	509	727	21
22	23	2005/05/25 02:40:21.000000000	2005/05/29 06:34:21.000000000	1	438	4441	22
23	24	2005/05/25 02:53:02.000000000	2005/05/27 01:15:02.000000000	1	350	3273	23
24	25	2005/05/25 03:21:20.000000000	2005/05/27 21:25:20.000000000	2	37	3961	24

Gambar 3. 27 Preview Data dim_rental

- Setelah itu, buat transformasi tabel dimensi dengan nama dim_staff. Proses ini dimulai dengan mengambil data dari tabel *staging* staff dan store. *Field* first_name dan last_name pada data tabel *staging* staff digabungkan menjadi satu *field*. Data dari kedua tabel digabungkan menggunakan *key* store_id. Setelah proses penggabungan, dilakukan seleksi *field* yang relevan serta penyesuaian nama *field* agar lebih deskriptif dan mudah dipahami. Selanjutnya, buat *Surrogate Key* sebagai *Primary Key* untuk tabel dimensi. Hasil akhir disimpan ke dalam tabel dim_staff menggunakan metode SCD (*Slowly Changing Dimension*) Type 2. Dapat dilihat pada Gambar 3.28, dengan *preview* data pada Gambar 3.29.



Gambar 3. 28 dim_staff

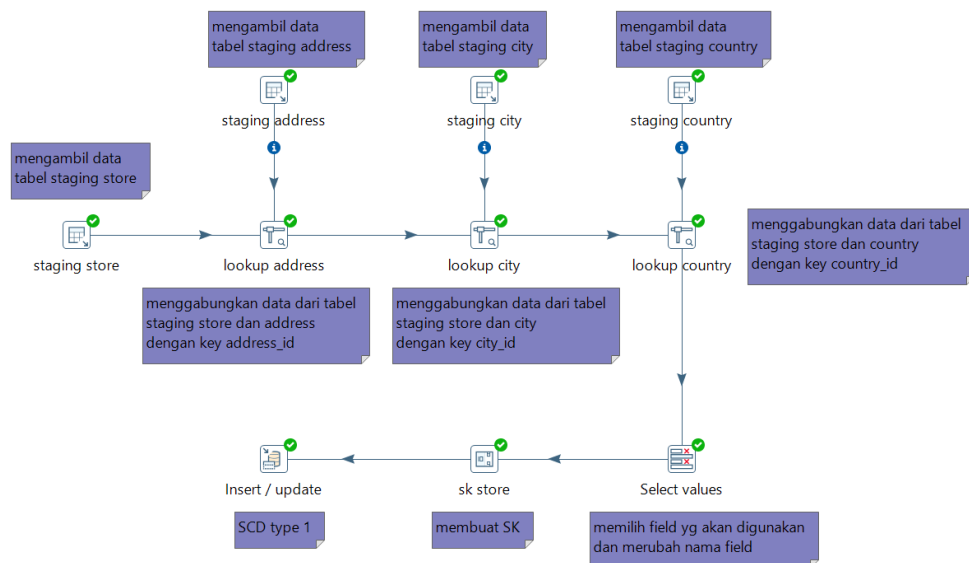
Logging Execution History Step Metrics Per

First rows Last rows Off

#	staff_id	staff_name	store_id	sk_staff
1	1	Mike Hillyer	1	1
2	2	Jon Stephens	2	2

Gambar 3. 29 Preview Data dim_staff

9. Berikutnya, buat transformasi tabel dimensi dengan nama dim_store. Proses ini dimulai dengan mengambil data dari tabel *staging* store, address, city, dan country. Data dari keempat tabel digabungkan menggunakan *key* seperti *address_id*, *city_id*, dan *country_id*. Setelah proses penggabungan, dilakukan seleksi *field* yang relevan serta penyesuaian nama *field* agar lebih deskriptif dan mudah dipahami. Selanjutnya, buat *Surrogate Key* sebagai *Primary Key* untuk tabel dimensi. Hasil akhir disimpan ke dalam tabel *dim_store* menggunakan metode SCD (*Slowly Changing Dimension*) *Type 1*. Dapat dilihat pada Gambar 3.30, dengan *preview* data pada Gambar 3.31.



Gambar 3. 30 dim_store

Logging

Execution History

Step Metrics

Performance Graph

Metrics

Preview data

First rows

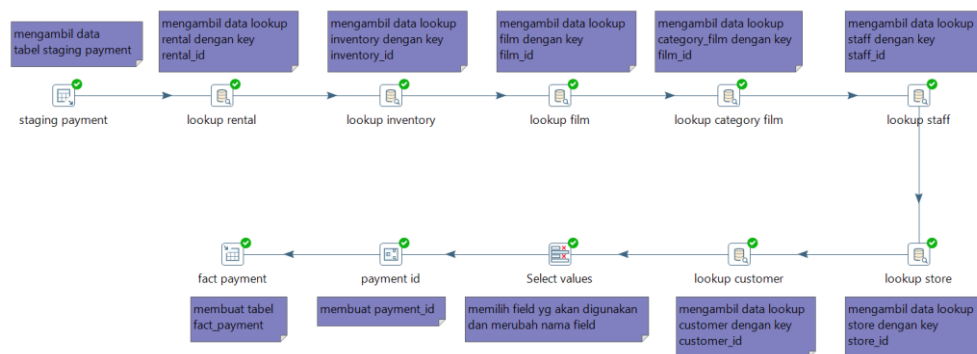
Last rows

Off

#	store_id	address_id	store_address	city_id	store_city	country_id	store_country	sk_store
1	1	1	47 MySakila Drive	300	Lethbridge	20	Canada	1
2	2	2	28 MySQL Boulevard	576	Woodridge	8	Australia	2

Gambar 3. 31 Preview Data dim_store

10. Terakhir, buat transformasi tabel fakta dengan nama fact_payment. Proses ini dimulai dengan mengambil data dari tabel *staging* payment sebagai tabel utama. Data ini kemudian diperkaya dengan melakukan proses *lookup* ke beberapa tabel dimensi seperti dim_rental, dim_inventory, dim_film, dim_category_film, dim_staff, dim_store, dan dim_customer. Data dari kedelapan tabel digabungkan menggunakan *key* seperti rental_id, inventory_id, film_id, staff_id, store_id, dan customer_id. Setelah proses penggabungan, dilakukan seleksi *field* yang relevan serta penyesuaian nama *field* agar lebih deskriptif dan mudah dipahami. Selanjutnya, buat ID unik sebagai *Primary Key* untuk tabel fakta. Hasil akhir disimpan ke dalam tabel fact_payment. Dapat dilihat pada Gambar 3.32, dengan *preview* data pada Gambar 3.33.



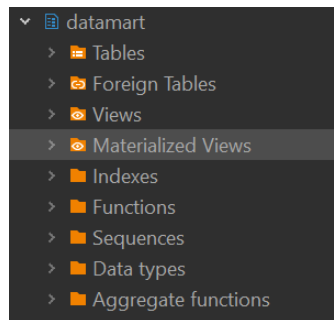
Gambar 3. 32 fact_payment

Logging Execution History Step Metrics Performance Graph Metrics Preview data										
First rows Last rows Off										
#	sk_rental	sk_inventory	sk_film	sk_category_film	sk_staff	sk_store	sk_customer	payment_amount	payment_date	payment_id
1	1518.0	3419.0	750.0	749.0	2	1.0	342	7.99	2007/02/15 22:25:46.996577000	1
2	1776.0	2512.0	555.0	552.0	1	2.0	342	1.99	2007/02/16 17:23:14.996577000	2
3	1847.0	2507.0	554.0	551.0	1	1.0	342	7.99	2007/02/16 22:41:45.996577000	3
4	2826.0	2047.0	447.0	445.0	2	1.0	342	2.99	2007/02/19 19:39:56.996577000	4
5	3127.0	2569.0	564.0	563.0	2	1.0	342	7.99	2007/02/20 17:31:48.996577000	5
6	3379.0	131.0	30.0	26.0	1	1.0	342	5.99	2007/02/21 12:33:49.996577000	6
7	2188.0	4398.0	961.0	961.0	2	2.0	343	5.99	2007/02/17 23:58:17.996577000	7
8	2911.0	2419.0	529.0	527.0	1	2.0	343	5.99	2007/02/20 02:11:44.996577000	8
9	3078.0	2071.0	395.0	449.0	1	2.0	343	2.99	2007/02/20 13:57:39.996577000	9
10	1545.0	2909.0	640.0	639.0	2	2.0	344	4.99	2007/02/16 00:10:50.996577000	10
11	1562.0	145.0	34.0	30.0	1	1.0	344	6.99	2007/02/16 01:15:33.996577000	11
12	1877.0	3983.0	869.0	868.0	2	1.0	344	0.99	2007/02/17 01:26:00.996577000	12
13	1920.0	2484.0	548.0	545.0	2	1.0	344	0.99	2007/02/17 04:32:51.996577000	13
14	2458.0	78.0	20.0	16.0	2	1.0	344	6.99	2007/02/18 18:26:38.996577000	14
15	2977.0	310.0	75.0	69.0	1	2.0	344	8.99	2007/02/20 07:03:29.996577000	15
16	3404.0	3919.0	856.0	855.0	1	2.0	344	0.99	2007/02/21 14:42:28.996577000	16
17	1339.0	3363.0	740.0	738.0	1	1.0	345	3.99	2007/02/15 10:54:44.996577000	17
18	1473.0	1834.0	399.0	398.0	2	2.0	345	4.99	2007/02/15 19:36:27.996577000	18
19	1729.0	3698.0	809.0	809.0	1	2.0	345	0.99	2007/02/16 14:00:38.996577000	19
20	1208.0	7.0	5.0	1.0	2	2.0	346	0.99	2007/02/15 01:26:17.996577000	20
21	1455.0	730.0	162.0	159.0	1	2.0	346	4.99	2007/02/15 18:34:15.996577000	21
22	1548.0	2003.0	438.0	436.0	2	2.0	346	0.99	2007/02/16 00:27:01.996577000	22
23	2763.0	3421.0	750.0	749.0	2	2.0	346	4.99	2007/02/19 16:13:41.996577000	23
24	1992.0	2203.0	476.0	476.0	1	2.0	347	5.99	2007/02/17 09:35:32.996577000	24

Gambar 3. 33 Preview Data fact_payment

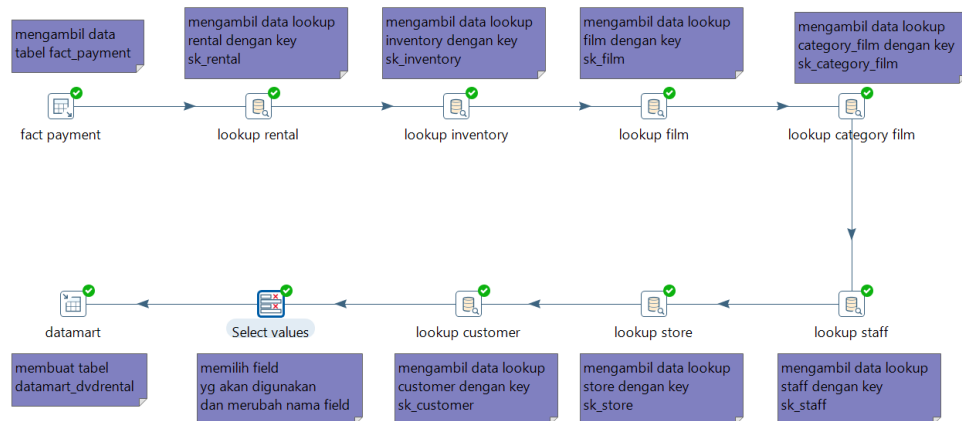
3.4. Pembuatan *Datamart*

1. Pertama, pada *database* *dwh_dvdrental* buat *schemas* baru dengan nama *datamart*. Dapat dilihat pada Gambar 3.34.



Gambar 3. 34 *Schemas* *datamart*

2. Terakhir, buat transformasi *datamart* dengan nama *datamart_dvdrental*. Proses ini dimulai dengan mengambil data dari tabel *staging* *payment* sebagai tabel utama. Data ini kemudian diperkaya dengan melakukan proses *lookup* ke beberapa tabel dimensi seperti *dim_rental*, *dim_inventory*, *dim_film*, *dim_category_film*, *dim_staff*, *dim_store*, dan *dim_customer*. Data dari ketujuh tabel digabungkan menggunakan *key* seperti *sk_rental*, *sk_inventory*, *sk_film*, *sk_staff*, *sk_store*, dan *sk_customer*. Setelah proses penggabungan, dilakukan seleksi *field* yang relevan serta penyesuaian nama *field* agar lebih deskriptif dan mudah dipahami. Hasil akhir disimpan ke dalam tabel *datamart_dvdrental*. Dapat dilihat pada Gambar 3.35 dengan *preview* data pada Gambar 3.36, Gambar 3.37, Gambar 3.38, dan Gambar 3.39.



Gambar 3. 35 *datamart_dvdrental*

Logging Execution History Step Metrics Performance Graph Metrics Preview data								
First rows Last rows Off								
#	payment_id	payment_amount	payment_date	sk_rental	rental_date	rental_return_date	sk_inventory	sk_film
1	1	7.99	2007/02/15 22:25:46.996577000	1518.0	2005/06/15 23:57:20.000000000	2005/06/24 23:46:20.000000000	3419.0	750.0
2	2	1.99	2007/02/16 17:23:14.996577000	1776.0	2005/06/16 18:54:48.000000000	2005/06/22 16:08:48.000000000	2512.0	555.0
3	3	7.99	2007/02/16 22:41:45.996577000	1847.0	2005/06/17 00:13:19.000000000	2005/06/23 18:37:19.000000000	2507.0	554.0
4	4	2.99	2007/02/19 19:39:56.996577000	2826.0	2005/06/19 21:11:30.000000000	2005/06/24 18:10:30.000000000	2047.0	447.0
5	5	7.99	2007/02/20 17:31:48.996577000	3127.0	2005/06/20 19:03:22.000000000	2005/06/29 18:05:22.000000000	2569.0	564.0
6	6	5.99	2007/02/21 12:33:49.996577000	3379.0	2005/06/21 14:05:23.000000000	2005/06/29 19:13:23.000000000	131.0	30.0
7	7	5.99	2007/02/17 23:58:17.996577000	2188.0	2005/06/18 01:29:51.000000000	2005/06/26 04:31:51.000000000	4398.0	961.0
8	8	5.99	2007/02/20 02:11:44.996577000	2911.0	2005/06/20 03:43:18.000000000	2005/06/25 03:44:18.000000000	2419.0	529.0
9	9	2.99	2007/02/20 13:57:39.996577000	3078.0	2005/06/20 15:29:13.000000000	2005/06/24 21:00:13.000000000	2071.0	395.0
10	10	4.99	2007/02/16 00:10:50.996577000	1545.0	2005/06/16 01:42:24.000000000	2005/06/19 01:13:24.000000000	2909.0	640.0
11	11	6.99	2007/02/16 01:15:33.996577000	1562.0	2005/06/16 02:47:07.000000000	2005/06/24 03:12:07.000000000	145.0	34.0
12	12	0.99	2007/02/17 01:26:00.996577000	1877.0	2005/06/17 02:57:34.000000000	2005/06/19 00:00:34.000000000	3983.0	869.0
13	13	0.99	2007/02/17 04:32:51.996577000	1920.0	2005/06/17 06:04:25.000000000	2005/06/18 09:15:25.000000000	2484.0	548.0
14	14	6.99	2007/02/18 18:26:38.996577000	2458.0	2005/06/18 19:58:12.000000000	2005/06/28 01:35:12.000000000	78.0	20.0
15	15	8.99	2007/02/20 07:03:29.996577000	2977.0	2005/06/20 08:35:03.000000000	2005/06/29 07:57:03.000000000	310.0	75.0
16	16	0.99	2007/02/21 14:42:28.996577000	3404.0	2005/06/21 16:14:02.000000000	2005/06/24 15:38:02.000000000	3919.0	856.0
17	17	3.99	2007/02/15 10:54:44.996577000	1339.0	2005/06/15 12:26:18.000000000	2005/06/21 07:26:18.000000000	3363.0	740.0
18	18	4.99	2007/02/15 19:36:27.996577000	1473.0	2005/06/15 21:08:01.000000000	2005/06/18 22:33:01.000000000	1834.0	399.0
19	19	0.99	2007/02/16 14:00:38.996577000	1729.0	2005/06/16 15:32:12.000000000	2005/06/19 18:58:12.000000000	3698.0	809.0
20	20	0.99	2007/02/15 01:26:17.996577000	1208.0	2005/06/15 02:57:51.000000000	2005/06/20 01:41:51.000000000	7.0	5.0
21	21	4.99	2007/02/15 18:34:15.996577000	1455.0	2005/06/15 20:05:49.000000000	2005/06/19 15:35:49.000000000	730.0	162.0
22	22	0.99	2007/02/16 00:27:01.996577000	1548.0	2005/06/16 01:58:35.000000000	2005/06/18 23:56:35.000000000	2003.0	438.0
23	23	4.99	2007/02/19 16:13:41.996577000	2763.0	2005/06/19 17:45:15.000000000	2005/06/23 20:11:15.000000000	3421.0	750.0

Gambar 3. 36 Preview Data datamart_dvdrental (1)

Logging Execution History Step Metrics Performance Graph Metrics Preview data								
First rows Last rows Off								
film_title	film_desc	film_release_year						
Rules Human	A Beautiful Epistle of a Astronaut And a Student who must Confront a Monkey in An Abandoned Fun House	2006						
Majestic Floats	A Thrilling Character Study of a Moose And a Student who must Escape a Butler in The First Manned Space Station	2006						
Maiden Home	A Lacklustre Saga of a Moose And a Teacher who must Kill a Forensic Psychologist in A MySQL Convention	2006						
Hyde Doctor	A Fanciful Documentary of a Boy And a Woman who must Redeem a Womanizer in A Jet Boat	2006						
Massacre Usual	A Fateful Reflection of a Waitress And a Crocodile who must Challenge a Forensic Psychologist in California	2006						
Annie Identity	A Amazing Panorama of a Pastry Chef And a Boat who must Escape a Woman in An Abandoned Amusement Park	2006						
Wash Heavenly	A Awe-Inspiring Reflection of a Cat And a Pioneer who must Escape a Hunter in Ancient China	2006						
Lola Agent	A Astounding Tale of a Mad Scientist And a Husband who must Redeem a Database Administrator in Ancient Japan	2006						
Identity Lover	A Boring Tale of a Composer And a Mad Cow who must Defeat a Car in The Outback	2006						
Opposite Necklace	A Fateful Epistle of a Crocodile And a Moose who must Kill a Explorer in Nigeria	2006						
Anything Savannah	A Epic Story of a Pastry Chef And a Woman who must Chase a Feminist in An Abandoned Fun House	2006						
Superfly Trip	A Beautiful Saga of a Lumberjack And a Teacher who must Build a Technical Writer in An Abandoned Fun House	2006						
Madness Attacks	A Fanciful Tale of a Squirrel And a Boat who must Defeat a Crocodile in The Gulf of Mexico	2006						
Alley Evolution	A Fast-Paced Drama of a Robot And a Composer who must Battle a Astronaut in New Orleans	2006						
Beverly Outlaw	A Fanciful Documentary of a Womanizer And a Boat who must Defeat a Madman in The First Manned Space Station	2006						
Streak Ridgemoor	A Astounding Character Study of a Hunter And a Waitress who must Sink a Man in New Orleans	2006						
Rocketeer Mother	A Awe-Inspiring Character Study of a Robot And a Sumo Wrestler who must Discover a Womanizer in A Shark Tank	2006						
Hanover Galaxy	A Stunning Reflection of a Girl And a Secret Agent who must Succumb a Boy in A MySQL Convention	2006						
Slipper Fidelity	A Taut Reflection of a Secret Agent And a Man who must Redeem a Explorer in A MySQL Convention	2006						
Academy Dinosaur	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies	2006						
Closer Bang	A Unbelievable Panorama of a Frisbee And a Hunter who must Vanquish a Monkey in Ancient India	2006						
Hours Rage	A Fateful Story of a Explorer And a Feminist who must Meet a Technical Writer in Soviet Georgia	2006						
Rules Human	A Beautiful Epistle of a Astronaut And a Student who must Confront a Monkey in An Abandoned Fun House	2006						

Gambar 3. 37 Preview Data datamart_dvdrental (2)

Logging Execution History Step Metrics Performance Graph Metrics Preview data								
First rows Last rows Off								
film_rental_duration	film_rental_rate	film_special_features	sk_category_film	category_film_name	sk_staff	staff_name	sk_store	
6	4.99	("Deleted Scenes","Behind the Scenes")	749.0	Horror	2	Jon Stephens	1.0	
5	0.99	(Trailers)	552.0	Documentary	1	Mike Hillyer	2.0	
3	4.99	("Behind the Scenes")	551.0	New	1	Mike Hillyer	1.0	
5	2.99	(Trailers,"Deleted Scenes")	445.0	Classics	2	Jon Stephens	1.0	
6	4.99	(Commentaries)	563.0	Games	2	Jon Stephens	1.0	
3	0.99	(Commentaries,"Deleted Scenes")	26.0	Sci-Fi	1	Mike Hillyer	1.0	
7	4.99	(Commentaries)	961.0	Foreign	2	Jon Stephens	2.0	
4	4.99	(Trailers,Commentaries)	527.0	Horror	1	Mike Hillyer	2.0	
4	2.99	("Deleted Scenes")	449.0	Sci-Fi	1	Mike Hillyer	2.0	
7	4.99	("Deleted Scenes")	639.0	Family	2	Jon Stephens	2.0	
4	2.99	(Trailers,"Deleted Scenes","Behind the Scenes")	30.0	Horror	1	Mike Hillyer	1.0	
5	0.99	(Commentaries,"Behind the Scenes")	868.0	Travel	2	Jon Stephens	1.0	
4	0.99	(Trailers)	545.0	Foreign	2	Jon Stephens	1.0	
6	2.99	(Trailers,Commentaries)	16.0	Foreign	2	Jon Stephens	1.0	
3	2.99	(Trailers)	69.0	Sci-Fi	1	Mike Hillyer	2.0	
7	0.99	(Trailers,"Behind the Scenes")	855.0	Documentary	1	Mike Hillyer	2.0	
3	0.99	("Deleted Scenes","Behind the Scenes")	738.0	Foreign	1	Mike Hillyer	1.0	
5	4.99	(Commentaries,"Deleted Scenes","Behind the Scenes")	398.0	Music	2	Jon Stephens	2.0	
5	0.99	(Trailers,"Deleted Scenes")	809.0	Sports	1	Mike Hillyer	2.0	
6	0.99	("Deleted Scenes","Behind the Scenes")	1.0	Documentary	2	Jon Stephens	2.0	
5	4.99	(Trailers,"Behind the Scenes")	159.0	Comedy	1	Mike Hillyer	2.0	
4	0.99	(Trailers,"Deleted Scenes")	436.0	New	2	Jon Stephens	2.0	
6	4.99	("Deleted Scenes","Behind the Scenes")	749.0	Horror	2	Jon Stephens	2.0	

Gambar 3. 38 Preview Data datamart_dvdrental (3)

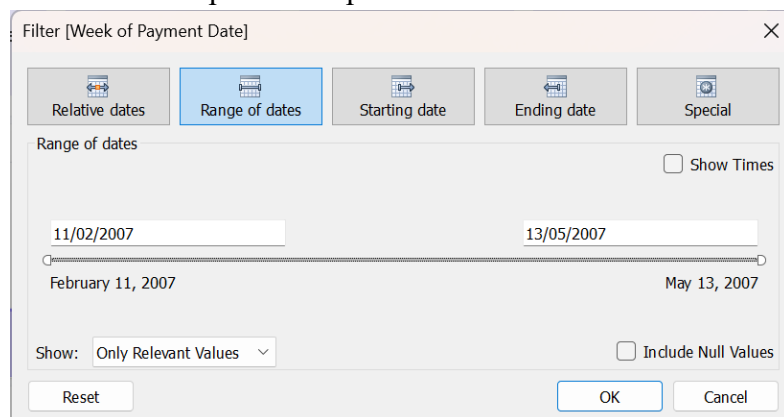
Preview data					
store_address	store_city	store_country	sk_customer	customer_name	customer_active
47 MySakila Drive	Lethbridge	Canada	342	Peter Menard	1
28 MySQL Boulevard	Woodridge	Australia	342	Peter Menard	1
47 MySakila Drive	Lethbridge	Canada	342	Peter Menard	1
47 MySakila Drive	Lethbridge	Canada	342	Peter Menard	1
47 MySakila Drive	Lethbridge	Canada	342	Peter Menard	1
47 MySakila Drive	Lethbridge	Canada	342	Peter Menard	1
28 MySQL Boulevard	Woodridge	Australia	343	Harold Martino	1
28 MySQL Boulevard	Woodridge	Australia	343	Harold Martino	1
28 MySQL Boulevard	Woodridge	Australia	343	Harold Martino	1
28 MySQL Boulevard	Woodridge	Australia	344	Douglas Graf	1
47 MySakila Drive	Lethbridge	Canada	344	Douglas Graf	1
47 MySakila Drive	Lethbridge	Canada	344	Douglas Graf	1
47 MySakila Drive	Lethbridge	Canada	344	Douglas Graf	1
47 MySakila Drive	Lethbridge	Canada	344	Douglas Graf	1
28 MySQL Boulevard	Woodridge	Australia	344	Douglas Graf	1
28 MySQL Boulevard	Woodridge	Australia	344	Douglas Graf	1
47 MySakila Drive	Lethbridge	Canada	345	Henry Billingsley	1
28 MySQL Boulevard	Woodridge	Australia	345	Henry Billingsley	1
28 MySQL Boulevard	Woodridge	Australia	345	Henry Billingsley	1
28 MySQL Boulevard	Woodridge	Australia	346	Carl Artis	1
28 MySQL Boulevard	Woodridge	Australia	346	Carl Artis	1
28 MySQL Boulevard	Woodridge	Australia	346	Carl Artis	1
28 MySQL Boulevard	Woodridge	Australia	346	Carl Artis	1

Gambar 3. 39 Preview Data datamart_dvdrental (4)

3.5. Visualisasi dan Analisis *Dashboard*

- **Pembuatan Filter**

1. Pada Tableau, buat Filter Date dengan rentang tanggal 11 Februari 2007 s.d 13 Mei 2007. Dapat dilihat pada Gambar 3.40.



Gambar 3. 40 Filter Date

- **Pembuatan Parameter**

1. Pada Tableau, buat parameter bernama Select Store yang berfungsi sebagai filter kedua toko DVD Rental. Pilih “View Data”, lalu pilih “Create Parameter”, beri nama Select Store, dan masukkan data seperti yang ditampilkan pada Gambar 3.41.

Edit Parameter [Select Store] x

Name
Select Store

Properties

Data type
String

Display format
28 MySQL Boulevard

Current value
28 MySQL Boulevard

Value when workbook opens
Current value

Allowable values
☐ All ☒ List ☐ Range

Value	Display As
28 MySQL Boulevard	28 MySQL Boulevard
47 MySakila Drive	47 MySakila Drive
Click to add	

☒ Fixed
☐ When workbook opens

Add values from ▼

Remove Selected

Cancel OK

Gambar 3. 41 Parameter Select Store

- **Pembuatan Calculation Field**

1. Pada Tableau, buat Calculation Field bernama Store Select yang digunakan sebagai filter dinamis pada *worksheet* informasi dan grafik. Pilih “View Data”, lalu “Create Calculated Field”, beri nama Store Select, dan masukkan formula berikut:

`CONTAINS([Selected Store], [Store Address])`

Jika muncul tulisan The calculation is valid, klik “Apply”. Formula ini berfungsi memeriksa apakah nama toko dalam *field* Store Address ada di dalam parameter Select Store. Jika hasilnya True, data toko akan ditampilkan. Jika False, data toko akan disembunyikan. Dapat dilihat pada Gambar 3.42.

store select x

`CONTAINS([Select Store], [Store Address])`

The calculation is valid. 12 Dependencies ▼

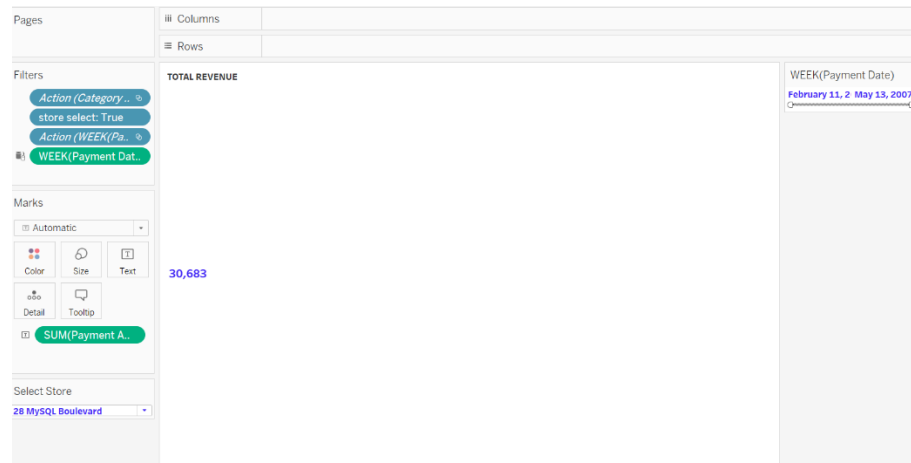
Apply OK

Gambar 3. 42 Calculation Field Store Select

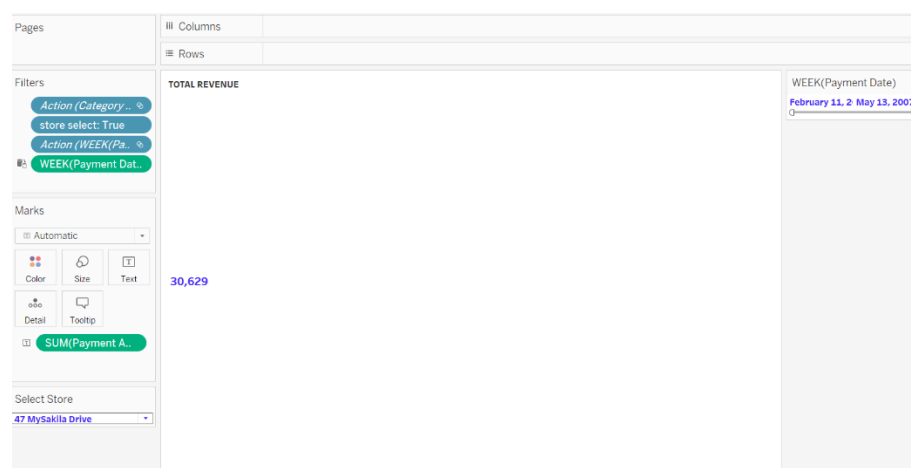
- **Pembuatan Grafik**

1. Pertama, pada Tableau, buat grafik dengan nama Total Revenue. Grafik ini menggunakan jenis visualisasi *KPI Card*. Total Revenue adalah total pendapatan yang dihasilkan oleh toko DVD Rental selama periode 11

Februari 2007 hingga 13 Mei 2007. Pendapatan ini dihitung berdasarkan biaya yang dibayarkan pelanggan untuk menyewa DVD di toko. Grafik ini mencakup pendapatan dari dua toko DVD Rental yang berlokasi di negara berbeda, diantaranya Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.43 dan Gambar 3.44.

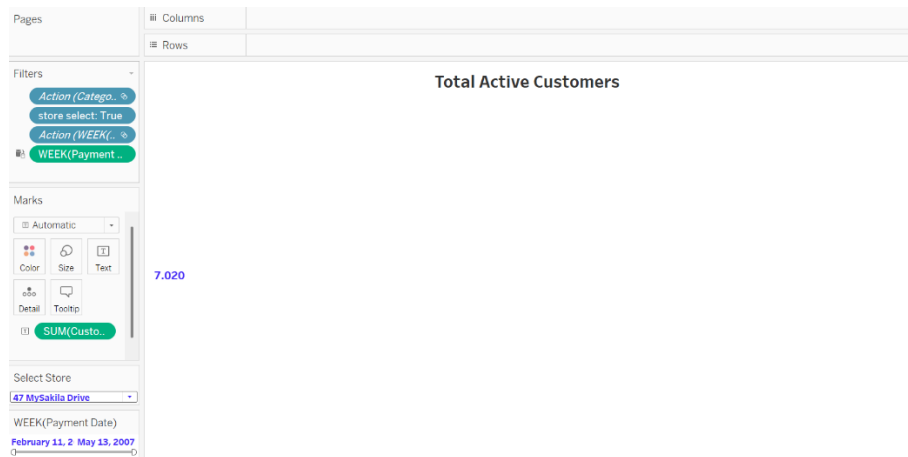


Gambar 3. 43 Total Revenue Toko 28 MySQL Boulevard

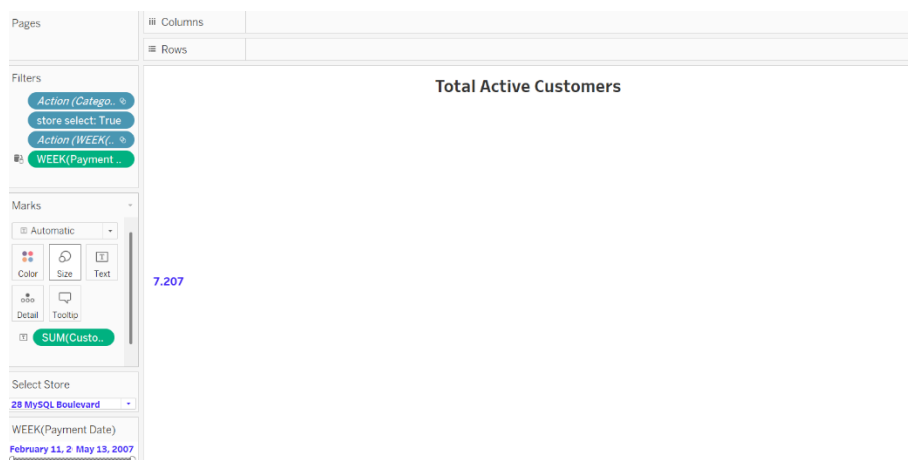


Gambar 3. 44 Total Revenue Toko 47 MySakila Drive

2. Berikutnya, buat grafik dengan nama Total Active Customers. Grafik ini menggunakan jenis visualisasi *KPI Card*. Total Active Customers adalah total pelanggan aktif yang melakukan transaksi penyewaan DVD dan terdaftar di toko DVD Rental selama periode 11 Februari 2007 hingga 13 Mei 2007. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.45 dan Gambar 4.46.

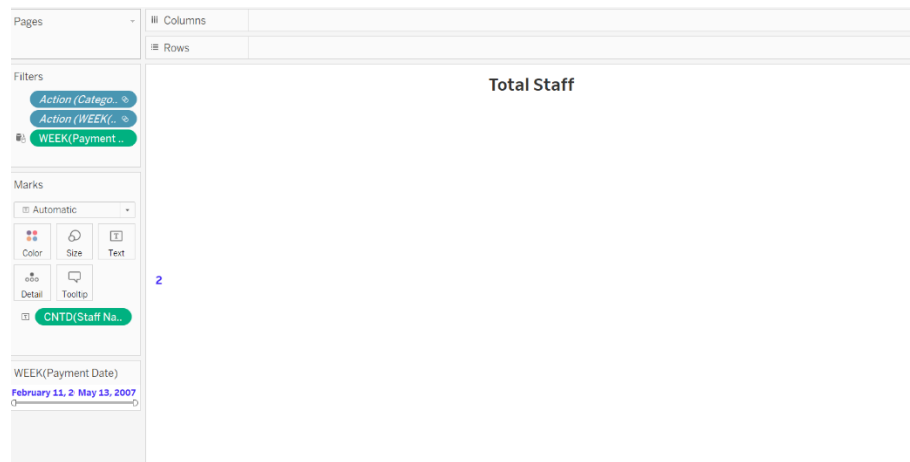


Gambar 3. 45 Total Active Customers Toko 28 MySQL Boulevard



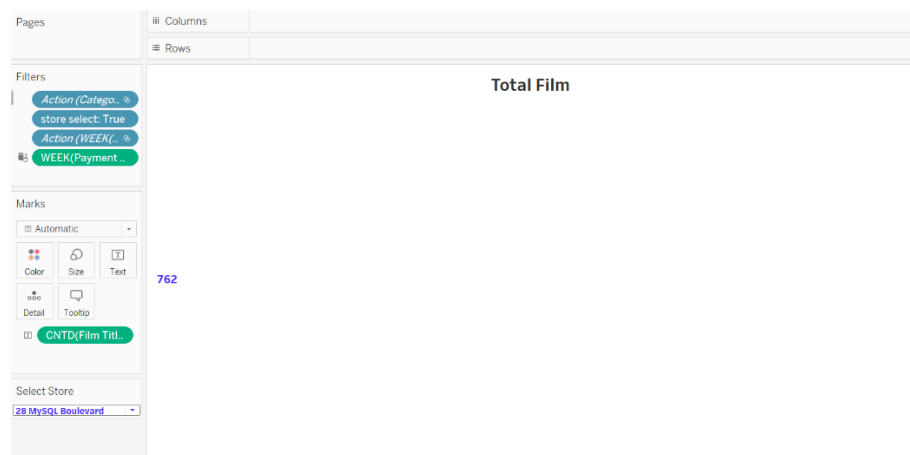
Gambar 3. 46 Total Active Customers Toko 47 MySakila Drive

- Selanjutnya, buat grafik dengan nama Total Staff. Grafik ini menggunakan jenis visualisasi *KPI Card*. Total Staff adalah total staff yang bekerja di toko DVD Rental selama periode 11 Februari 2007 hingga 13 Mei 2007. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.47.

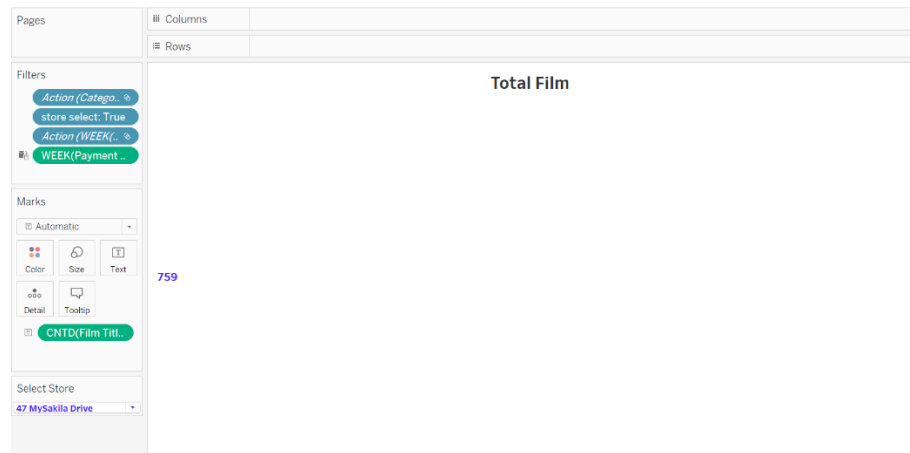


Gambar 3. 47 Total Staff

4. Setelah itu, buat grafik dengan nama Total Film. Grafik ini menggunakan jenis visualisasi *KPI Card*. Total Film adalah total film yang tersedia di toko DVD Rental. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.48 dan Gambar 3.49.

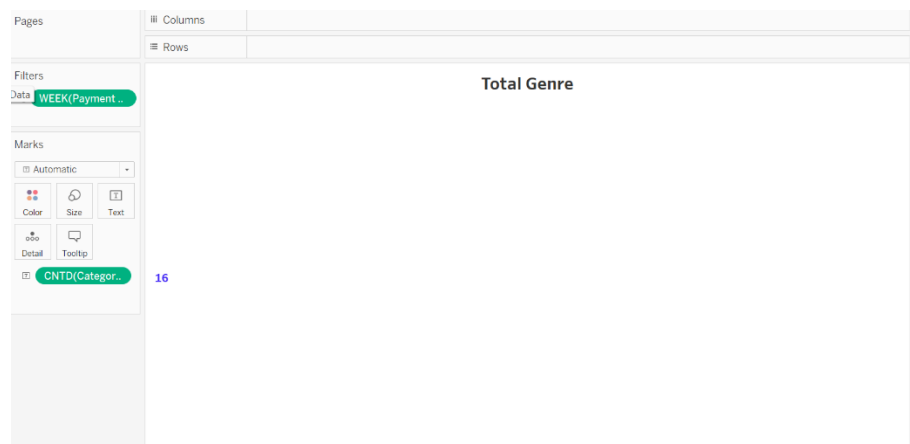


Gambar 3. 48 Total Film Toko 28 MySQL Boulevard



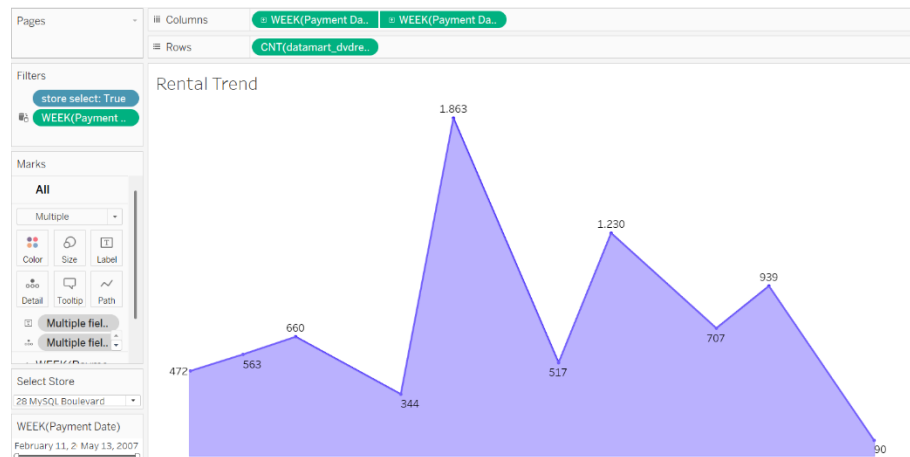
Gambar 3. 49 Total Film Toko 47 MySakila Drive

5. Lalu, buat grafik dengan nama Total Genre. Grafik ini menggunakan jenis visualisasi *KPI Card*. Total Genre adalah total kategori atau jenis film yang tersedia di toko DVD Rental. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.50.

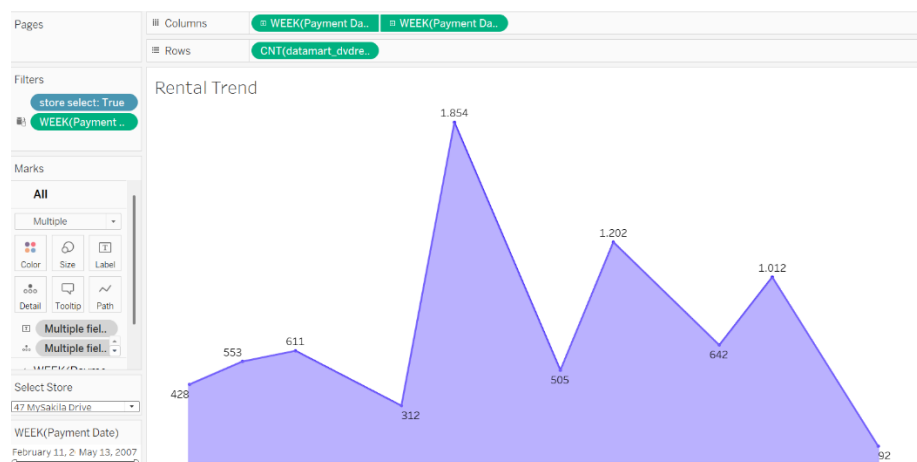


Gambar 3. 50 Total Genre

6. Kemudian, buat grafik dengan nama Rental Trend. Grafik ini menggunakan jenis visualisasi *Area Chart*. Rental Trend menunjukkan jumlah transaksi penyewaan DVD selama periode 11 Februari 2007 hingga 13 Mei 2007. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.51 dan Gambar 3.52.

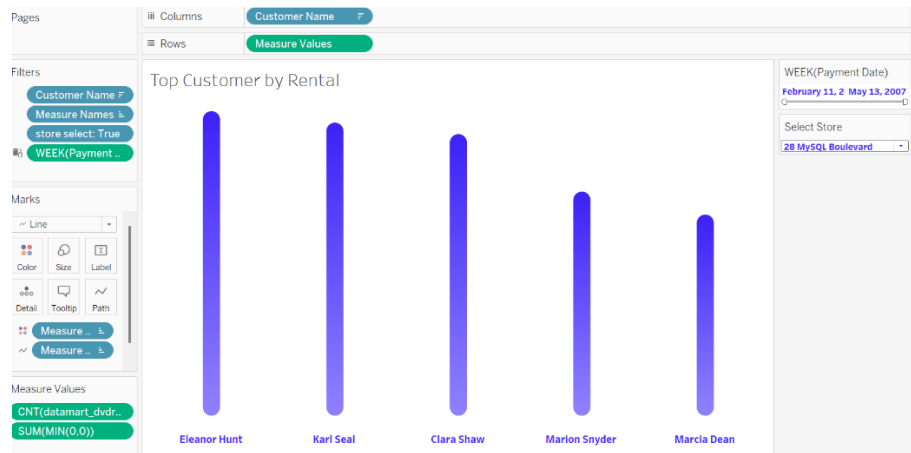


Gambar 3. 51 Rental Trend Toko 28 MySQL Boulevard

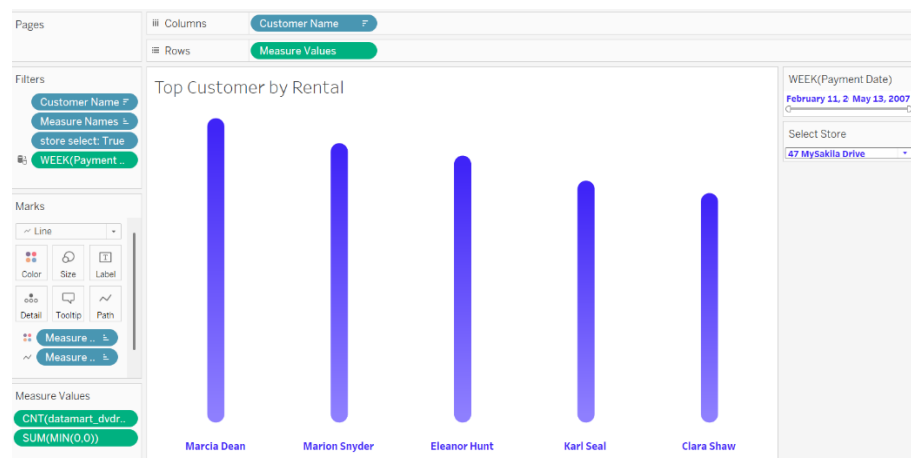


Gambar 3. 52 Rental Trend Toko 47 MySakila Drive

7. Kemudian, buat grafik dengan nama Top Customer by Rental. Grafik ini menggunakan jenis visualisasi *Vertical Bar Chart*. Top Customer by Rental menunjukkan 5 pelanggan teratas dengan jumlah transaksi penyewaan DVD terbanyak selama periode 11 Februari 2007 hingga 13 Mei 2007. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.53 dan Gambar 3.54.

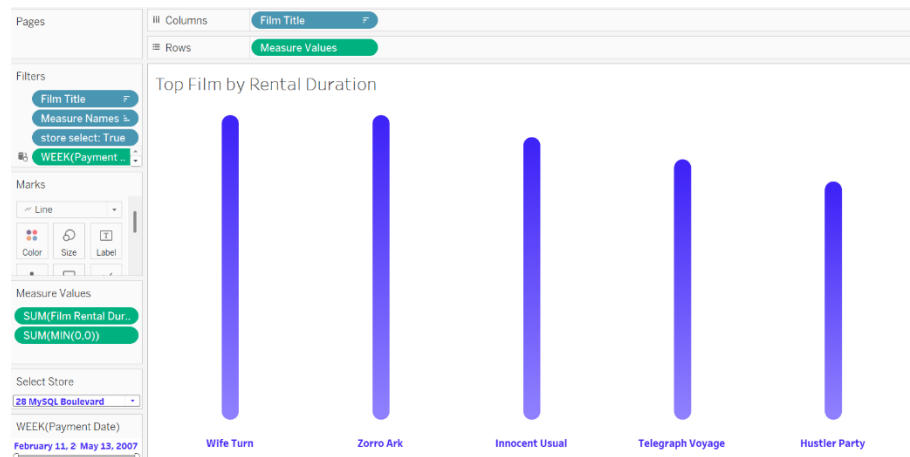


Gambar 3. 53 Top Customer by Rental Toko 28 MySQL Boulevard

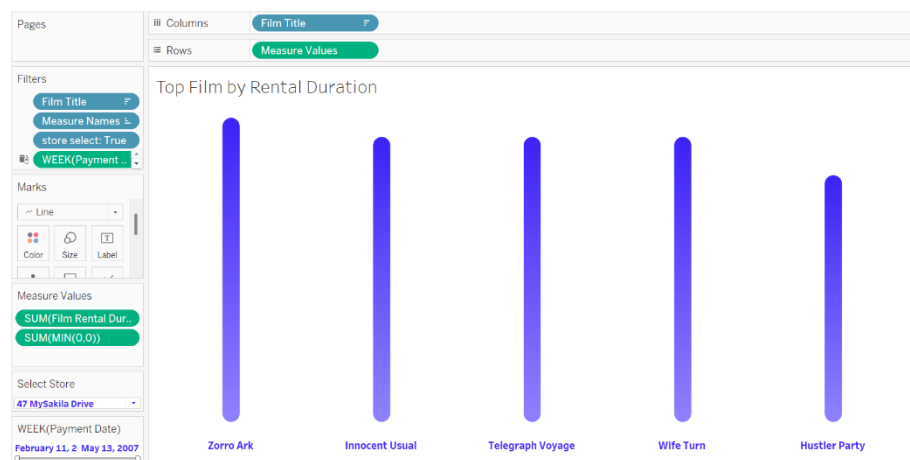


Gambar 3. 54 Top Customer by Rental Toko 47 MySakila Drive

8. Kemudian, buat grafik dengan nama Top Film by Rental Duration. Grafik ini menggunakan jenis visualisasi *Vertical Bar Chart*. Top Film by Rental Duration menunjukkan 5 film teratas dengan durasi penyewaan DVD terlama selama periode 11 Februari 2007 hingga 13 Mei 2007. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.55 dan Gambar 3.56.

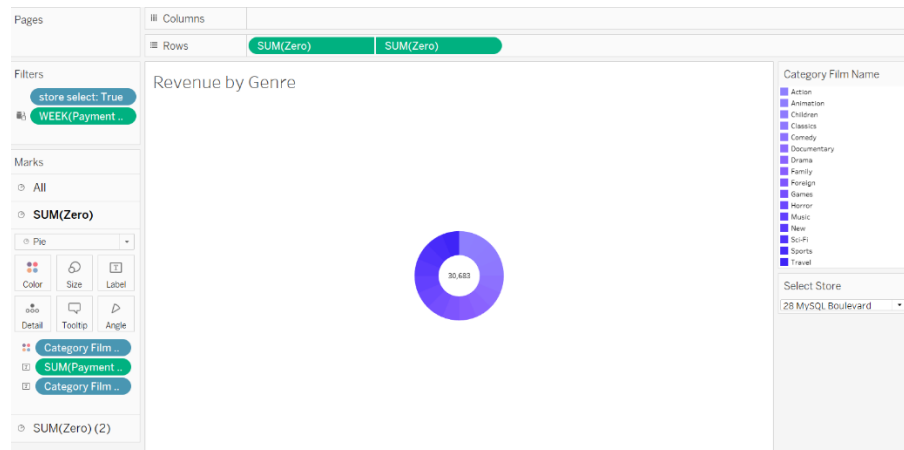


Gambar 3. 55 Top Film by Rental Duration Toko 28 MySQL Boulevard

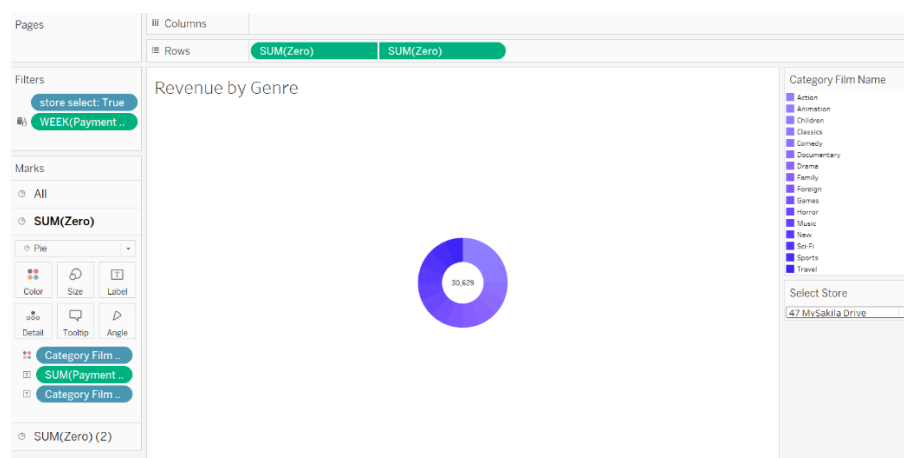


Gambar 3. 56 Top Film by Rental Duration Toko 47 MySakila Drive

9. Kemudian, buat grafik dengan nama Revenue by Genre. Grafik ini menggunakan jenis visualisasi *Donut Chart*. Revenue by Genre menunjukkan total pendapatan dari masing-masing kategori film selama periode 11 Februari 2007 hingga 13 Mei 2007. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.57 dan Gambar 3.58.

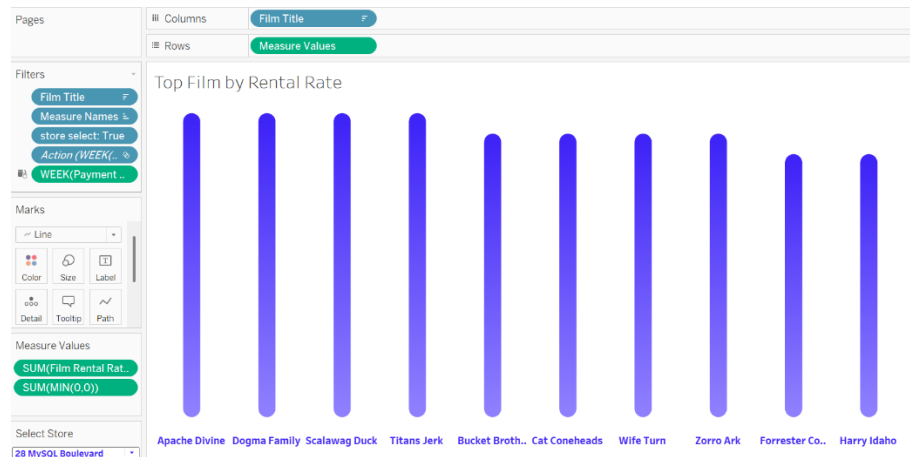


Gambar 3. 57 Revenue by Genre Toko 28 MySQL Boulevard

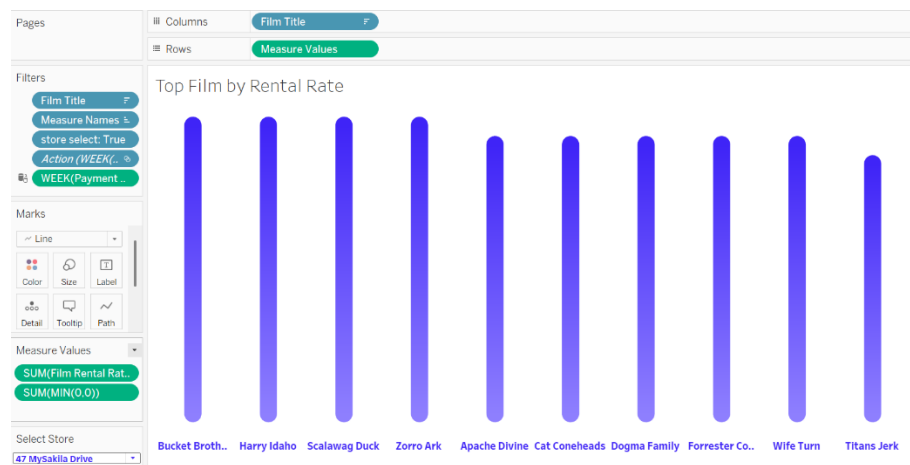


Gambar 3. 58 Revenue by Genre Toko 47 MySakila Drive

10. Kemudian, buat grafik dengan nama Top Film by Rental Rate. Grafik ini menggunakan jenis visualisasi *Vertical Bar Chart*. Top Film by Rental Rate menunjukkan 10 film dengan tarif penyewaan tertinggi selama periode 11 Februari 2007 hingga 13 Mei 2007. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.59 dan Gambar 3.60.

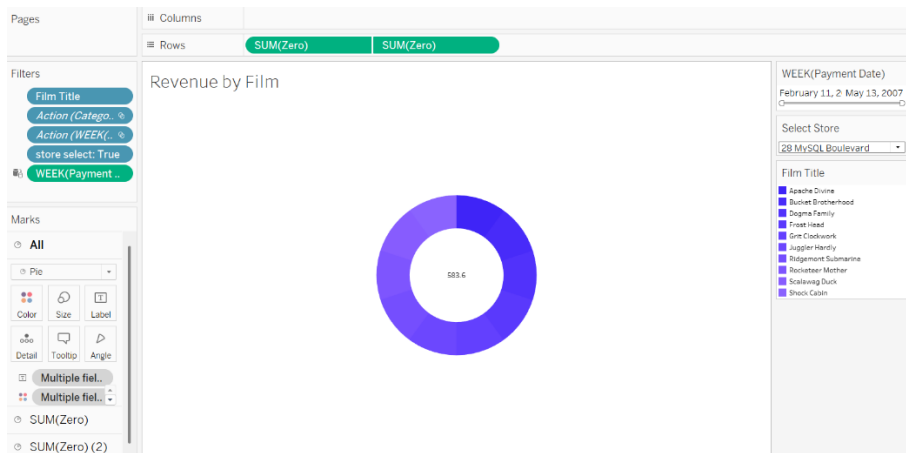


Gambar 3. 59 Top Film by Rental Rate Toko 28 MySQL Boulevard

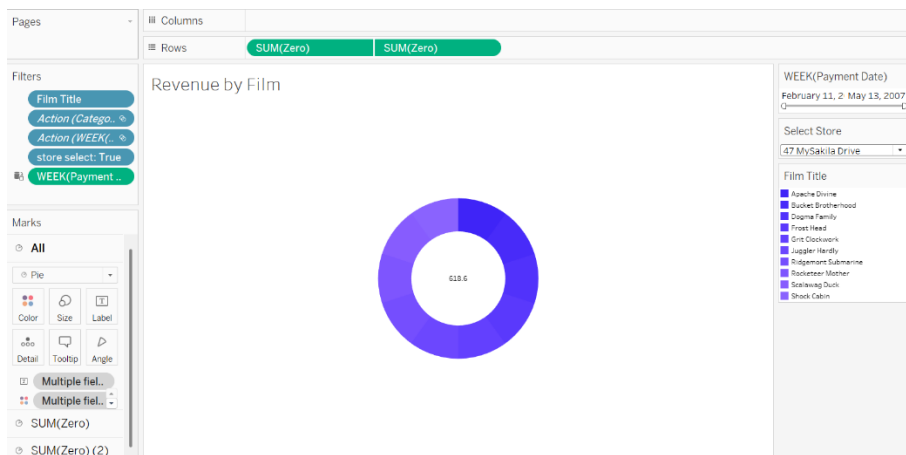


Gambar 3. 60 Top Film by Rental Rate Toko 47 MySakila Drive

11. Kemudian, buat grafik dengan nama Revenue by Film. Grafik ini menggunakan jenis visualisasi *Donut Chart*. Revenue by Film menunjukkan total pendapatan dari masing-masing judul film selama periode 11 Februari 2007 hingga 13 Mei 2007. Grafik ini mencakup data dari dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive. Dapat dilihat pada Gambar 3.61 dan Gambar 3.62.



Gambar 3. 61 Revenue by Film Toko 28 MySQL Boulevard



Gambar 3. 62 Revenue by Film Toko 47 MySakila Drive

- **Pembuatan *Dashboard 1* (DVD Rental Summary)**

1. *Dashboard* DVD Rental Summary ini dirancang untuk menyajikan analisis mendalam mengenai performa operasional toko DVD Rental, dengan menampilkan berbagai indikator kunci yang memberikan wawasan komprehensif tentang tren penyewaan, pendapatan, dan kinerja toko secara keseluruhan.
2. *Dashboard* ini memiliki fitur Filter Store untuk memilih dua toko DVD Rental yang berlokasi di negara berbeda, yaitu Toko 28 MySQL Boulevard dan Toko 47 MySakila Drive, serta ada Filter Date untuk mengatur periode waktu antara 11 Februari 2007 hingga 13 Mei 2007.
3. Terdapat 9 informasi yang ditampilkan dalam *dashboard* ini, diantaranya: Total Revenue, Total Customer, Total Staff, Total Film, Total Genre, Rental Trend, Top Customer by Rental, Top Film by Rental Duration, dan Revenue by Genre.
4. Pada grafik Revenue by Genre, jika salah satu *genre* diklik, maka pengguna akan langsung terhubung ke *dashboard 2* untuk analisis lebih detail tentang *genre* tersebut.
5. *Dashboard* DVD Rental Summary dapat dilihat pada Gambar 3.63 dan Gambar 3.64.



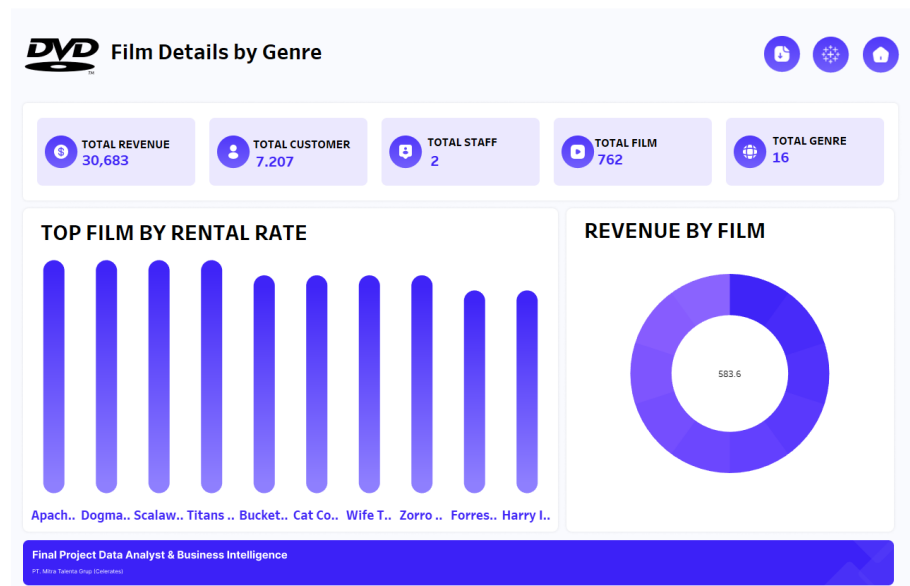
Gambar 3. 63 Dashboard DVD Rental Summary Toko 28 MySQL Boulevard



Gambar 3. 64 Dashboard DVD Rental Summary Toko 47 MySakila Drive

- **Pembuatan Dashboard 2 (Film Details by Genre)**

1. *Dashboard* Film Details by Genre ini dirancang untuk memberikan analisis lebih mendalam mengenai performa film berdasarkan *genre* yang dipilih, dengan fokus pada pendapatan film dan tingkat penyewaan film secara spesifik.
2. Terdapat 7 informasi yang ditampilkan dalam *dashboard* ini, diantaranya: Total Revenue, Total Customer, Total Staff, Total Film, Total Genre, Top Film by Rental Rate, dan Revenue by Film.
3. *Dashboard* Film Details by Genre dapat dilihat pada Gambar 3.65.



Gambar 3. 65 *Dashboard* Film Details by Genre

BAB 4

PENUTUP

4.1. Kesimpulan

Kesimpulan mencakup hasil akhir *project* dan manfaat yang diraih dari sisi teknis dan bisnis, memberikan gambaran lengkap tentang pencapaian dan dampak *project* ini.

- **Sisi Teknis**

1. *Database* dvdrental berhasil dianalisis untuk memahami pola pelanggan dan operasional toko, menghasilkan data yang terstruktur dan siap diolah lebih lanjut.
2. Proses analisis data *end-to-end* diterapkan, melibatkan *staging area*, *data warehouse*, dan visualisasi data secara efektif.
3. *Dashboard* interaktif dirancang untuk menyajikan hasil analisis secara intuitif, memudahkan interpretasi oleh pengambil keputusan.
4. Penggunaan *tools* seperti PostgreSQL, DBeaver, Pentaho, dan Tableau memperkuat penerapan teknologi untuk menyelesaikan masalah nyata.
5. *Project* ini meningkatkan pemahaman tentang integrasi sistem data serta penerapan analitik untuk mendukung pengambilan keputusan berbasis data.

- **Sisi Bisnis**

1. *Genre* film terpopuler di tiap lokasi toko berhasil diidentifikasi, mendukung pengelolaan stok yang lebih sesuai dengan permintaan pelanggan.
2. Informasi berbasis data memberikan landasan untuk merancang koleksi film yang relevan dan meningkatkan kepuasan pelanggan.
3. Hasil analisis membantu menyusun strategi pemasaran berbasis data untuk mempromosikan *genre* favorit di masing-masing wilayah.
4. Efisiensi pengelolaan inventaris meningkat, mengurangi biaya operasional dan memaksimalkan pendapatan.
5. Wawasan yang diperoleh mendukung pengambilan keputusan strategis untuk memperkuat daya saing di pasar lokal dan internasional.

4.2. Saran

Sebagai tindak lanjut dari hasil *project* ini, terdapat beberapa saran yang diharapkan dapat mendukung pengembangan lebih lanjut. Saran-saran tersebut difokuskan pada aspek teknis dan penerapan analisis data, yang berperan penting dalam meningkatkan efisiensi serta efektivitas hasil *project* di masa mendatang.

- **Sisi Teknis**

1. Kembangkan sistem untuk memonitor stok film secara otomatis berdasarkan data penjualan.
2. Implementasikan integrasi data *real-time* untuk meng-*update* informasi inventaris dan permintaan pelanggan.
3. Gunakan *tools* visualisasi yang lebih interaktif untuk memudahkan pembacaan data, seperti menyertakan grafik prediktif.

- **Penerapan Analisis Data**

1. Buat rekomendasi film otomatis untuk pelanggan berdasarkan pola sewa mereka.
2. Gunakan data preferensi *genre* untuk membuat paket promosi menarik, seperti diskon khusus untuk *genre* tertentu.
3. Manfaatkan analisis ini untuk meningkatkan koleksi film klasik atau langka yang sulit ditemukan di pasar *streaming*.

DAFTAR PUSTAKA

- Colosimo, M. A. (2018). *A Data-Driven And Mixed Methods Analysis Of Automotive Retail Operations Management*.
- Kamaliya, P., Sabaniah, F., Lutpi, M., & Halawa, E. M. (2024). Perancangan M-Dashboard Pemantauan Project Approval Request (PAR) Operasional Dengan Metode Rapid Application Development (RAD) (Studi Kasus : PT Global Evolusi Teknologi). *Jurnal Ilmu Komputer, Teknik, Dan Multimedia*, 2(3), 1–9.
- Marvaro, E., & Sefina Samosir, R. (2021). Penerapan Business Intelligence dan Visualisasi Informasi di CV. Mitra Makmur Dengan Menggunakan Dashboard Tableau. *KALBISCIENTIA Jurnal Sains Dan Teknologi*, 8(2), 37–46.
<https://doi.org/10.53008/kalbiscientia.v8i2.197>
- Saraswati, N. W. S., & Martarini, N. M. L. (2020). Extract Transform Loading Data Absensi Stmik Stikom Indonesia Menggunakan Pentaho. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 19(2), 273–281.
<https://doi.org/10.30812/matrik.v19i2.564>
- Sri Wahyuningsih, A., & Junianto, S. (2020). Penerapan Knowledge Management Sistem Layanan Perbaikan Device Distribution Center (Dc) Support Pt.Indomarco Prismaatama Cabang Bekasi. *Jurnal Informatika SIMANTIK*, 5(1), 47–51.
www.jurnal.stmikcikarang.ac.id
- Sudjiran, S., Saefudin, M., & Perdana, S. A. (2023). Digital System Ui/Ux Design Management Submission of Agricultural Cost Loans Using Figma Software. *Journal of Information System, Informatics and Computing*, 7(1), 74.
<https://doi.org/10.52362/jisicom.v7i1.1090>

LAMPIRAN

Lampiran 1 *Script Query Database staging_dvdrental*

```
-- 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,
    CONSTRAINT category_pkey PRIMARY KEY (category_id)
);

-- public.country definition

-- Drop table
```

```

-- DROP TABLE public.country;

CREATE TABLE public.country (
    country_id serial4 NOT NULL,
    country varchar(50) NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT country_pkey PRIMARY KEY (country_id)
);

-- public."language" definition

-- Drop table

-- DROP TABLE public."language";

CREATE TABLE public."language" (
    language_id serial4 NOT NULL,
    "name" bpchar(20) NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT language_pkey PRIMARY KEY (language_id)
);

-- public.city definition

-- Drop table

-- DROP TABLE public.city;

CREATE TABLE public.city (
    city_id serial4 NOT NULL,
    city varchar(50) NOT NULL,
    country_id int2 NOT NULL,

```

```

        last_update timestamp DEFAULT now() NOT NULL,
        CONSTRAINT city_pkey PRIMARY KEY (city_id),
        CONSTRAINT fk_city FOREIGN KEY (country_id) REFERENCES
public.country(country_id)
);
CREATE INDEX idx_fk_country_id ON public.city USING btree (country_id);

-- public.film definition

-- Drop table

-- DROP TABLE public.film;

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

```



```

-- public.film_actor definition

-- Drop table

-- DROP TABLE public.film_actor;

CREATE TABLE public.film_actor (
    actor_id int2 NOT NULL,
    film_id int2 NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT film_actor_pkey PRIMARY KEY (actor_id, film_id),
    CONSTRAINT film_actor_actor_id_fkey FOREIGN KEY (actor_id) REFERENCES
public.actor(actor_id) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT film_actor_film_id_fkey FOREIGN KEY (film_id) REFERENCES
public.film(film_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

CREATE INDEX idx_fk_film_id ON public.film_actor USING btree (film_id);

-- public.film_category definition

-- Drop table

-- DROP TABLE public.film_category;

CREATE TABLE public.film_category (
    film_id int2 NOT NULL,
    category_id int2 NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT film_category_pkey PRIMARY KEY (film_id, category_id),
    CONSTRAINT film_category_category_id_fkey FOREIGN KEY (category_id)
REFERENCES public.category(category_id) ON DELETE RESTRICT ON UPDATE
CASCADE,
    CONSTRAINT film_category_film_id_fkey FOREIGN KEY (film_id) REFERENCES
public.film(film_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

```

```

-- public.inventory definition

-- Drop table

-- DROP TABLE public.inventory;

CREATE TABLE public.inventory (
    inventory_id serial4 NOT NULL,
    film_id int2 NOT NULL,
    store_id int2 NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT inventory_pkey PRIMARY KEY (inventory_id),
    CONSTRAINT inventory_film_id_fkey FOREIGN KEY (film_id) REFERENCES
public.film(film_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

CREATE INDEX idx_store_id_film_id ON public.inventory USING btree (store_id,
film_id);

-- public.address definition

-- Drop table

-- DROP TABLE public.address;

CREATE TABLE public.address (
    address_id serial4 NOT NULL,
    address varchar(50) NOT NULL,
    address2 varchar(50) NULL,
    district varchar(20) NOT NULL,
    city_id int2 NOT NULL,
    postal_code varchar(10) NULL,
    phone varchar(20) NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT address_pkey PRIMARY KEY (address_id),

```

```

        CONSTRAINT fk_address_city FOREIGN KEY (city_id) REFERENCES
public.city(city_id)
);

CREATE INDEX idx_fk_city_id ON public.address USING btree (city_id);

-- public.customer definition

-- Drop table

-- DROP TABLE public.customer;

CREATE TABLE public.customer (
    customer_id serial4 NOT NULL,
    store_id int2 NOT NULL,
    first_name varchar(45) NOT NULL,
    last_name varchar(45) NOT NULL,
    email varchar(50) NULL,
    address_id int2 NOT NULL,
    activebool bool DEFAULT true NOT NULL,
    create_date date DEFAULT 'now'::text::date NOT NULL,
    last_update timestamp DEFAULT now() NULL,
    active int4 NULL,
    CONSTRAINT customer_pkey PRIMARY KEY (customer_id),
    CONSTRAINT customer_address_id_fkey FOREIGN KEY (address_id)
REFERENCES public.address(address_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

CREATE INDEX idx_fk_address_id ON public.customer USING btree (address_id);
CREATE INDEX idx_fk_store_id ON public.customer USING btree (store_id);
CREATE INDEX idx_last_name ON public.customer USING btree (last_name);

-- public.staff definition

-- Drop table

-- DROP TABLE public.staff;

```

```

CREATE TABLE public.staff (
    staff_id serial4 NOT NULL,
    first_name varchar(45) NOT NULL,
    last_name varchar(45) NOT NULL,
    address_id int2 NOT NULL,
    email varchar(50) NULL,
    store_id int2 NOT NULL,
    active bool DEFAULT true NOT NULL,
    username varchar(16) NOT NULL,
    "password" varchar(40) NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    picture bytea NULL,
    CONSTRAINT staff_pkey PRIMARY KEY (staff_id),
    CONSTRAINT staff_address_id_fkey FOREIGN KEY (address_id) REFERENCES
public.address(address_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

-- public.store definition

-- Drop table

-- DROP TABLE public.store;

CREATE TABLE public.store (
    store_id serial4 NOT NULL,
    manager_staff_id int2 NOT NULL,
    address_id int2 NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT store_pkey PRIMARY KEY (store_id),
    CONSTRAINT store_address_id_fkey FOREIGN KEY (address_id) REFERENCES
public.address(address_id) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT store_manager_staff_id_fkey FOREIGN KEY (manager_staff_id)
REFERENCES public.staff(staff_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

```

```

CREATE UNIQUE INDEX idx_unq_manager_staff_id ON public.store USING btree
(manager_staff_id);

-- public.rental definition

-- Drop table

-- DROP TABLE public.rental;

CREATE TABLE public.rental (
    rental_id serial4 NOT NULL,
    rental_date timestamp NOT NULL,
    inventory_id int4 NOT NULL,
    customer_id int2 NOT NULL,
    return_date timestamp NULL,
    staff_id int2 NOT NULL,
    last_update timestamp DEFAULT now() NOT NULL,
    CONSTRAINT rental_pkey PRIMARY KEY (rental_id),
    CONSTRAINT rental_customer_id_fkey FOREIGN KEY (customer_id)
REFERENCES public.customer(customer_id) ON DELETE RESTRICT ON UPDATE
CASCADE,
    CONSTRAINT rental_inventory_id_fkey FOREIGN KEY (inventory_id)
REFERENCES public.inventory(inventory_id) ON DELETE RESTRICT ON UPDATE
CASCADE,
    CONSTRAINT rental_staff_id_key FOREIGN KEY (staff_id) REFERENCES
public.staff(staff_id)
);

CREATE INDEX idx_fk_inventory_id ON public.rental USING btree
(inventory_id);

CREATE UNIQUE INDEX idx_unq_rental_rental_date_inventory_id_customer_id ON
public.rental USING btree (rental_date, inventory_id, customer_id);

-- public.payment definition

-- Drop table

-- DROP TABLE public.payment;

```

```

CREATE TABLE public.payment (
    payment_id serial4 NOT NULL,
    customer_id int2 NOT NULL,
    staff_id int2 NOT NULL,
    rental_id int4 NOT NULL,
    amount numeric(5, 2) NOT NULL,
    payment_date timestamp NOT NULL,
    CONSTRAINT payment_pkey PRIMARY KEY (payment_id),
    CONSTRAINT payment_customer_id_fkey FOREIGN KEY (customer_id)
REFERENCES public.customer(customer_id) ON DELETE RESTRICT ON UPDATE
CASCADE,
    CONSTRAINT payment_rental_id_fkey FOREIGN KEY (rental_id) REFERENCES
public.rental(rental_id) ON DELETE SET NULL ON UPDATE CASCADE,
    CONSTRAINT payment_staff_id_fkey FOREIGN KEY (staff_id) REFERENCES
public.staff(staff_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

CREATE INDEX idx_fk_customer_id ON public.payment USING btree (customer_id);
CREATE INDEX idx_fk_rental_id ON public.payment USING btree (rental_id);
CREATE INDEX idx_fk_staff_id ON public.payment USING btree (staff_id);

```

Lampiran 2 *Script Query Schemas* log

```

-- log.log_tabel_masuk definition
-- Drop table
-- DROP TABLE log.log_tabel_masuk;
CREATE TABLE log.log_tabel_masuk (
    table_name varchar(100) NULL,
    input_date timestamp NULL
);

-- log.staging_log definition
-- Drop table
-- DROP TABLE log.staging_log;
CREATE TABLE log.staging_log (
    table_name varchar(100) NULL,
    source_records varchar(100) NULL,
    source_date timestamp NULL,
    target_records varchar(100) NULL,

```

```
target_date timestamp NULL,  
etl_date_start timestamp NULL,  
logical varchar(100) NULL  
);
```