

LAPORAN KONVERSI DATA WAREHOUSE & OLAP
“PENGEMBANGAN DATA WAREHOUSE DALAM ANALISIS
DATA DVD RENTAL”
SEMESTER GANJIL 2024/2025

Dosen Pengampu

Mohamad Irwan Afandi, ST., MSC.



Disusun Oleh

Muhammad Rakha Naufal (22082010060)

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

Universitas Pembangunan Nasional “Veteran” Jawa Timur

S U R A B A Y A

2024

DAFTAR ISI

DAFTAR ISI	2
BAB I PENDAHULUAN	3
1.1. Latar Belakang	3
1.2. Rumusan Masalah	4
1.3. Batasan	4
BAB II METODE	5
2.1. Metode Pengembangan	5
2.1.1. Perencanaan	5
2.1.2. Perancangan Skema	5
2.1.3. Ekstraksi Data	6
2.1.4. Pembuatan CUBE Mondrian di Tomcat	6
BAB III HASIL DAN PEMBAHASAN	7
3.1. Implementasi Data Warehouse	7
3.1.1 Dimensi Customer	7
3.1.2 Dimensi Film	8
3.1.3 Dimensi Payment	8
3.1.4 Dimensi Rental Date	9
3.1.5 Dimensi staff	10
3.1.6 Dimensi Store	11
3.1.7 Tabel Fakta	12
3.2. Implementasi Cube dan OLAP	13
3.2.1 Customer	15
3.2.2 Film	15
3.2.3 Payment	16
3.2.4 Rental Date	17
3.2.5 Staff	18
3.2.6 Store	19
LAMPIRAN	20
Link Github: https://github.com/rakhanaufalllll/KonversiDWO_MuhammadRakhaNaufal_060	20

BAB I PENDAHULUAN

1.1. Latar Belakang

Data Warehouse adalah sistem penyimpanan data yang dirancang untuk mengumpulkan, mengintegrasikan, dan mengorganisasikan data dari berbagai sumber dalam suatu organisasi. Sistem ini berperan penting dalam mendukung proses pengambilan keputusan strategis dengan menyediakan data yang terstruktur, terintegrasi, dan mudah diakses. Dengan meningkatnya kebutuhan akan analisis data yang komprehensif, pengembangan Data Warehouse menjadi salah satu solusi yang paling efektif untuk menghadirkan informasi relevan bagi manajemen dalam memahami tren bisnis dan mengoptimalkan kinerja operasional.

Dalam proyek ini, DVD Rental Database digunakan sebagai sumber data. Database ini merupakan basis data fiktif yang sering digunakan dalam studi dan demonstrasi pengolahan data. Database ini dirancang untuk merepresentasikan operasi bisnis dari sebuah perusahaan penyewaan DVD yang memiliki pelanggan, koleksi film, kategori film, dan catatan transaksi penyewaan yang lengkap. Beberapa tabel utama dalam database ini meliputi City, Country, Customer, Film, Film Category, Inventory, Language, dan Rental.

Dalam analisis data penjualan, terutama yang berkaitan dengan transaksi penyewaan film, pengolahan data yang efisien dan dapat memberikan wawasan yang lebih mendalam sangat penting bagi pengambilan keputusan bisnis yang tepat. Salah satu solusi yang dapat digunakan untuk mempermudah proses ini adalah menggunakan OLAP (Online Analytical Processing), yang memungkinkan analisis data multidimensional secara cepat dan interaktif.

Mondrian OLAP merupakan salah satu platform yang populer digunakan untuk tujuan ini, memungkinkan pengguna untuk melakukan analisis data berbasis multidimensional. Proyek ini akan mengembangkan tampilan query yang menghubungkan beberapa dimensi yang relevan dengan analisis penjualan, seperti Customer, Film, Payment, Rental Date, Staff, dan Store, yang semuanya terhubung melalui tabel fakta.

Dimensi-dimensi ini akan diorganisasikan dalam bentuk data yang komprehensif, yang memungkinkan pengguna untuk mengeksplorasi data dari berbagai sudut pandang, misalnya, melihat performa penjualan berdasarkan jenis pelanggan, kategori film, cara pembayaran, waktu penyewaan, serta lokasi atau staff yang terlibat dalam transaksi. Dengan demikian, data dapat dianalisis lebih mendalam untuk memahami tren dan pola yang muncul dalam bisnis penyewaan film.

Measure utama yang digunakan dalam analisis ini adalah Total Amount, yang mewakili total pendapatan yang dihasilkan dari transaksi penyewaan film. Total Amount dihitung berdasarkan informasi yang terdapat dalam tabel fakta yang mencakup rincian setiap transaksi, termasuk data

transaksi terkait pelanggan, film yang disewa, pembayaran yang dilakukan, tanggal penyewaan, staff yang memproses transaksi, dan store tempat transaksi dilakukan.

Melalui penggunaan Mondrian OLAP, query ini memungkinkan pengguna untuk mendapatkan gambaran yang lebih jelas mengenai faktor-faktor yang mempengaruhi pendapatan, serta bagaimana pendapatan tersebut tersebar berdasarkan dimensi yang berbeda. Dengan informasi ini, manajemen dapat membuat keputusan yang lebih tepat dalam merencanakan strategi bisnis, meningkatkan layanan pelanggan, atau mengoptimalkan operasional penyewaan.

1.2. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah bagaimana menganalisis total pendapatan (Total Amount) dari transaksi penyewaan film berdasarkan dimensi Customer, Film, Payment, Rental Date, Staff, dan Store. Penelitian ini bertujuan untuk menjawab pertanyaan-pertanyaan mengenai film yang paling menguntungkan, distribusi pendapatan berdasarkan kategori pelanggan dan cara pembayaran, serta bagaimana tanggal penyewaan, lokasi store, dan staff mempengaruhi pendapatan. Selain itu, penelitian ini juga akan mengeksplorasi penggunaan Mondrian OLAP untuk mempermudah dan mempercepat analisis data multidimensional, sehingga dapat memberikan wawasan yang lebih mendalam untuk mendukung pengambilan keputusan bisnis.

1.3. Batasan

Batasan penelitian ini mencakup analisis data yang hanya menggunakan tabel fakta yang berhubungan dengan transaksi penyewaan film, yaitu tabel Rental dan Payment. Dimensi yang dianalisis meliputi Customer, Film, Payment, Rental Date, Staff, dan Store, dengan fokus pada penghitungan Total Amount sebagai measure utama. Data yang digunakan terbatas pada periode yang tercatat dalam basis data dan hanya mencakup transaksi yang valid. Selain itu, pengembangan analisis ini difokuskan pada penggunaan Mondrian OLAP untuk query multidimensional, tanpa memperhitungkan faktor eksternal lain seperti promosi atau faktor pasar yang dapat mempengaruhi hasil analisis.

BAB II METODE

2.1. Metode Pengembangan

2.1.1. Perencanaan

Tahap perencanaan dimulai dengan pemahaman mendalam tentang tujuan proyek, yang berfokus pada analisis data transaksi penyewaan film menggunakan Mondrian OLAP untuk menghasilkan laporan penjualan berbasis Total Amount. Pada tahap ini, langkah pertama adalah mendefinisikan dimensi dan measure yang relevan, seperti Customer, Film, Payment, Rental Date, Staff, dan Store, serta memastikan ketersediaan data yang diperlukan dalam database.

Perencanaan juga mencakup penentuan skema OLAP yang akan digunakan, yaitu star schema, yang memisahkan data transaksi ke dalam tabel fakta dan dimensi. Penggunaan star schema bertujuan untuk menyederhanakan struktur data sehingga memudahkan proses ekstraksi dan analisis data. Selain itu, pada tahap perencanaan juga dilakukan penentuan alat dan teknologi yang digunakan, yakni Mondrian OLAP untuk mengelola query multidimensional dan visualisasi data, serta perangkat lunak untuk mengakses dan mengelola database, seperti Pentaho dan SQL.

Setelah itu, dilakukan identifikasi kebutuhan bisnis dan pengguna yang akan menggunakan dashboard hasil analisis, serta menetapkan tujuan laporan yang akan dihasilkan, seperti laporan pendapatan berdasarkan kategori pelanggan, film terpopuler, serta distribusi pendapatan per store dan staff. Tahap perencanaan juga melibatkan pembuatan jadwal untuk setiap tahap pengembangan, pengujian, dan implementasi, serta menentukan sumber daya yang diperlukan untuk mendukung pengembangan proyek ini.

2.1.2. Perancangan Skema

Pada tahap perancangan skema, model star schema dipilih untuk mendesain data warehouse DVD Rental Database karena strukturnya yang sederhana dan efisien untuk analisis multidimensional. Tabel fact_rental berfungsi sebagai tabel fakta utama yang menyimpan data transaksi penyewaan, dengan measure total_amount sebagai ukuran utama. Tabel dimensi yang terkait meliputi Customer (informasi pelanggan), Film (data film yang disewa), Payment (metode pembayaran), Rental Date (tanggal transaksi penyewaan), Staff (data staf yang memproses transaksi), dan Store (lokasi store). Setiap dimensi dihubungkan ke tabel fakta melalui foreign key yang memudahkan analisis berdasarkan berbagai perspektif, seperti pelanggan, film, tanggal, dan lokasi, serta memungkinkan pembuatan query OLAP yang efisien untuk menganalisis pendapatan dan kinerja bisnis.

2.1.3. Ekstraksi Data

Proses ekstraksi data dimulai dengan pengambilan data dari DVD Rental Database untuk mengisi tabel fakta dan dimensi dalam data warehouse. Langkah pertama dalam ekstraksi ini adalah melakukan ETL (Extract, Transform, Load) menggunakan Pentaho untuk mengolah data yang ada dalam database sumber.

Pada tahap Extract, data diambil dari berbagai tabel dalam DVD Rental Database yang berisi informasi terkait transaksi penyewaan, pelanggan, film, pembayaran, dan lainnya. Data ini mencakup tabel seperti rental, customer, payment, film, staff, dan store. Pentaho digunakan untuk mengakses dan mengimpor data mentah ini ke dalam sistem ETL.

Pada tahap Transform, data yang diambil akan dibersihkan dan disiapkan untuk dimasukkan ke dalam skema star schema. Proses transformasi ini meliputi penghapusan duplikasi, konversi format data (misalnya, mengubah format tanggal menjadi standar yang lebih mudah dianalisis), serta pembuatan kolom baru atau kalkulasi yang diperlukan (seperti total amount pada tabel fakta). Selain itu, data yang diambil dari berbagai sumber ini juga akan digabungkan melalui proses join untuk membentuk tabel fakta dan dimensi yang sesuai.

Tahap terakhir adalah Load, di mana data yang telah diproses akan dimuat ke dalam data warehouse. Tabel fakta fact_rental dan tabel dimensi seperti dim_customer, dim_film, dim_payment, dim_rental_date, dim_staff, dan dim_store akan diisi dengan data yang telah diekstrak dan ditransformasikan. Proses ini memastikan bahwa data yang ada dalam data warehouse sudah siap untuk dianalisis menggunakan Mondrian OLAP dan menghasilkan laporan yang relevan bagi pengambilan keputusan bisnis.

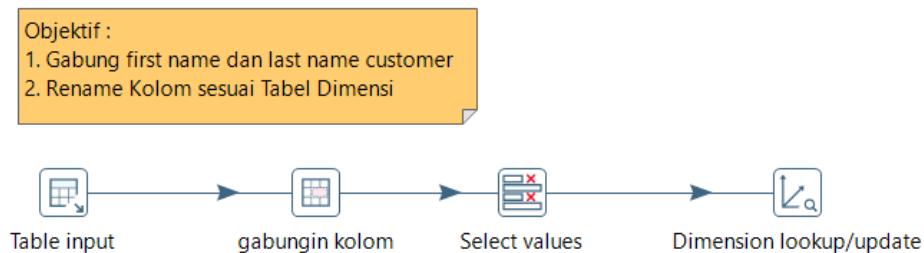
2.1.4. Pembuatan CUBE Mondrian di Tomcat

Mondrian adalah salah satu solusi OLAP (Online Analytical Processing) berbasis Java yang memungkinkan pengguna untuk membuat dan menjalankan cube untuk analisis data multidimensional. Adapun langkah-langkah untuk membuat Cube Mondrian di Tomcat seperti mempersiapkan tools (Tomcat, Mondrian, JRE, dan Pentaho), menyiapkan file Mondrian schema, konfigurasikan Mondrian di tomcat, menghubungkan Mondrian dengan database, restart Tomcat untuk menampilkan perubahan, dan mengakses cube melalui <http://localhost:8080/mondrian>. Setelah cube berhasil dibuat, maka dapat menjalankan query OLAP untuk menganalisis data transaksi penyewaan berdasarkan dimensi yang telah ditentukan.

BAB III HASIL DAN PEMBAHASAN

3.1. Implementasi Data Warehouse

3.1.1 Dimensi Customer



Gambar 3.1.1.1 ETL Customer

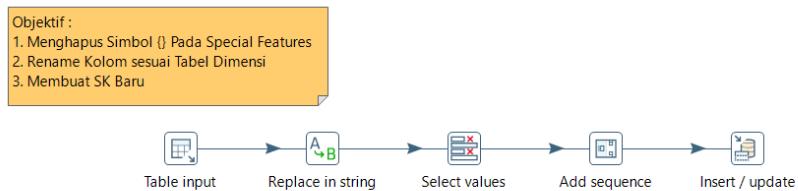
customer_id	customer_name	customer_address	customer_city	customer_state	customer_postal_code	customer_phone	customer_email	customer_create_date	customer_last_update
1	Purnawarta	Jl. Puri Indah No. 123	Surabaya	Jawa Timur	6531119997	25972	08123456789	2006/02/14 00:00:00.000	2013/05/26 14:49:45.738C
2	Sasebo	Jl. Sasebo 123	Nagasaki	Japan	28303384290	35200	08123456789	2006/02/14 00:00:00.000	2013/05/26 14:49:45.738C
3	San Bernardino	Jl. San Bernardino 123	California	United States	838615266649	17886	08123456789	2006/02/14 00:00:00.000	2013/05/26 14:49:45.738C
4	Athenai	Jl. Athenai 123	Attika	Greece	448477190408	83579	08123456789	2006/02/14 00:00:00.000	2013/05/26 14:49:45.738C
5	Myinyan	Jl. Myinyan 123	Mandalay	Myanmar	705814003527	53561	08123456789	2006/02/14 00:00:00.000	2013/05/26 14:49:45.738C
6	Nantou	Jl. Nantou 123	Nantou	Taiwan	10655648674	42399	08123456789	2006/02/14 00:00:00.000	2013/05/26 14:49:45.738C
7	Laredo	Jl. Laredo 123	Texas	United States	860452626434	18743	08123456789	2006/02/14 00:00:00.000	2013/05/26 14:49:45.738C
8	Kragujevac	Jl. Kragujevac 123	Yugoslavia	Central Serbia	71657120373	93896	08123456789	2006/02/14 00:00:00.000	2013/05/26 14:49:45.738C
9	Urawa	Jl. Urawa 123	Urawa	New Zealand	44776789099	77040	08123456789	2006/02/14 00:00:00.000	2013/05/26 14:49:45.738C

Gambar 3.1.1.2 Running Pentaho

Langkah pertama adalah membuat koneksi ke database, kemudian menyambungkan target schema dengan tabel customer yang telah melalui proses staging. Dalam proses ini, dilakukan JOIN dengan tabel address, city, dan country untuk tujuan tertentu. Address menyimpan data detail alamat pelanggan, city berisi informasi mengenai kota dari alamat tersebut, sedangkan country memuat informasi negara dari kota tersebut. Penggunaan JOIN ini bertujuan untuk memperkaya data pelanggan dengan informasi yang lebih lengkap sekaligus menghindari redudansi data. Contohnya, address digunakan untuk detail alamat pelanggan, city_name untuk nama kota, dan country_name untuk nama negara.

Kolom first_name dan last_name pada tabel customer digabungkan menjadi satu menggunakan fungsi Concat Field untuk menciptakan kolom nama lengkap. Selanjutnya, langkah Select Value digunakan untuk mengubah nama tabel agar sesuai dengan struktur dimensi yang diinginkan. Pada tahap akhir, modul Dimension Lookup/Update digunakan untuk membuat ID baru (sk_customer) pada tabel dimensi customer. Sebelum menjalankan proses di Pentaho, SQL script harus dieksekusi terlebih dahulu untuk memastikan schema siap digunakan. Setelah itu, jalankan ETL di Pentaho untuk memproses data dan menyimpannya ke dalam database dengan struktur yang sudah sesuai kebutuhan.

3.1.2 Dimensi Film



Gambar 3.1.2.1 ETL Dimensi Film

The screenshot shows the execution results of the ETL process for the Dimension Film. The data is presented in a table with the following columns:

	film_rental_duration	film_rental_rate	film_length	film_replacement_cost	film_rating	film_last_update	film_special_features	sk_film
1	7	4.99	117	14.99	NC-17	2013/05/26 14:50:58.951000000	Trailers	1
2	5	4.99	49	19.99	R	2013/05/26 14:50:58.951000000	"Behind the Scenes"	2
3	6	4.99	54	15.99	R	2013/05/26 14:50:58.951000000	Trailers	3
4	4	4.99	73	12.99	PG-13	2013/05/26 14:50:58.951000000	Trailers	4
5	6	0.99	86	20.99	PG	2013/05/26 14:50:58.951000000	"Deleted Scenes";"Behind the Scenes"	5
6	3	4.99	48	12.99	G	2013/05/26 14:50:58.951000000	Trailers;"Deleted Scenes"	6
7	7	2.99	50	18.99	NC-17	2013/05/26 14:50:58.951000000	Trailers;"Deleted Scenes"	7
8	5	2.99	117	26.99	G	2013/05/26 14:50:58.951000000	Commentaries;"Behind the Scenes"	8
9	6	2.99	117	26.99	C	2013/05/26 14:50:58.951000000	"Deleted Scenes"	9

Gambar 3.1.2.2 Running Pentaho

Dalam proses ini mengambil data dari tabel staging film menggunakan step input dan lakukan preview untuk memastikan data terbaca dengan benar. Bersihkan data dengan Replace in String, menghapus tanda kurung {} dari kolom special_features agar rapi. Gunakan Select Values untuk mengubah nama tabel sesuai dimensi, seperti "film_filmtitle", sambil menghapus kolom yang tidak relevan. Tambahkan surrogate key (SK) dengan add sequence untuk memberikan primary key pada dimensi film. Masukkan data ke tabel database menggunakan Insert/Update, memastikan data tersimpan sesuai struktur. Jalankan perintah SQL untuk memvalidasi struktur tabel dimensi sebelum transformasi, lalu eksekusi proses di Pentaho untuk memindahkan data ke tabel dimensi.

3.1.3 Dimensi Payment



Gambar 3.1.3.1 ETL Dimensi Payment

The screenshot shows the execution results of the ETL process for the Dimension Payment. The data is presented in a table with the following columns:

#	payment_id	customer_id	staff_id	rental_id	amount	payment_date	sk_payment
1	17503	341	2	1520	7.99	2007/02/15 22:25:46.996577000	1
2	17504	341	1	1778	1.99	2007/02/16 17:23:14.996577000	2
3	17505	341	1	1849	7.99	2007/02/16 22:41:45.996577000	3
4	17506	341	2	2829	2.99	2007/02/19 19:39:56.996577000	4
5	17507	341	2	3130	7.99	2007/02/20 17:31:48.996577000	5
6	17508	341	1	3382	5.99	2007/02/21 12:33:49.996577000	6
7	17509	342	2	2190	5.99	2007/02/17 23:58:17.996577000	7
8	17510	342	1	2914	5.99	2007/02/20 02:11:44.996577000	8
9	17511	342	1	3081	2.99	2007/02/20 13:57:39.996577000	9

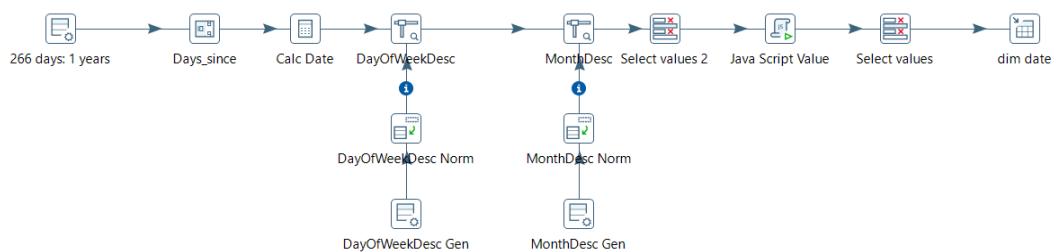
Gambar 3.1.3.2 Running Pentaho

Gunakan step input di Pentaho untuk mengambil data mentah dari tabel staging payment, yang akan diproses lebih lanjut ke dalam tabel dimensi yang sesuai. Lakukan

preview data pada step input untuk memastikan data dari staging payment berhasil diambil dan ditampilkan dengan benar, sehingga tidak ada kesalahan dalam konfigurasi koneksi, query, atau mapping kolom.

Tambahkan sk_payment sebagai surrogate key menggunakan step Add Sequence, yang berfungsi sebagai primary key unik pada tabel dimensi untuk memastikan integritas data dalam data warehouse. Gunakan step Insert/Update untuk memasukkan data yang telah diproses ke tabel dimensi payment di database, dengan konfigurasi yang sesuai dengan struktur tabel target. Sebelum menjalankan transformasi, eksekusi perintah SQL terlebih dahulu untuk memastikan tabel di database siap digunakan. Setelah itu, jalankan transformasi di Pentaho untuk memindahkan data dari staging payment ke tabel dim_payment di database. Transformasi dianggap berhasil jika seluruh data tersimpan tanpa error.

3.1.4 Dimensi Rental Date



Gambar 3.1.4.1 ETL Dimensi Rental Date

#	sk_rentaldate	tanggal	deskripsi_tanggal	sk_tahun	tahun_angka	sk_kuartal	kuartal_angka	kuartal	kuartal_tahun	sk_bulan	bulan_angka	bulan	sk_j
1	20050524	2005/05/24 00:00:00.000	24-05-2005	2005	2005	2	2 Q2	Q2	Q22005	5	5	May	
2	20050525	2005/05/25 00:00:00.000	25-05-2005	2005	2005	2	2 Q2	Q2	Q22005	5	5	May	
3	20050526	2005/05/26 00:00:00.000	26-05-2005	2005	2005	2	2 Q2	Q2	Q22005	5	5	May	
4	20050527	2005/05/27 00:00:00.000	27-05-2005	2005	2005	2	2 Q2	Q2	Q22005	5	5	May	
5	20050528	2005/05/28 00:00:00.000	28-05-2005	2005	2005	2	2 Q2	Q2	Q22005	5	5	May	
6	20050529	2005/05/29 00:00:00.000	29-05-2005	2005	2005	2	2 Q2	Q2	Q22005	5	5	May	
7	20050530	2005/05/30 00:00:00.000	30-05-2005	2005	2005	2	2 Q2	Q2	Q22005	5	5	May	
8	20050531	2005/05/31 00:00:00.000	31-05-2005	2005	2005	2	2 Q2	Q2	Q22005	5	5	May	
n	2	2 Q2	Q2	Q22005	5	5	May	

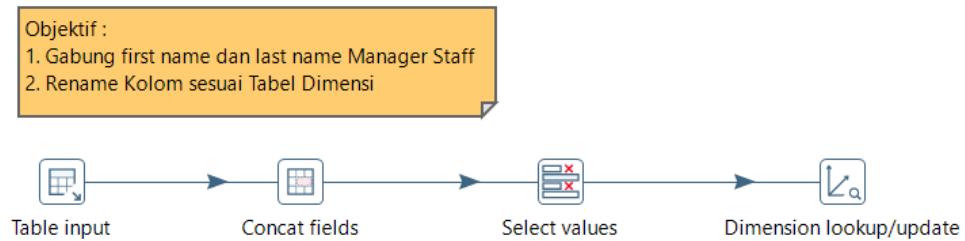
Gambar 3.1.4.2 Running Pentaho

Tahap pembuatan dimensi rental date dimulai dengan menentukan rentang waktu yang akan diproses, yaitu 266 hari atau sekitar 1 tahun, sebagai dasar untuk membangun dimensi tanggal. Perhitungan jumlah hari sejak tanggal awal dilakukan pada tahap Days_since, menghasilkan data tanggal bertahap yang kemudian dihitung ulang pada langkah Calc Date untuk membuat kolom tanggal yang akan digunakan. Selanjutnya, nama hari ditambahkan menggunakan DayOfWeekDesc, yang dilengkapi dengan proses normalisasi pada DayOfWeekDesc Norm dan pembuatan deskripsi melalui DayOfWeekDesc Gen.

Proses serupa dilakukan untuk nama bulan melalui MonthDesc, dengan normalisasi pada MonthDesc Norm dan pembuatan data deskripsi pada MonthDesc Gen. Data kemudian

disaring menggunakan Select values 2 untuk memilih kolom relevan, sementara langkah Java Script Value digunakan untuk transformasi tambahan pada data. Setelah itu, tahap Select values kembali menyaring kolom yang akan dimasukkan ke tabel dim date, menghasilkan tabel dimensi tanggal yang berisi kolom tanggal, deskripsi hari, dan deskripsi bulan. Tabel ini siap digunakan untuk kebutuhan analisis data lebih lanjut.

3.1.5 Dimensi staff



Gambar 3.1.5.1 ETL Dimensi Staff

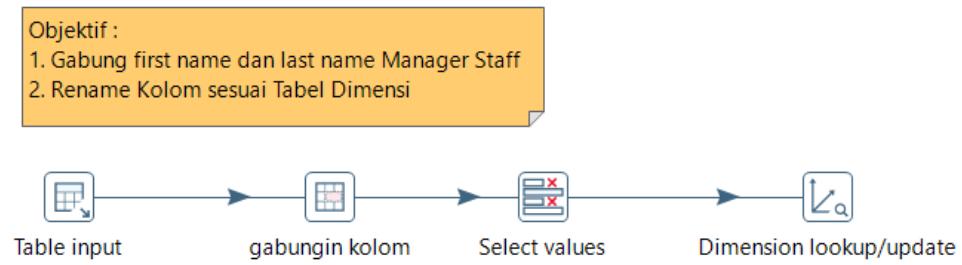
Execution Results													
Logging Execution History Step Metrics Performance Graph Metrics Preview data													
<input checked="" type="radio"/> First rows <input type="radio"/> Last rows <input type="radio"/> Off													
#	staff_id	staff_addressID	staff_storeID	staff.FullName	staff_email	staff_address.name	staff_district	staff_city	staff_country	staff_postal_code	staff_phone		
1	1	3	1	Mike Hiller	Mike.Hiller@sakilastaff.com	23 Workhaven Lane	Alberta	Lethbridge	Canada	<null>	14033335568		
2	2	4	2	Jon Stephens	Jon.Stephens@sakilastaff.com	1411 Lilydale Drive	QLD	Woodridge	Australia	<null>	6172235589		

Gambar 3.1.5.2 Running Pentaho

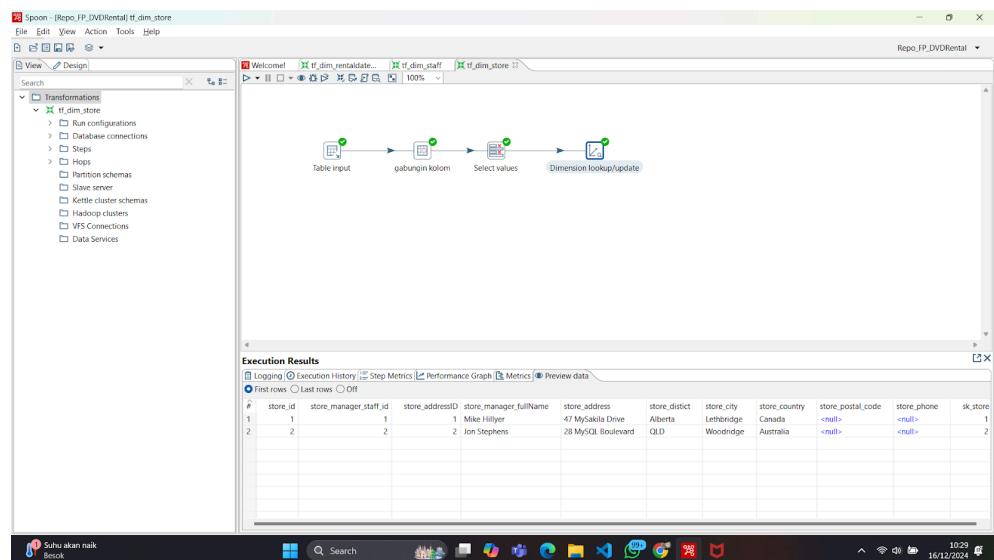
Data dari tabel staging staff diambil sebagai sumber utama untuk memproses informasi lengkap tentang karyawan. Proses JOIN dilakukan dengan tabel address, city, dan country untuk melengkapi informasi, termasuk alamat lengkap, kota, dan negara tempat karyawan tinggal. Tabel staff menyediakan data dasar seperti nama dan jabatan, sementara tabel address menambahkan detail lokasi, tabel city menyebutkan kota, dan tabel country mencantumkan negara. Kolom first_name dan last_name dari tabel staff digabung menggunakan Concat Field untuk membuat nama lengkap karyawan dalam format "FirstName LastName."

Select Values digunakan untuk mengganti nama kolom agar sesuai dengan struktur tabel dimensi staff, menghapus kolom yang tidak relevan, serta mengatur tipe dan urutan kolom sesuai kebutuhan. Pada tahap akhir, step Dimension Lookup/Update digunakan untuk memasukkan atau memperbarui data di tabel dimensi staff di database target. Sebelum transformasi dijalankan, perintah SQL dieksekusi untuk memvalidasi struktur tabel. Setelah itu, transformasi di Pentaho dijalankan untuk memindahkan data dari staging staff ke tabel dimensi staff di database, memastikan data tersimpan dengan benar tanpa error.

3.1.6 Dimensi Store



Gambar 3.1.6.1 ETL Dimensi Store

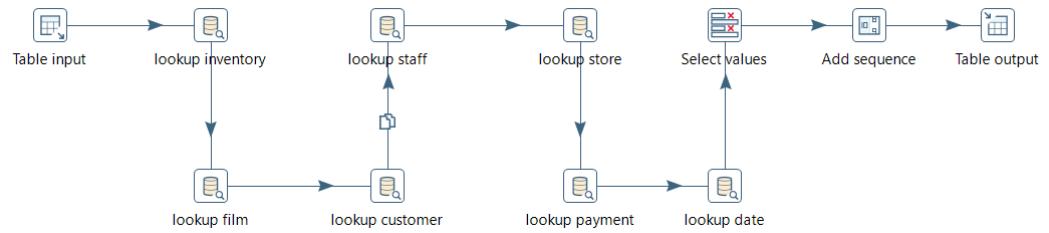


Gambar 3.1.6.2 Running Pentaho

Data dari tabel staging store diambil menggunakan step input di Pentaho sebagai sumber utama untuk dimensi store. Konfigurasi step input dilakukan dengan menambahkan query SQL untuk memilih kolom yang relevan dari tabel staging. Setelah konfigurasi selesai, data dipreview untuk memastikan bahwa pengambilan data dari staging store berhasil tanpa error. Langkah berikutnya adalah menggunakan step Concat Field untuk menggabungkan kolom manager_firstname dan manager_lastname menjadi kolom baru bernama manager_fullname, sehingga informasi nama lengkap manajer dapat disimpan dengan lebih sederhana.

Select Values digunakan untuk mengganti nama kolom agar sesuai dengan struktur tabel dimensi store, sambil menghapus kolom yang tidak diperlukan dan menyesuaikan tipe data dengan kebutuhan desain tabel. Pada tahap akhir, step Dimension Lookup/Update digunakan untuk memasukkan atau memperbarui data ke tabel dimensi store di database. Sebelum menjalankan transformasi, perintah SQL dieksekusi terlebih dahulu untuk memvalidasi tabel target. Setelah itu, transformasi di Pentaho dijalankan untuk memindahkan data dari staging store ke tabel dimensi store, memastikan data tersimpan dengan benar di database tanpa error.

3.1.7 Tabel Fakta



Gambar 3.1.7.1 ETL Fact Table

#	rental_id	sk_rentaldate	sk_inventory	sk_film	sk_customer	sk_staff	sk_store	sk_payment	amount	film_rental_duration	id_penyewaan
1	1520	20050615.0	3419.0	750.0	342	1	1	1.0	7.99	6	1
2	1778	20050616.0	2512.0	555.0	342	2	2	2.0	1.99	5	2
3	1849	20050617.0	2507.0	554.0	342	2	1	3.0	7.99	3	3
4	2829	20050619.0	2047.0	447.0	342	1	1	4.0	2.99	5	4
5	3130	20050620.0	2569.0	564.0	342	2	1	5.0	7.99	6	5
6	3382	20050621.0	131.0	30.0	342	2	1	6.0	5.99	3	6
7	2190	20050618.0	4398.0	961.0	343	2	2	7.0	5.99	7	7
8	2914	20050620.0	2419.0	529.0	343	2	2	8.0	5.99	4	8
9	3081	20050620.0	2071.0	395.0	343	2	2	9.0	2.99	4	9
10	3517	20050616.0	640.0	221.0	343	2	2	10.0	1.00	7	10

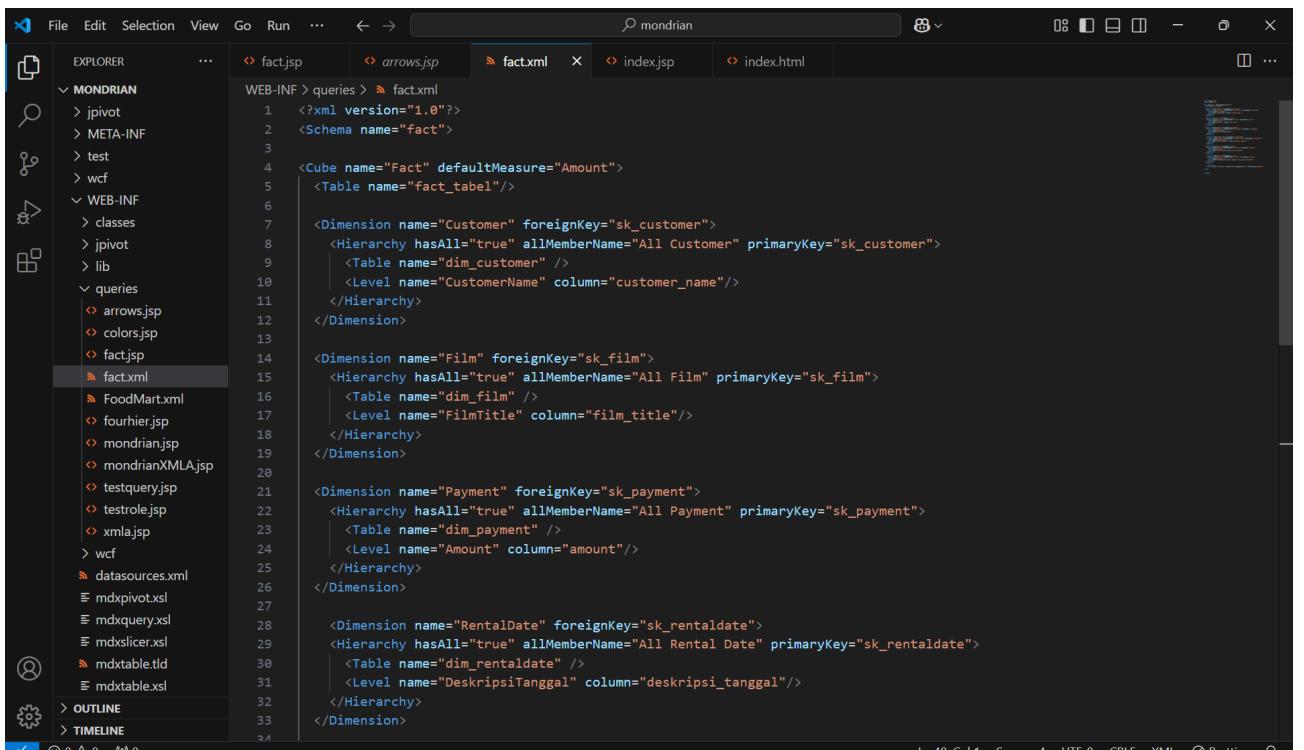
Gambar 3.1.7.2 Running Pentaho

Dalam proses ETL ini, setiap langkah DB Lookup harus dilakukan secara berurutan karena setiap langkah 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, langkah Lookup Store dan Lookup Staff bergantung pada data kunci yang diperoleh dari langkah sebelumnya. Jika urutan ini tidak diikuti, kolom kunci yang diperlukan untuk lookup selanjutnya tidak akan tersedia, yang dapat menyebabkan kegagalan join atau data yang tidak lengkap. Oleh karena itu, menjaga urutan lookup sangat penting agar aliran data berjalan dengan benar dan menghasilkan output yang akurat.

Setelah semua lookup selesai, data rental diambil dari tabel staging payment menggunakan step input di Pentaho, dan dilakukan Preview untuk memastikan data berhasil diambil. Proses lookup database dilakukan dengan Database Lookup hingga semua dimensi yang diperlukan selesai diproses. Setelah itu, kolom-kolom yang relevan dipilih untuk dimuat ke dalam tabel output, memastikan hanya data yang diperlukan yang disertakan. Pada langkah terakhir, data dimasukkan ke dalam database menggunakan step Dimension Lookup/Update. Sebelum menjalankan transformasi di Pentaho, eksekusi perintah SQL terlebih dahulu untuk memvalidasi struktur tabel di database. Dengan semua langkah ini, data dari staging payment dapat berhasil dipindahkan ke tabel dimensi dengan struktur yang sesuai.

3.2. Implementasi Cube dan OLAP

File fact.xml mendefinisikan skema untuk sebuah cube OLAP bernama "Fact" dengan fokus pada model data yang terkait dengan penjualan. Skema ini mencakup sebuah tabel fakta (fact_tabel) dan beberapa dimensi yang terhubung ke tabel terkait, seperti dim_customer, dim_film, dim_payment, dim_rentaldate, dim_staff, dan dim_store. Setiap dimensi memiliki hierarki dengan level-level tertentu, seperti CustomerName, FilmTitle, dan Amount, yang digunakan untuk analisis data. Kubus ini juga mendefinisikan dua ukuran (measure): TotalAmount, yang menghitung jumlah kolom amount, dan Total Sales, yang menghitung jumlah kolom sales_id. Struktur-struktur ini memungkinkan analisis multi-dimensional yang memudahkan pengambilan keputusan berdasarkan atribut pelanggan, film, pembayaran, tanggal penyewaan, staf, dan toko.



```
<?xml version="1.0"?>
<Schema name="fact">
  <Cube name="Fact" defaultMeasure="Amount">
    <Table name="fact_tabel"/>

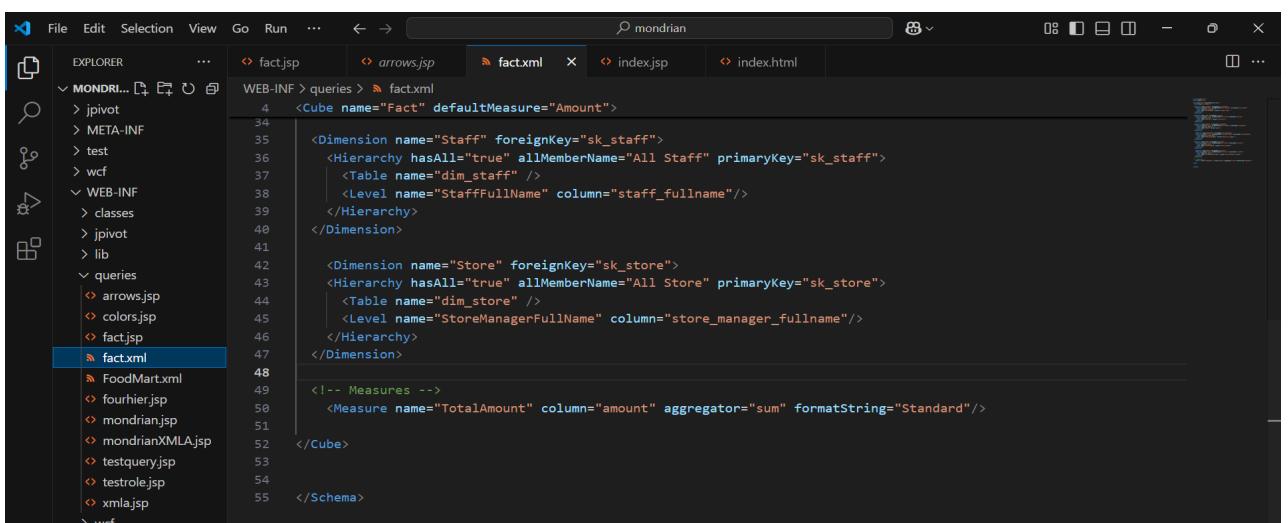
    <Dimension name="Customer" foreignKey="sk_customer">
      <Hierarchy hasAll="true" allMemberName="All Customer" primaryKey="sk_customer">
        <Table name="dim_customer" />
        <Level name="CustomerName" column="customer_name"/>
      </Hierarchy>
    </Dimension>

    <Dimension name="Film" foreignKey="sk_film">
      <Hierarchy hasAll="true" allMemberName="All Film" primaryKey="sk_film">
        <Table name="dim_film" />
        <Level name="FilmTitle" column="film_title"/>
      </Hierarchy>
    </Dimension>

    <Dimension name="Payment" foreignKey="sk_payment">
      <Hierarchy hasAll="true" allMemberName="All Payment" primaryKey="sk_payment">
        <Table name="dim_payment" />
        <Level name="Amount" column="amount"/>
      </Hierarchy>
    </Dimension>

    <Dimension name="RentalDate" foreignKey="sk_rentaldate">
      <Hierarchy hasAll="true" allMemberName="All Rental Date" primaryKey="sk_rentaldate">
        <Table name="dim_rentaldate" />
        <Level name="DeskripsiTanggal" column="deskripsi_tanggal"/>
      </Hierarchy>
    </Dimension>
  </Cube>
</Schema>
```

Gambar 3.2.1 Query Fact.XML



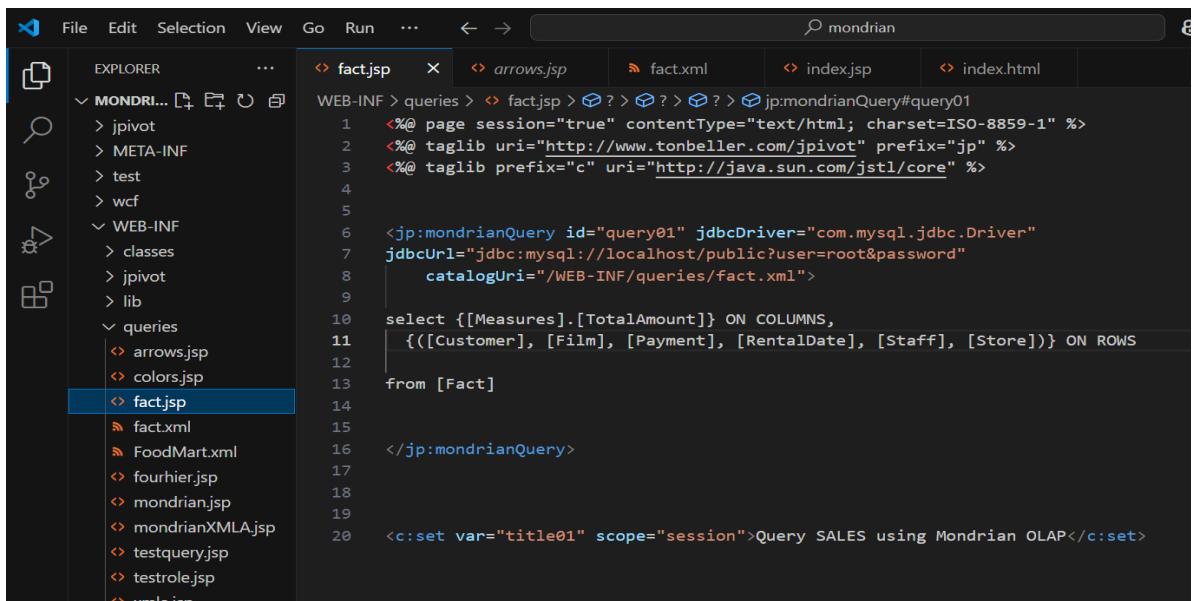
```
<?xml version="1.0"?>
<Schema name="fact">
  <Cube name="Fact" defaultMeasure="Amount">
    <Dimension name="Staff" foreignKey="sk_staff">
      <Hierarchy hasAll="true" allMemberName="All Staff" primaryKey="sk_staff">
        <Table name="dim_staff" />
        <Level name="StaffFullName" column="staff_fullname"/>
      </Hierarchy>
    </Dimension>

    <Dimension name="Store" foreignKey="sk_store">
      <Hierarchy hasAll="true" allMemberName="All Store" primaryKey="sk_store">
        <Table name="dim_store" />
        <Level name="StoreManagerFullName" column="store_manager_fullname"/>
      </Hierarchy>
    </Dimension>

    <!-- Measures -->
    <Measure name="TotalAmount" column="amount" aggregator="sum" formatString="Standard"/>
  </Cube>
</Schema>
```

Gambar 3.2.2 Query Fact.XML

Berikutnya terdapat query fact.jsp yang digunakan untuk menjalankan sebuah query OLAP menggunakan Mondrian. Dalam halaman JSP, query ini menghubungkan ke database MySQL melalui JDBC dengan menggunakan driver com.mysql.jdbc.Driver dan URL koneksi yang mengarah ke database lokal public. Query ini mengakses kubus Fact yang didefinisikan dalam file fact.xml. Dalam query, ukuran TotalAmount dipilih untuk ditampilkan di kolom, sementara dimensi Customer, Film, Payment, RentalDate, Staff, dan Store dipilih untuk ditampilkan di baris. Hasil query ini menampilkan analisis total jumlah berdasarkan kombinasi dimensi tersebut. Di akhir, sebuah atribut sesi dengan nama title01 diatur untuk menyimpan nilai "Query SALES using Mondrian OLAP".



```

<%@ page session="true" contentType="text/html; charset=ISO-8859-1" %>
<%@ taglib uri="http://www.tonbeller.com/jpivot" prefix="jp" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>

<jp:mondrianQuery id="query01" jdbcDriver="com.mysql.jdbc.Driver"
    jdbcUrl="jdbc:mysql://localhost/public?user=root&password"
    catalogUri="/WEB-INF/queries/fact.xml">

    select {[Measures].[TotalAmount]} ON COLUMNS,
    {[Customer], [Film], [Payment], [RentalDate], [Staff], [Store]} ON ROWS
    from [Fact]

</jp:mondrianQuery>

<c:set var="title01" scope="session">Query SALES using Mondrian OLAP</c:set>

```

Gambar 3.2.3 Query Fact.JSP

Berikut adalah tampilan hasil query penjualan (SALES) menggunakan Mondrian OLAP. Data diagregasi di semua dimensi (Customer, Film, Payment, RentalDate, Staff, dan Store), sehingga menampilkan total penjualan untuk keseluruhan data, yaitu 122,624. Simbol "+" pada setiap baris menunjukkan bahwa setiap dimensi dapat diekspansi untuk melihat detail data yang lebih spesifik.

Query SALES using Mondrian OLAP



						Measures
Customer	Film	Payment	RentalDate	Staff	Store	TotalAmount
+All Customer	+All Film	+All Payment	+All Rental Date	+All Staff	+All Store	122,624

Slicer:

[back to index](#)

Gambar 3.2.4 Tampilan Hasil Mondrian OLAP

3.2.1 Customer

Dimensi "Customer" dalam query Mondrian menggunakan foreign key sk_customer dan mengambil data dari tabel dim_customer dengan level "CustomerName," di mana hierarki dimensi memiliki anggota "All Customer" sebagai tingkat tertinggi. Hasil query ditampilkan dalam bentuk tabel yang menunjukkan total transaksi (TotalAmount) per pelanggan, dengan data diagregasi tanpa filter berdasarkan film, pembayaran, tanggal penyewaan, staf, atau toko. Bagian ini memberikan gambaran agregasi transaksi berdasarkan pelanggan.

```
<Dimension name="Customer" primaryKey="sk_customer">
    <Hierarchy hasAll="true" allMemberName="All Customer" primaryKey="sk_customer">
        <Table name="dim_customer" />
        <Level name="CustomerName" column="customer_name"/>
    </Hierarchy>
</Dimension>
```

Gambar 3.2.1.1 Query Customer

Query SALES using Mondrian OLAP



Customer	Film	Payment	RentalDate	Staff	Store	Measures
All Customer	All Film	All Payment	All Rental Date	All Staff	All Store	122,624
Aaron Selby	All Film	All Payment	All Rental Date	All Staff	All Store	178
Adam Gooch	All Film	All Payment	All Rental Date	All Staff	All Store	196
Adrian Clary	All Film	All Payment	All Rental Date	All Staff	All Store	138
Agnes Bishop	All Film	All Payment	All Rental Date	All Staff	All Store	174
Alan Kahn	All Film	All Payment	All Rental Date	All Staff	All Store	240
Albert Crouse	All Film	All Payment	All Rental Date	All Staff	All Store	194
Alberto Henning	All Film	All Payment	All Rental Date	All Staff	All Store	134
Alex Gresham	All Film	All Payment	All Rental Date	All Staff	All Store	287
Alexander Fennell	All Film	All Payment	All Rental Date	All Staff	All Store	275
Alfred Casillas	All Film	All Payment	All Rental Date	All Staff	All Store	241
Alfredo Mcadams	All Film	All Payment	All Rental Date	All Staff	All Store	172
Alice Stewart	All Film	All Payment	All Rental Date	All Staff	All Store	247
Alicia Mills	All Film	All Payment	All Rental Date	All Staff	All Store	148
Allan Cornish	All Film	All Payment	All Rental Date	All Staff	All Store	128
Allen Butterfield	All Film	All Payment	All Rental Date	All Staff	All Store	150
Allison Stanley	All Film	All Payment	All Rental Date	All Staff	All Store	174
Alma Austin	All Film	All Payment	All Rental Date	All Staff	All Store	257
Alvin Deloach	All Film	All Payment	All Rental Date	All Staff	All Store	269
Amanda Carter	All Film	All Payment	All Rental Date	All Staff	All Store	211
Amber Dixon	All Film	All Payment	All Rental Date	All Staff	All Store	227
Amy Lopez	All Film	All Payment	All Rental Date	All Staff	All Store	225
Ana Bradley	All Film	All Payment	All Rental Date	All Staff	All Store	335
Andre Rapp	All Film	All Payment	All Rental Date	All Staff	All Store	221
Andrea Henderson	All Film	All Payment	All Rental Date	All Staff	All Store	186
Andrew Purdy	All Film	All Payment	All Rental Date	All Staff	All Store	180

Gambar 3.2.1.2 Tampilan Customer

3.2.2 Film

Dimensi "Film" menggunakan foreign key sk_film, mengambil data dari tabel dim_film, dan menggunakan kolom film_title untuk level "FilmTitle". Tampilan data menunjukkan ringkasan penjualan, di mana kolom-kolom seperti Customer, Payment, RentalDate, Staff, dan Store menampilkan "All", menandakan data diagregasi berdasarkan film tanpa filter pada dimensi lain. Kolom TotalAmount menunjukkan total transaksi untuk setiap film, dan simbol "+" menunjukkan opsi untuk drill-down dan melihat detail lebih

lanjut berdasarkan dimensi lain. Bagian ini menunjukkan total penjualan yang dikelompokkan berdasarkan film dan memberikan opsi untuk analisis lebih lanjut.

```
<Dimension name="Film" primaryKey="sk_film">
  <Hierarchy hasAll="true" allMemberName="All Film" primaryKey="sk_film">
    <Table name="dim_film" />
    <Level name="FilmTitle" column="film_title"/>
  </Hierarchy>
</Dimension>
```

Gambar 3.2.2.1 Query Film

Query SALES using Mondrian OLAP



Customer	Film	Payment	RentalDate	Staff	Store	Measures
		All Payment	All Rental Date	All Staff	All Store	TotalAmount
All Customer	All Film	All Payment	All Rental Date	All Staff	All Store	122,624
	Academy Dinosaur	All Payment	All Rental Date	All Staff	All Store	68
	Ace Goldfinger	All Payment	All Rental Date	All Staff	All Store	106
	Adaptation Holes	All Payment	All Rental Date	All Staff	All Store	70
	Affair Prejudice	All Payment	All Rental Date	All Staff	All Store	168
	African Egg	All Payment	All Rental Date	All Staff	All Store	96
	Agent Truman	All Payment	All Rental Date	All Staff	All Store	224
	Airplane Sierra	All Payment	All Rental Date	All Staff	All Store	166
	Airport Pollock	All Payment	All Rental Date	All Staff	All Store	174
	Alabama Devil	All Payment	All Rental Date	All Staff	All Store	144
	Aladdin Calendar	All Payment	All Rental Date	All Staff	All Store	264
	Alamo videotape	All Payment	All Rental Date	All Staff	All Store	62
	Alaska Phantom	All Payment	All Rental Date	All Staff	All Store	76
	Ali Forever	All Payment	All Rental Date	All Staff	All Store	110
	Alice Fantasia	All Payment	All Rental Date	All Staff	All Store	
	Alien Center	All Payment	All Rental Date	All Staff	All Store	170
	Alley Evolution	All Payment	All Rental Date	All Staff	All Store	94
	Alone Trip	All Payment	All Rental Date	All Staff	All Store	124
	Alter Victory	All Payment	All Rental Date	All Staff	All Store	62
	Amadeus Holy	All Payment	All Rental Date	All Staff	All Store	66
	Amelie Hellfighters	All Payment	All Rental Date	All Staff	All Store	136
	American Circus	All Payment	All Rental Date	All Staff	All Store	294
	Amistad Midsummer	All Payment	All Rental Date	All Staff	All Store	142
	Anaconda Confessions	All Payment	All Rental Date	All Staff	All Store	104
	Analyze Hoosiers	All Payment	All Rental Date	All Staff	All Store	112
	Angels Life	All Payment	All Rental Date	All Staff	All Store	208

Gambar 3.2.2.2 Tampilan Film

3.2.3 Payment

Dimensi "Payment" menggunakan foreign key sk_payment, mengambil data dari tabel dim_payment, dan menggunakan kolom amount untuk level "Amount". Tampilan data menunjukkan ringkasan penjualan yang dikelompokkan berdasarkan jumlah pembayaran, dengan kolom lain seperti Customer, Film, RentalDate, Staff, dan Store menampilkan "All", yang berarti data diagregasi berdasarkan metode pembayaran tanpa filter pada dimensi lain. Kolom TotalAmount menunjukkan total transaksi untuk setiap jumlah pembayaran. Bagian ini memberikan gambaran total penjualan yang dikelompokkan berdasarkan jumlah pembayaran dan menyediakan opsi untuk analisis lebih lanjut.

```
<Dimension name="Payment" primaryKey="sk_payment">
  <Hierarchy hasAll="true" allMemberName="All Payment" primaryKey="sk_payment">
    <Table name="dim_payment" />
    <Level name="Amount" column="amount"/>
  </Hierarchy>
</Dimension>
```

Gambar 3.2.3.1 Query Payment

Query SALES using Mondrian OLAP

Customer	Film	Payment	RentalDate	Staff	Store	Measures
						TotalAmount
*All Customer	All Film	All Payment	All Rental Date	All Staff	All Store	122,624
		0	All Rental Date	All Staff	All Store	0
		0.99	All Rental Date	All Staff	All Store	5,386
		1.98	All Rental Date	All Staff	All Store	4
		1.99	All Rental Date	All Staff	All Store	2,308
		2.99	All Rental Date	All Staff	All Store	19,333
		3.98	All Rental Date	All Staff	All Store	64
		3.99	All Rental Date	All Staff	All Store	7,884
		4.99	All Rental Date	All Staff	All Store	34,172
		5.98	All Rental Date	All Staff	All Store	84
		5.99	All Rental Date	All Staff	All Store	14,232
		6.99	All Rental Date	All Staff	All Store	14,218
		7.98	All Rental Date	All Staff	All Store	80
		7.99	All Rental Date	All Staff	All Store	9,924
		8.97	All Rental Date	All Staff	All Store	18
		8.99	All Rental Date	All Staff	All Store	7,875
		9.98	All Rental Date	All Staff	All Store	20
		9.99	All Rental Date	All Staff	All Store	4,655
		10.99	All Rental Date	All Staff	All Store	2,176
		11.99	All Rental Date	All Staff	All Store	192

Slicer:

[back to index](#)

Gambar 3.2.3.2 Tampilan Payment

3.2.4 Rental Date

Definisi dimensi "Rental Date" menggunakan foreign key sk_rentaldate, mengambil data dari tabel dim_rentaldata, dan menggunakan kolom deskripsi_tanggal untuk level "DeskripsiTanggal". Tampilan data menunjukkan ringkasan penjualan yang dikelompokkan berdasarkan tanggal penyewaan, dengan kolom-kolom seperti Customer, Film, Payment, Staff, dan Store menampilkan "All", menandakan data diagregasi berdasarkan tanggal penyewaan tanpa filter pada dimensi lain. Kolom TotalAmount menunjukkan total transaksi untuk setiap tanggal penyewaan. Bagian ini menyajikan gambaran total penjualan yang dikelompokkan berdasarkan tanggal penyewaan.

```

<Dimension name="RentalDate" primaryKey="sk_rentaldate">
  <Hierarchy hasAll="true" allMemberName="All Rental Date" primaryKey="sk_rentaldate">
    <Table name="dim_rentaldate" />
    <Level name="DeskripsiTanggal" column="deskripsi_tanggal"/>
  </Hierarchy>
</Dimension>

```

Gambar 3.2.4.1 Query Rental Date

Query SALES using Mondrian OLAP



						Measures
Customer	Film	Payment	RentalDate	Staff	Store	TotalAmount
+All Customer	+All Film	+All Payment	+All Rental Date	+All Staff	+All Store	122,624
			01-01-2006	+All Staff	+All Store	
			01-02-2006	+All Staff	+All Store	
			01-06-2005	+All Staff	+All Store	
			01-07-2005	+All Staff	+All Store	
			01-08-2005	+All Staff	+All Store	5,635
			01-09-2005	+All Staff	+All Store	
			01-10-2005	+All Staff	+All Store	
			01-11-2005	+All Staff	+All Store	
			01-12-2005	+All Staff	+All Store	
			02-01-2006	+All Staff	+All Store	
			02-02-2006	+All Staff	+All Store	
			02-06-2005	+All Staff	+All Store	
			02-07-2005	+All Staff	+All Store	
			02-08-2005	+All Staff	+All Store	5,453
			02-09-2005	+All Staff	+All Store	
			02-10-2005	+All Staff	+All Store	
			02-11-2005	+All Staff	+All Store	
			02-12-2005	+All Staff	+All Store	
			03-01-2006	+All Staff	+All Store	
			03-02-2005	+All Staff	+All Store	
			03-06-2005	+All Staff	+All Store	
			03-07-2005	+All Staff	+All Store	
			03-08-2005	+All Staff	+All Store	
			03-09-2005	+All Staff	+All Store	
			03-10-2005	+All Staff	+All Store	

Gambar 3.2.4.2 Tampilan Rental Date

3.2.5 Staff

Dimensi "Staff" menggunakan foreign key sk_staff, mengambil data dari tabel dim_staff, dan menggunakan kolom staff_fullname untuk level "StaffFullName" (Nama Lengkap Staf). Tampilan data menunjukkan ringkasan penjualan yang dikelompokkan berdasarkan staf, dengan kolom-kolom seperti Customer, Film, Payment, RentalDate, dan Store menampilkan "All", menandakan data diagregasi berdasarkan staf tanpa filter pada dimensi lain. Kolom TotalAmount menunjukkan total transaksi untuk setiap staf. Terlihat total penjualan untuk semua staf adalah 122,624, yang kemudian dipecah menjadi penjualan oleh Jon Stephens (61,627) dan Mike Hillyer (60,997).

```
<Dimension name="Staff" primaryKey="sk_staff">
    <Hierarchy hasAll="true" allMemberName="All Staff" primaryKey="sk_staff">
        <Table name="dim_staff" />
        <Level name="StaffFullName" column="staff_fullname"/>
    </Hierarchy>
</Dimension>
```

Gambar 3.2.5.1 Query Staff

Query SALES using Mondrian OLAP



						Measures
Customer	Film	Payment	RentalDate	Staff	Store	TotalAmount
+All Customer	+All Film	+All Payment	+All Rental Date	-All Staff	+All Store	122,624
				Jon Stephens	+All Store	61,627
				Mike Hillyer	+All Store	60,997

Slicer:

[back to index](#)

Gambar 3.2.5.2 Tampilan Staff

3.2.6 Store

Dimensi "Store" menggunakan foreign key sk_store, mengambil data dari tabel dim_store, dan menggunakan kolom store_manager_fullname (Nama Lengkap Manajer Toko) untuk level "StoreManagerFullName". Tampilan data menunjukkan ringkasan penjualan, yang tampaknya difokuskan pada penjualan yang dilakukan oleh dua staf (Jon Stephens dan Mike Hillyer), dan dikelompokkan berdasarkan toko. Karena kolom "Store" dalam tabel hasil menunjukkan "All Store" untuk kedua staf, ini mengindikasikan bahwa data penjualan mereka diagregasi di semua toko tempat mereka bekerja. Kolom TotalAmount menunjukkan total transaksi untuk setiap staf di semua toko. Total penjualan untuk semua staf di semua toko adalah 122,624, dengan Jon Stephens berkontribusi 53,258 dan Mike Hillyer berkontribusi 61,268.

```
<Dimension name="Store" primaryKey="sk_store">
  <Hierarchy hasAll="true" allMemberName="All Store" primaryKey="sk_store">
    <Table name="dim_store" />
    <Level name="StoreManagerFullName" column="store_manager_fullname"/>
  </Hierarchy>
</Dimension>
```

Gambar 3.2.6.1 Query Store

Query SALES using Mondrian OLAP



						Measures
Customer	Film	Payment	RentalDate	Staff	Store	TotalAmount
+All Customer	+All Film	+All Payment	+All Rental Date	+All Staff	-All Store	122,624
					Jon Stephens	61,366
					Mike Hillyer	61,258

Slicer:

[back to index](#)

Gambar 3.2.6.2 Tampilan Store

LAMPIRAN

Link Github:

https://github.com/rakhanaufalllll/KonversiDWO_MuhammadRakhaNaufal_060