

**LAPORAN KONVERSI DATA WAREHOUSE & OLAP
“PENGEMBANGAN DATA WAREHOUSE DALAM ANALISIS
PENYEWAAN DAN PERFORMA FILM BERDASARKAN GENRE,
AKTOR DAN TREN PENYEWAAN”
SEMESTER GANJIL 2024/2025**



Dosen Pengampu :

Mohammad Irwan Afandi, S.T, M.Sc.

Disusun Oleh :

Marcellio Aurel Christian - 22082010019

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR
2024**

DAFTAR ISI

DAFTAR ISI.....	1
BAB 1 PENDAHULUAN.....	2
1.1 Deskripsi.....	2
1.2 Rumusan Masalah.....	2
1.3 Batasan.....	2
BAB 2 METODOLOGI.....	3
2.1 Metode Pengembangan.....	3
2.1.1 Perencanaan.....	3
2.1.2 Perancangan Skema.....	3
2.1.3 Ekstraksi Data.....	4
2.1.4 Pembuatan CUBE Mondrian di Tomcat.....	5
BAB 3 HASIL DAN IMPLEMENTASI.....	6
3.1 Implementasi Model Multidimensi.....	6
3.1.1 Skema DWH.....	6
3.2 Implementasi Cube dan OLAP.....	13
3.2.1 Cube dan OLAP.....	13
LAMPIRAN.....	19

BAB 1

PENDAHULUAN

1.1 Deskripsi

Dalam industri hiburan, khususnya penyewaan film dapat memahami tentang preferensi pelanggan dan performa produk yang sangat penting dalam peningkatan pendapatan dan kepuasan pelanggan. Genre film, popularitas aktor, dan tren penyewaan merupakan faktor utama yang mempengaruhi keputusan pelanggan dalam menyewa film. Namun, banyak perusahaan menghadapi tantangan dalam menganalisis data ini secara efektif karena kurangnya infrastruktur data yang memadai dan alat analisis yang mendukung pengambilan keputusan strategis.

Proyek ini berfokus pada analisis penyewaan dan performa film berdasarkan genre, aktor, dan tren penyewaan. Maka dari itu digunakan **DVD Rental Database**, sebuah database yang mencakup data transaksi penyewaan, informasi pelanggan, dan detail produk (film). Database ini relevan karena menyediakan informasi yang lengkap untuk mengidentifikasi pola penyewaan dan mengukur performa film secara kuantitatif.

1.2 Rumusan Masalah

Bagaimana pengembangan data warehouse yang dapat digunakan untuk mendukung analisis data penyewaan film berdasarkan genre, aktor, dan tren penyewaan untuk mengidentifikasi preferensi pelanggan? Selain itu, bagaimana data tersebut dapat diolah untuk memberikan wawasan mendalam mengenai preferensi pelanggan?

1.3 Batasan

Pada proyek ini, pengembangan data warehouse dalam analisis penyewaan dan performa film berdasarkan genre, aktor, dan tren penyewaan hanya mencakup data yang berasal dari database DVD Rental. Pembahasan dan analisis dibatasi pada dua measures yaitu, total transaksi (Jumlah keseluruhan transaksi penyewaan film), serta total pendapatan (Jumlah total pendapatan yang dihasilkan dari transaksi tersebut).

BAB 2

METODOLOGI

2.1 Metode Pengembangan

2.1.1 Perencanaan

Proses perencanaan dalam pengembangan data warehouse untuk analisis penyewaan dan performa film berdasarkan genre, aktor, dan tren penyewaan dimulai dengan analisis kebutuhan data dan identifikasi struktur database. Database yang digunakan adalah database DVD Rental, yang terdiri dari berbagai tabel utama seperti rental, payment, customer, store, staff, date, inventory, dan film. Setiap tabel memiliki peran spesifik dalam mencatat transaksi penyewaan, data pelanggan, hingga informasi detail film.

Pada tahap ini, hubungan antar-tabel dipetakan untuk memastikan data yang diperlukan dapat diintegrasikan dengan baik, terutama untuk menghasilkan informasi yang relevan terkait total transaksi dan total pendapatan. Pemahaman menyeluruh terhadap struktur dan sumber data ini menjadi dasar utama dalam merancang skema data warehouse yang efektif dan sesuai dengan tujuan analisis.

Selain itu, dirancang pula strategi untuk proses Extract, Transform, Load (ETL) guna memastikan data yang ada dapat diproses secara optimal. Proses ETL ini bertujuan untuk mengintegrasikan data dari berbagai tabel, membersihkan dan mengubah format data jika diperlukan, serta memuatnya ke dalam data warehouse, sehingga menghasilkan informasi yang terstruktur dan mendukung analisis penyewaan serta performa film secara komprehensif.

2.1.2 Perancangan Skema

Data warehouse yang dirancang untuk mendukung analisis penyewaan dan performa film berdasarkan genre, aktor, dan tren penyewaan menggunakan model *star schema*. Model ini dipilih karena memiliki struktur yang sederhana, mudah dipahami, dan lebih efisien dalam implementasi dibandingkan dengan model lainnya. *Star schema* memungkinkan pengelolaan data yang terorganisir dengan baik melalui pemisahan antara tabel fakta dan tabel dimensi.

Tabel fakta yang dirancang adalah fact_rental, yang berisi informasi utama terkait transaksi, seperti tanggal sewa, jumlah transaksi, dan total pendapatan. Data ini menjadi pusat dari analisis dan dihubungkan dengan berbagai tabel dimensi yang relevan.

Adapun tabel dimensi yang dirancang adalah sebagai berikut:

1. Dim_Film: Berisi informasi detail mengenai genre, judul, dan rating film. Dimensi ini menggunakan *Slowly Changing Dimension (SCD) Type 2* untuk mencatat perubahan historis data film.
2. Dim_Aktor: Menyimpan daftar aktor yang berperan dalam setiap film. Dimensi ini juga menggunakan *SCD Type 2* untuk melacak perubahan data aktor dari waktu ke waktu.
3. Dim_Waktu: Menyediakan data kalender yang mencakup tanggal, bulan, dan tahun. Dimensi ini dirancang dengan *SCD Type 1* karena data waktu bersifat tetap dan tidak memerlukan pelacakan historis.
4. Dim_Customer: Berisi informasi demografis pelanggan, seperti nama, alamat, dan preferensi. Dimensi ini menggunakan *SCD Type 2* untuk mencatat perubahan informasi pelanggan seiring waktu.
5. Dim_Staff: Berisi data staf yang terlibat dalam setiap transaksi penyewaan. Dimensi ini menggunakan *SCD Type 2* untuk melacak perubahan data staf.
6. Dim_Category: Menyimpan informasi kategori film yang tersedia. Dimensi ini menggunakan *SCD Type 1* karena kategori bersifat tetap dan jarang berubah.
7. Dim_Store: Berisi informasi detail mengenai toko penyewaan film, seperti lokasi dan identitas toko. Dimensi ini menggunakan *SCD Type 2* untuk mencatat perubahan historis data toko.

2.1.3 Ekstraksi Data

Pada proyek ini, proses ekstraksi data dilakukan dengan mengimpor database DVD Rental ke PostgreSQL sebagai langkah awal dalam pengelolaan data mentah. Proses ini mencakup pengambilan data dari tabel-tabel utama, seperti film, customer, rental, payment, dan store, dengan menggunakan *query* PostgreSQL untuk memperoleh subset data yang relevan dengan kebutuhan analisis.

Data yang diekstrak kemudian divalidasi untuk memastikan integritasnya, termasuk menghindari duplikasi data dan nilai *null* pada key utama. Setelah validasi, data dipindahkan ke staging area menggunakan Pentaho Data Integration (PDI). Di staging area, dilakukan proses pembersihan dan transformasi data untuk memastikan struktur data sesuai dengan kebutuhan analisis yang melibatkan total transaksi, total pendapatan, dan dimensi seperti genre, aktor, serta tren penyewaan.

Data yang telah diproses di staging area menjadi dasar integrasi ke dalam data warehouse. Proses ini memastikan bahwa data yang dihasilkan siap digunakan untuk analisis penyewaan dan performa film secara komprehensif, mendukung pengambilan wawasan yang relevan sesuai dengan tujuan proyek ini.

2.1.4 Pembuatan CUBE Mondrian di Tomcat

Pembuatan *cube* Mondrian dilakukan untuk mendukung analisis multidimensional pada data warehouse dalam proyek analisis penyewaan dan performa film berdasarkan genre, aktor, dan tren penyewaan. Proses ini dimulai dengan mendefinisikan file skema XML yang menghubungkan tabel fakta dan tabel dimensi dalam *star schema*. Skema ini dirancang untuk memungkinkan analisis data secara mendalam dari berbagai sudut pandang, seperti berdasarkan genre, aktor, waktu, toko, atau pelanggan.

Setelah skema *cube* selesai dibuat, implementasi dilakukan pada server Tomcat. Server Tomcat bertindak sebagai penghubung antara data warehouse dan tools OLAP, seperti Pentaho. Dengan integrasi ini, pengguna dapat mengakses dan menganalisis data secara fleksibel melalui antarmuka OLAP yang mendukung eksplorasi data secara interaktif.

Dengan adanya *cube* Mondrian, analisis data menjadi lebih efisien dan mendalam, memungkinkan pengguna untuk menggali informasi berdasarkan kebutuhan, seperti pola penyewaan berdasarkan genre tertentu atau kontribusi aktor terhadap total pendapatan. Hal ini memberikan fleksibilitas lebih dalam menjelajahi data serta mendukung pengambilan keputusan berbasis wawasan yang dihasilkan dari data warehouse.

BAB 3

HASIL DAN IMPLEMENTASI

3.1 Implementasi Model Multidimensi

3.1.1 Skema DWH

Desain data warehouse ini menggunakan model star schema yang terdiri dari tabel fakta dan tabel dimensi:

1. dim_actor

A. Create Table SQL

```
CREATE TABLE dwh.dim_actor (  
    sk_actor bigserial NOT NULL,  
    actor_id int2 NULL,  
    actor_name text NULL,  
    film_id int2 NULL,  
    "version" int4 NULL,  
    date_from timestamp NULL,  
    date_to timestamp NULL,  
    CONSTRAINT dim_actor_pkey PRIMARY KEY (sk_actor)  
);  
CREATE INDEX idx_dim_actor_lookup ON dwh.dim_actor USING btree  
(actor_id);  
CREATE INDEX idx_dim_actor_tk ON dwh.dim_actor USING btree (sk_actor);
```

B. Dimension Table

Column Name	#	Data type	Identity	Collation	Not Null	Default
123 sk_actor	1	bigserial			[v]	nextval('dwh.dim_actor_sk_actor_seq'::regclass)
123 actor_id	2	int2			[]	
A-Z actor_name	3	text		default	[]	
123 film_id	4	int2			[]	
123 version	5	int4			[]	
🕒 date_from	6	timestamp			[]	
🕒 date_to	7	timestamp			[]	

2. dim_category

A. Create Table SQL

```
CREATE TABLE dwh.dim_category (  
    sk_category int4 NOT NULL,  
    category_id int2 NULL,  
    category_name varchar(25) NULL,  
    film_id int2 NULL,  
    CONSTRAINT dim_category_pkey PRIMARY KEY (sk_category)  
);
```

```
CREATE INDEX idx_dim_category_lookup ON dwh.dim_category USING btree
(sk_category);
```

B. Dimension Table

Column Name	#	Data type	Identity	Collation	Not N
123 sk_category	1	int4			[v]
123 category_id	2	int2			[]
A-Z category_name	3	varchar(25)		default	[]
123 film_id	4	int2			[]

3. dim_customer

A. Create Table SQL

```
CREATE TABLE dwh.dim_customer (
    sk_customer bigserial NOT NULL,
    customer_id int4 NULL,
    customer_store_id int2 NULL,
    customer_name text NULL,
    customer_email varchar(50) NULL,
    customer_active text NULL,
    customer_address_id int2 NULL,
    customer_address varchar(50) NULL,
    customer_district varchar(20) NULL,
    customer_city_id int2 NULL,
    customer_postal_code varchar(10) NULL,
    customer_phone varchar(20) NULL,
    customer_city varchar(50) NULL,
    customer_country_id int2 NULL,
    customer_country varchar(50) NULL,
    "version" int4 NULL,
    date_from timestamp NULL,
    date_to timestamp NULL,
    CONSTRAINT dim_customer_pkey PRIMARY KEY (sk_customer)
);
CREATE INDEX idx_dim_customer_lookup ON dwh.dim_customer USING
btree (customer_id);
CREATE INDEX idx_dim_customer_tk ON dwh.dim_customer USING btree
(sk_customer);
```


B. Dimension Table

Column Name	#	Data type	Identity	Collation	Not Null	Default
I23 sk_customer	1	bigserial			[v]	nextval('dwh.dim_customer_sk_customer_seq':regclass)
I23 customer_id	2	int4			[]	
I23 customer_store_id	3	int2			[]	
A-Z customer_name	4	text		default	[]	
A-Z customer_email	5	varchar(50)		default	[]	
A-Z customer_active	6	text		default	[]	
I23 customer_address_id	7	int2			[]	
A-Z customer_address	8	varchar(50)		default	[]	
A-Z customer_district	9	varchar(20)		default	[]	
I23 customer_city_id	10	int2			[]	
A-Z customer_postal_code	11	varchar(10)		default	[]	
A-Z customer_phone	12	varchar(20)		default	[]	
A-Z customer_city	13	varchar(50)		default	[]	
I23 customer_country_id	14	int2			[]	
A-Z customer_country	15	varchar(50)		default	[]	
I23 version	16	int4			[]	
date_from	17	timestamp			[]	
date_to	18	timestamp			[]	

4. dim_date

A. Create Table

```
CREATE TABLE dwh.dim_date (
    sk_waktu int4 NOT NULL,
    tanggal timestamp NULL,
    deskripsi_tanggal text NULL,
    sk_tahun float8 NULL,
    tahun_angka int2 NULL,
    sk_kuartal float8 NULL,
    kuartal_angka float8 NULL,
    kuartal varchar(2) NULL,
    kuartal_tahun varchar(32) NULL,
    sk_bulan float8 NULL,
    bulan_angka int2 NULL,
    bulan varchar(30) NULL,
    sk_minggu float8 NULL,
    minggu varchar(32) NULL,
    minggu_dalam_tahun_angka int2 NULL,
    minggu_dalam_bulan_angka float8 NULL,
    minggu_dalam_bulan text NULL,
    hari varchar(30) NULL,
    hari_dalam_tahun_angka int2 NULL,
    hari_dalam_bulan_angka int2 NULL,
    hari_dalam_minggu_angka int2 NULL,
    is_weekend varchar(1) NULL,
    banyak_hari_dalam_bulan int2 NULL,
    tahun_sort varchar(4) NULL,
    hari_dalam_minggu_sort varchar(60) NULL,
    CONSTRAINT dim_date_pkey PRIMARY KEY (sk_waktu)
```

);

B. Dimension Table

Column Name	#	Data type	Identity	Collation	Not Null	Default
I23 sk_customer	1	bigserial			[v]	nextval('dwh.dim_customer_sk_customer_seq'::regclass)
I23 customer_id	2	int4			[]	
I23 customer_store_id	3	int2			[]	
A-Z customer_name	4	text		default	[]	
A-Z customer_email	5	varchar(50)		default	[]	
A-Z customer_active	6	text		default	[]	
I23 customer_address_id	7	int2			[]	
A-Z customer_address	8	varchar(50)		default	[]	
A-Z customer_district	9	varchar(20)		default	[]	
I23 customer_city_id	10	int2			[]	
A-Z customer_postal_code	11	varchar(10)		default	[]	
A-Z customer_phone	12	varchar(20)		default	[]	
A-Z customer_city	13	varchar(50)		default	[]	
I23 customer_country_id	14	int2			[]	
A-Z customer_country	15	varchar(50)		default	[]	
I23 version	16	int4			[]	
date_from	17	timestamp			[]	
date_to	18	timestamp			[]	

5. dim_film

A. Create Table

```
CREATE TABLE dwh.dim_film (
    sk_film bigserial NOT NULL,
    film_id int4 NULL,
    film_title varchar(255) NULL,
    film_description text NULL,
    film_year int4 NULL,
    film_language_id int2 NULL,
    film_language_name varchar(20) NULL,
    film_rental_duration int2 NULL,
    film_rental_rate numeric(6, 2) NULL,
    film_length int2 NULL,
    film_replacement_cost numeric(7, 2) NULL,
    film_rating text NULL,
    film_special_features text NULL,
    "version" int4 NULL,
    date_from timestamp NULL,
    date_to timestamp NULL,
    CONSTRAINT dim_film_pkey PRIMARY KEY (sk_film)
);

CREATE INDEX idx_dim_film_lookup ON dwh.dim_film USING btree
(film_id);

CREATE INDEX idx_dim_film_tk ON dwh.dim_film USING btree (sk_film);
```

B. Dimension Table

Column Name	#	Data type	Identity	Collation	Not Null	Default
123 sk_film	1	bigserial			[v]	nextval('dwh.dim_film_sk_film_seq'::regclass)
123 film_id	2	int4			[]	
A-Z film_title	3	varchar(255)		default	[]	
A-Z film_description	4	text		default	[]	
123 film_year	5	int4			[]	
123 film_language_id	6	int2			[]	
A-Z film_language_name	7	varchar(20)		default	[]	
123 film_rental_duration	8	int2			[]	
123 film_rental_rate	9	numeric(6, 2)			[]	
123 film_length	10	int2			[]	
123 film_replacement_cost	11	numeric(7, 2)			[]	
A-Z film_rating	12	text		default	[]	
A-Z film_special_features	13	text		default	[]	
123 version	14	int4			[]	
🕒 date_from	15	timestamp			[]	
🕒 date_to	16	timestamp			[]	

6. dim_staff

A. Create Table SQL

```
CREATE TABLE dwh.dim_staff (
    sk_staff bigserial NOT NULL,
    staff_id int4 NULL,
    staff_name text NULL,
    staff_email varchar(50) NULL,
    staff_address_id int2 NULL,
    staff_address varchar(255) NULL,
    staff_district varchar(50) NULL,
    staff_city_id int2 NULL,
    staff_postal_code varchar(20) NULL,
    staff_phone varchar(20) NULL,
    staff_city varchar(50) NULL,
    staff_country_id int2 NULL,
    staff_country varchar(50) NULL,
    staff_active text NULL,
    staff_store_id int2 NULL,
    staff_username varchar(40) NULL,
    staff_password varchar(40) NULL,
    "version" int4 NULL,
    date_from timestamp NULL,
    date_to timestamp NULL,
    CONSTRAINT dim_staff_pkey PRIMARY KEY (sk_staff)
);
CREATE INDEX idx_dim_staff_lookup ON dwh.dim_staff USING btree
(staff_id);
CREATE INDEX idx_dim_staff_tk ON dwh.dim_staff USING btree (sk_staff);
```

B. Dimension Table

Column Name	#	Data type	Identity	Collation	Not Null	Default
123 sk_staff	1	bigserial			[v]	nextval('dwh.dim_staff_sk_staff_seq'::regclass)
123 staff_id	2	int4			[]	
A-Z staff_name	3	text		default	[]	
A-Z staff_email	4	varchar(50)		default	[]	
123 staff_address_id	5	int2			[]	
A-Z staff_address	6	varchar(255)		default	[]	
A-Z staff_district	7	varchar(50)		default	[]	
123 staff_city_id	8	int2			[]	
A-Z staff_postal_code	9	varchar(20)		default	[]	
A-Z staff_phone	10	varchar(20)		default	[]	
A-Z staff_city	11	varchar(50)		default	[]	
123 staff_country_id	12	int2			[]	
A-Z staff_country	13	varchar(50)		default	[]	
A-Z staff_active	14	text		default	[]	
123 staff_store_id	15	int2			[]	
A-Z staff_username	16	varchar(40)		default	[]	
A-Z staff_password	17	varchar(40)		default	[]	
123 version	18	int4			[]	
🕒 date_from	19	timestamp			[]	
🕒 date_to	20	timestamp			[]	

7. dim_store

A. Create Table SQL

```

CREATE TABLE dwh.dim_store (
    sk_store bigserial NOT NULL,
    store_id int4 NULL,
    store_staff_id int2 NULL,
    store_staff_name text NULL,
    store_staff_email varchar(50) NULL,
    store_address_id int2 NULL,
    store_address varchar(255) NULL,
    store_district varchar(50) NULL,
    store_city_id int2 NULL,
    store_city varchar(50) NULL,
    store_postal_code varchar(10) NULL,
    store_phone varchar(20) NULL,
    store_country_id int2 NULL,
    store_country varchar(50) NULL,
    "version" int4 NULL,
    date_from timestamp NULL,
    date_to timestamp NULL,
    CONSTRAINT dim_store_pkey PRIMARY KEY (sk_store)
);
CREATE INDEX idx_dim_store_lookup ON dwh.dim_store USING btree
(store_id);
CREATE INDEX idx_dim_store_tk ON dwh.dim_store USING btree (sk_store);

```

B. Dimension Table

Column Name	#	Data type	Identity	Collation	Not Null	Default
123 sk_store	1	bigserial			[v]	nextval('dwh.dim_store_sk_store_seq::regclass')
123 store_id	2	int4			[]	
123 store_staff_id	3	int2			[]	
AZ store_staff_name	4	text		default	[]	
AZ store_staff_email	5	varchar(50)		default	[]	
123 store_address_id	6	int2			[]	
AZ store_address	7	varchar(255)		default	[]	
AZ store_district	8	varchar(50)		default	[]	
123 store_city_id	9	int2			[]	
AZ store_city	10	varchar(50)		default	[]	
AZ store_posta_code	11	varchar(10)		default	[]	
AZ store_phone	12	varchar(20)		default	[]	
123 store_country_id	13	int2			[]	
AZ store_country	14	varchar(50)		default	[]	
123 version	15	int4			[]	
🕒 date_from	16	timestamp			[]	
🕒 date_to	17	timestamp			[]	

8. fact_rental

A. Create Table SQL

```
CREATE TABLE dwh.fact_rental (
    trx_id float8 NULL,
    sk_customer int8 NULL,
    sk_staff int8 NULL,
    sk_film int8 NULL,
    sk_category int4 NULL,
    sk_actor int8 NULL,
    sk_store int8 NULL,
    sk_rental_date int4 NULL,
    sk_return_date int4 NULL,
    sk_payment_date int4 NULL,
    amount numeric(7, 2) NULL
);
```

B. Dimension Table

Column Name	#	Data type	Identity	Collation	Not Null
123 trx_id	1	float8			[]
123 sk_customer	2	int8			[]
123 sk_staff	3	int8			[]
123 sk_film	4	int8			[]
123 sk_category	5	int4			[]
123 sk_actor	6	int8			[]
123 sk_store	7	int8			[]
123 sk_rental_date	8	int4			[]
123 sk_return_date	9	int4			[]
123 sk_payment_date	10	int4			[]
123 amount	11	numeric(7, 2)			[]

3.2 Implementasi Cube dan OLAP

3.2.1 Cube dan OLAP

a. Cube rental.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Schema name="New Schema1">
  <Dimension type="StandardDimension" visible="true" highCardinality="false" name="Customer">...
</Dimension>
  <Dimension type="TimeDimension" visible="true" highCardinality="false" name="Date">...
</Dimension>
  <Dimension type="StandardDimension" visible="true" highCardinality="false" name="Category">...
</Dimension>
  <Dimension type="StandardDimension" visible="true" highCardinality="false" name="Film">...
</Dimension>
  <Dimension type="StandardDimension" visible="true" name="Actor">...
</Dimension>
  <Cube name="Rental" visible="true" cache="true" enabled="true">...
</Cube>
</Schema>

```

b. Cube dw_phiquery.jsp

```
dw_phiquery.jsp > ?
1 <%@ page session="true" contentType="text/html; charset=ISO-8859-1" %>
2 <%@ taglib uri="http://www.tonbeller.com/jpivot" prefix="jp" %>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
4
5 <jp:mondrianQuery
6   id="query01"
7   jdbcDriver="org.postgresql.Driver"
8   jdbcUrl="jdbc:postgresql://localhost:5432/dwh_dvdrental?user=postgres&password=J0y0b0y0"
9   catalogUri="/WEB-INF/queries/rental.xml">
10
11   SELECT
12     {[Measures].[Total Transaksi], [Measures].[Total Pendapatan]} ON COLUMNS,
13     {[Customer].[Semua Customer],[Date].[Semua Waktu], [Category].[Semua Genre], [Film].[Semua Film], [Actor].[Semua Actor]} ON ROWS
14 FROM [Rental]
15 </jp:mondrianQuery>
16
17 <c:set var="title01" scope="session">PHI-Minimart - Cube Pendapatan</c:set>
18

```

c. Customer

- Query

```
<Dimension type="StandardDimension" visible="true" highCardinality="false"
name="Customer">
```

```
  <Hierarchy name="Customer" visible="true" hasAll="true"
allMemberName="Semua Customer" primaryKey="sk_customer">
```

```
    <Table name="dim_customer" schema="dwh">
```

```
    </Table>
```

```
    <Level name="Customer" visible="true" column="customer_id"
nameColumn="customer_name" ordinalColumn="customer_id" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="IfBlankName">
    </Level>
  </Hierarchy>
</Dimension>
```

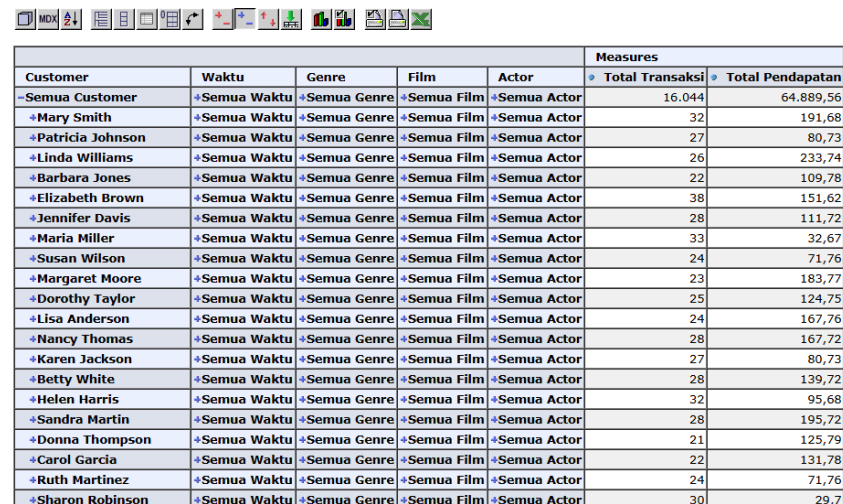
```

<Level name="City" visible="true" column="customer_city_id"
nameColumn="customer_city" ordinalColumn="customer_city_id" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="IfBlankName">
</Level>
<Level name="Country" visible="true" column="customer_country_id"
nameColumn="customer_country" ordinalColumn="customer_country_id"
type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="IfBlankName">
</Level>
</Hierarchy>
</Dimension>

```

- Hasil

PHI-Minimart - Cube Pendapatan



					Measures	
Customer	Waktu	Genre	Film	Actor	Total Transaksi	Total Pendapatan
Semua Customer	Semua Waktu	Semua Genre	Semua Film	Semua Actor	16,044	64,889,56
Mary Smith	Semua Waktu	Semua Genre	Semua Film	Semua Actor	32	191,68
Patricia Johnson	Semua Waktu	Semua Genre	Semua Film	Semua Actor	27	80,73
Linda Williams	Semua Waktu	Semua Genre	Semua Film	Semua Actor	26	233,74
Barbara Jones	Semua Waktu	Semua Genre	Semua Film	Semua Actor	22	109,78
Elizabeth Brown	Semua Waktu	Semua Genre	Semua Film	Semua Actor	38	151,62
Jennifer Davis	Semua Waktu	Semua Genre	Semua Film	Semua Actor	28	111,72
Maria Miller	Semua Waktu	Semua Genre	Semua Film	Semua Actor	33	32,67
Susan Wilson	Semua Waktu	Semua Genre	Semua Film	Semua Actor	24	71,76
Margaret Moore	Semua Waktu	Semua Genre	Semua Film	Semua Actor	23	183,77
Dorothy Taylor	Semua Waktu	Semua Genre	Semua Film	Semua Actor	25	124,75
Lisa Anderson	Semua Waktu	Semua Genre	Semua Film	Semua Actor	24	167,76
Nancy Thomas	Semua Waktu	Semua Genre	Semua Film	Semua Actor	28	167,72
Karen Jackson	Semua Waktu	Semua Genre	Semua Film	Semua Actor	27	80,73
Betty White	Semua Waktu	Semua Genre	Semua Film	Semua Actor	28	139,72
Helen Harris	Semua Waktu	Semua Genre	Semua Film	Semua Actor	32	95,68
Sandra Martin	Semua Waktu	Semua Genre	Semua Film	Semua Actor	28	195,72
Donna Thompson	Semua Waktu	Semua Genre	Semua Film	Semua Actor	21	125,79
Carol Garcia	Semua Waktu	Semua Genre	Semua Film	Semua Actor	22	131,78
Ruth Martinez	Semua Waktu	Semua Genre	Semua Film	Semua Actor	24	71,76
Sharon Robinson	Semua Waktu	Semua Genre	Semua Film	Semua Actor	30	29,7

d. Waktu

- Query

```

<Dimension type="TimeDimension" visible="true" highCardinality="false"
name="Date">
<Hierarchy name="Waktu" visible="true" hasAll="true"
allMemberName="Semua Waktu" primaryKey="sk_waktu">
<Table name="dim_date" schema="dwh">
</Table>
<Level name="Tahun" visible="true" column="tahun_angka"
type="Numeric" uniqueMembers="false" levelType="TimeYears"
hideMemberIf="IfBlankName">
</Level>

```

```

    <Level name="Bulan" visible="true" column="bulan" type="String"
uniqueMembers="false" levelType="TimeMonths"
hideMemberIf="IfBlankName">
    </Level>
</Hierarchy>
</Dimension>

```

- Hasil

PHI-Minimart - Cube Pendapatan



					Measures	
Customer	Waktu	Genre	Film	Actor	Total Transaksi	Total Pendapatan
+Semua Customer	+Semua Waktu	+Semua Genre	+Semua Film	+Semua Actor	16.044	64.889,56
	+2000	+Semua Genre	+Semua Film	+Semua Actor		
	+2001	+Semua Genre	+Semua Film	+Semua Actor		
	+2002	+Semua Genre	+Semua Film	+Semua Actor		
	+2003	+Semua Genre	+Semua Film	+Semua Actor		
	+2004	+Semua Genre	+Semua Film	+Semua Actor		
	+2005	+Semua Genre	+Semua Film	+Semua Actor	15.862	64.142,38
	+2006	+Semua Genre	+Semua Film	+Semua Actor	182	747,18
	+2007	+Semua Genre	+Semua Film	+Semua Actor		
	+2008	+Semua Genre	+Semua Film	+Semua Actor		
	+2009	+Semua Genre	+Semua Film	+Semua Actor		
	+2010	+Semua Genre	+Semua Film	+Semua Actor		
	+2011	+Semua Genre	+Semua Film	+Semua Actor		
	+2012	+Semua Genre	+Semua Film	+Semua Actor		
	+2013	+Semua Genre	+Semua Film	+Semua Actor		
	+2014	+Semua Genre	+Semua Film	+Semua Actor		
	+2015	+Semua Genre	+Semua Film	+Semua Actor		
	+2016	+Semua Genre	+Semua Film	+Semua Actor		
	+2017	+Semua Genre	+Semua Film	+Semua Actor		
	+2018	+Semua Genre	+Semua Film	+Semua Actor		
	+2019	+Semua Genre	+Semua Film	+Semua Actor		

e. Genre

- Query

```

<Dimension type="StandardDimension" visible="true" highCardinality="false"
name="Category">
    <Hierarchy name="Genre" visible="true" hasAll="true"
allMemberName="Semua Genre" primaryKey="sk_category">
        <Table name="dim_category" schema="dwh">
            </Table>
        <Level name="Category" visible="true" column="category_id"
nameColumn="category_name" ordinalColumn="category_id" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="IfBlankName">
            </Level>
        </Hierarchy>
    </Dimension>

```


- Hasil

PHI-Minimart - Cube Pendapatan



					Measures	
Customer	Waktu	Genre	Film	Actor	Total Transaksi	Total Pendapatan
+Semua Customer	+Semua Waktu	-Semua Genre	+Semua Film	+Semua Actor	16.044	64.889,56
		Action	+Semua Film	+Semua Actor	1.112	4.617,88
		Animation	+Semua Film	+Semua Actor	1.166	4.735,34
		Children	+Semua Film	+Semua Actor	945	3.827,55
		Classics	+Semua Film	+Semua Actor	939	3.759,61
		Comedy	+Semua Film	+Semua Actor	941	3.904,59
		Documentary	+Semua Film	+Semua Actor	1.050	4.150,5
		Drama	+Semua Film	+Semua Actor	1.060	4.271,4
		Family	+Semua Film	+Semua Actor	1.096	4.411,04
		Foreign	+Semua Film	+Semua Actor	1.033	4.063,67
		Games	+Semua Film	+Semua Actor	969	3.935,31
		Horror	+Semua Film	+Semua Actor	846	3.476,54
		Music	+Semua Film	+Semua Actor	830	3.379,7
		New	+Semua Film	+Semua Actor	940	3.758,6
		Sci-Fi	+Semua Film	+Semua Actor	1.101	4.410,99
		Sports	+Semua Film	+Semua Actor	1.179	4.829,21
		Travel	+Semua Film	+Semua Actor	837	3.357,63

f. Film

- Query

```

<Dimension type="StandardDimension" visible="true" highCardinality="false"
name="Film">
  <Hierarchy name="Film" visible="true" hasAll="true"
allMemberName="Semua Film" primaryKey="sk_film">
    <Table name="dim_film" schema="dwh">
      </Table>
    <Level name="Film" visible="true" column="film_id"
nameColumn="film_title" ordinalColumn="film_id" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="IfBlankName">
      </Level>
    <Level name="Tahun Film" visible="true" column="film_id"
nameColumn="film_year" ordinalColumn="film_id" type="Numeric"
uniqueMembers="false" levelType="Regular" hideMemberIf="IfBlankName">
      </Level>
    </Hierarchy>
  </Dimension>

```

- Hasil

PHI-Minimart - Cube Pendapatan



Customer	Waktu	Genre	Film	Actor	Measures	
					Total Transaksi	Total Pendapatan
+Semua Customer	+Semua Waktu	+Semua Genre	+Semua Film	+Semua Actor	16.044	64.889,56
			+Academy Dinosaur	+Semua Actor	23	85,77
			+Ace Goldfinger	+Semua Actor	7	24,93
			+Adaptation Holes	+Semua Actor	12	64,88
			+Affair Prejudice	+Semua Actor	23	92,77
			+African Egg	+Semua Actor	12	61,88
			+Agent Truman	+Semua Actor	21	69,79
			+Airplane Sierra	+Semua Actor	15	58,85
			+Airport Pollock	+Semua Actor	18	81,82
			+Alabama Devil	+Semua Actor	12	52,88
			+Aladdin Calendar	+Semua Actor	23	103,77
			+Alamo Videotape	+Semua Actor	24	109,76
			+Alaska Phantom	+Semua Actor	26	116,74
			+Ali Forever	+Semua Actor	9	38,91
			+Alice Fantasia	+Semua Actor		
			+Alien Center	+Semua Actor	22	75,78
			+Alley Evolution	+Semua Actor	14	59,86
			+Alone Trip	+Semua Actor	18	70,82
			+Alter Victory	+Semua Actor	22	99,78
			+Amadeus Holy	+Semua Actor	21	87,79
			+Amelia Hellfighters	+Semua Actor	10	37,9

g. Actor

- Query

```
<Dimension type="StandardDimension" visible="true" highCardinality="false"
name="Film">
```

```
<Hierarchy name="Film" visible="true" hasAll="true"
allMemberName="Semua Film" primaryKey="sk_film">
```

```
<Table name="dim_film" schema="dwh">
```

```
</Table>
```

```
<Level name="Film" visible="true" column="film_id"
nameColumn="film_title" ordinalColumn="film_id" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="IfBlankName">
</Level>
```

```
<Level name="Tahun Film" visible="true" column="film_id"
nameColumn="film_year" ordinalColumn="film_id" type="Numeric"
uniqueMembers="false" levelType="Regular" hideMemberIf="IfBlankName">
</Level>
```

```
</Hierarchy>
```

```
</Dimension>
```

```
<Dimension type="StandardDimension" visible="true" name="Actor">
```

```
<Hierarchy name="Actor" visible="true" hasAll="true"
allMemberName="Semua Actor" primaryKey="sk_actor">
```

```
<Table name="dim_actor" schema="dwh" alias="">
```

```
</Table>
```

```

    <Level name="Actor" visible="true" column="actor_id"
nameColumn="actor_name" ordinalColumn="actor_id" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="IfBlankName">
    </Level>
</Hierarchy>
</Dimension>

```

- Hasil

PHI-Minimart - Cube Pendapatan



					Measures	
Customer	Waktu	Genre	Film	Actor	Total Transaksi	Total Pendapatan
+Semua Customer	+Semua Waktu	+Semua Genre	+Semua Film	-Semua Actor	16.044	64.889,56
				Penelope Guinness	305	1.222,95
				Nick Wahlberg	387	1.553,13
				Ed Chase	311	1.269,89
				Jennifer Davis	211	857,89
				Johnny Lollobrigida	455	1.784,45
				Bette Nicholson	252	1.074,48
				Grace Mostel	407	1.654,93
				Matthew Johansson	254	1.053,46
				Joe Swank	315	1.297,85
				Christian Gable	243	979,57
				Zero Cage	338	1.415,62
				Karl Berry	444	1.806,56
				Uma Wood	392	1.533,08
				Vivien Bergen	434	1.787,66
				Cuba Olivier	338	1.321,62
				Fred Costner	227	955,73
				Helen Voight	327	1.380,73
				Dan Torn	194	806,06
				Bob Fawcett	250	934,5
				Lucille Tracy	248	983,52

LAMPIRAN

Link Github:

https://github.com/aceldut/Konversi-DWO_22082010019_Marcellio-Aurel-Christian