

**PROYEK AKHIR PERANCANGAN DESIGN DAN IMPLEMENTASI  
SAMPEL DATA WAREHOUSE PENJUALAN MUSIK  
MENGUNAKAN “CHINOOK DATABASE” SEMESTER GANJIL  
2024/2025**

**Dosen Pengampu :**  
Mohamad Irwan Afandi. ST.,MSC



Disusun Oleh:  
**Muhammad Ananda Giovanny Ramadhan**  
(22082010110)

**PRODI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UPN “VETERAN” JAWA TIMUR  
2024**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>3</b>
<b>BAB I PENDAHULUAN.....</b>	<b>4</b>
1.1 Latar Belakang.....	5
1.2 Rumusan Masalah .....	6
1.3 Batasan Penelitian .....	6
<b>BAB II DASAR TEORI.....</b>	<b>8</b>
2.1 Metode Pengembangan .....	8
<b>BAB III HASIL DAN PEMBAHASAN .....</b>	<b>11</b>
3.1 Tahapan Project.....	11
3.2 Perancangan Data Warehouse.....	16
3.3 Datamart Dan Datamart Olap .....	17
3.4 Pembuatan Staging Area.....	18
3.5 Implementasi Model Multi Dimensi .....	22
3.6 Implementasi Mondrian OLAP Cube .....	26
<b>LAMPIRAN .....</b>	<b>29</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Di era digital yang semakin maju, data telah menjadi aset strategis yang penting bagi perusahaan. Pengelolaan data yang baik memungkinkan bisnis untuk memperoleh wawasan yang mendalam, mendukung pengambilan keputusan yang lebih tepat, serta meningkatkan efisiensi operasional. Dalam industri yang berfokus pada transaksi dan distribusi konten digital, kemampuan untuk mengelola dan menganalisis data dengan sistematis menjadi krusial. Salah satu alat pembelajaran yang sering digunakan dalam memahami pengelolaan data adalah Chinook Database, sebuah database yang dirancang untuk meniru lingkungan bisnis berbasis penjualan musik digital seperti iTunes atau Google Play. Dengan data yang mencakup informasi tentang artis, album, lagu, pelanggan, dan transaksi, Chinook Database menawarkan simulasi realistis yang mencerminkan kompleksitas sistem data relasional di dunia nyata.

Proyek ini bertujuan untuk mengembangkan data warehouse menggunakan Chinook Database sebagai sumber utama. Data warehouse, sebagai sistem penyimpanan data yang terintegrasi dan fokus pada analisis, memungkinkan perusahaan untuk mengidentifikasi pola, tren, serta wawasan strategis yang mendukung pengambilan keputusan berbasis data. Proses pengembangan data warehouse ini melibatkan beberapa tahap, mulai dari eksplorasi dan pemahaman struktur database, pengolahan data melalui metode ETL (Extract, Transform, Load), hingga pembuatan tabel fakta dan dimensi yang mendukung analisis mendalam. Implementasi dilakukan dengan memanfaatkan alat seperti Pentaho Data Integration untuk pengolahan data dan PostgreSQL sebagai sistem manajemen basis data. Selain itu, datamart dan OLAP (Online Analytical Processing) digunakan untuk mengorganisasi data secara multidimensi, memungkinkan analisis yang efisien.

Sebagai hasil akhir, proyek ini juga mencakup visualisasi data dalam bentuk dashboard interaktif yang dibuat menggunakan Tableau. Dashboard ini menyajikan berbagai metrik kunci, seperti pendapatan berdasarkan negara, tren penjualan, dan analisis genre musik, yang dirancang untuk mendukung pengambilan keputusan strategis. Proyek ini tidak hanya melatih keterampilan teknis seperti pengelolaan database, desain data warehouse, dan visualisasi data, tetapi juga kemampuan non-teknis seperti kerja sama tim, dokumentasi, dan penyajian hasil analisis. Dengan menggunakan teknologi terkini, proyek ini memberikan pengalaman langsung yang relevan dengan kebutuhan industri saat ini, mempersiapkan peserta untuk menghadapi tantangan pengelolaan data skala besar di era transformasi digital.

## 1.2 Rumusan Masalah

Dalam proyek pengembangan data warehouse ini, terdapat beberapa masalah utama yang perlu diselesaikan. Pertama, bagaimana cara mengolah data mentah dari Chinook Database menjadi informasi yang terstruktur dan bermanfaat untuk mendukung analisis bisnis. Selain itu, diperlukan perancangan dan implementasi data warehouse yang efisien untuk menyimpan dan mengelola data secara terintegrasi. Proses ETL (Extract, Transform, Load) juga menjadi tantangan penting, yaitu bagaimana memastikan data dari sumber operasional dapat diproses, ditransformasi, dan dimuat ke data warehouse dengan baik. Selanjutnya, diperlukan pengembangan datamart dan OLAP (Online Analytical Processing) yang mampu mendukung analisis multidimensi dengan efisien. Tantangan lainnya adalah bagaimana menyajikan hasil analisis data dalam bentuk visualisasi interaktif melalui dashboard yang dapat membantu pengambilan keputusan strategis. Terakhir, bagaimana mengidentifikasi dan mengatasi kendala yang muncul selama proses integrasi data untuk memastikan keberhasilan implementasi data warehouse ini.

## 1.3 Batasan Penelitian

Dalam pengembangan data warehouse untuk Chinook Database, penelitian ini dibatasi pada aspek-aspek berikut:

### 1. Cakupan Data :

- Data yang digunakan digunakan berasal dari Chinook Database dalam format bawaan (CSV dan SQL) yang mencakup informasi pelanggan, faktur, lagu, genre, dan transaksi.
- Penelitian ini hanya menggunakan data yang tersedia di database Chinook tanpa menambahkan data eksternal atau real-time.

### 2. Teknologi Yang Digunakan :

- Proses pengambilan, transformasi, dan pemuatan data dilakukan menggunakan **Pentaho Data Integration (PDI)** sebagai alat ETL utama.
- Database yang digunakan untuk menyimpan data warehouse adalah **PostgreSQL**, sementara datamart dan OLAP dibuat menggunakan alat yang sama.
- Visualisasi data dilakukan menggunakan **Tableau**, tanpa integrasi dengan alat visualisasi lain yang lebih kompleks.

3. Fokus Analisis :

- Analisis data difokuskan pada transaksi penjualan musik, perilaku pelanggan, serta distribusi pendapatan berdasarkan negara, genre musik, dan waktu.
- Visualisasi data dirancang untuk mendukung analisis strategis, seperti pola penjualan, segmen pelanggan, dan pendapatan berdasarkan genre musik

4. Lingkup Implementasi :

- Penelitian ini mencakup tahap desain, implementasi, dan analisis data warehouse hingga visualisasi menggunakan dashboard interaktif.
- Evaluasi sistem dibatasi pada pengujian fungsionalitas sistem secara lokal dan tidak mencakup penerapan langsung di lingkungan produksi nyata.

5. Konteks Destinasi :

- Penelitian ini hanya berfokus pada data penjualan musik yang ada di Chinook Database dan tidak mencakup data dari sektor bisnis atau industri lainnya.

## BAB II

### METODE PENELITIAN

#### 2.1 Metode Pengembangan

##### 2.1.1 Perencanaan

Penelitian ini direncanakan melalui beberapa tahapan untuk memastikan pengembangan data warehouse yang efektif dalam mendukung analisis penjualan musik menggunakan *Chinook Database*. Tahap pertama adalah persiapan, yang meliputi identifikasi kebutuhan analisis data bisnis musik, pengumpulan literatur terkait data warehouse dan teknik ETL, serta eksplorasi struktur dan hubungan tabel pada *Chinook Database*.

Tahap kedua adalah perancangan sistem, yang mencakup desain arsitektur pipeline data mulai dari proses ETL hingga penyimpanan di data warehouse. Teknologi yang dipilih termasuk PostgreSQL untuk database OLTP, Pentaho Data Integration (PDI) untuk proses ETL, dan Tableau untuk visualisasi data. Skema database dirancang dengan struktur tabel fakta dan dimensi yang relevan, seperti tabel pelanggan, faktur, lagu, dan tanggal.

Tahap ketiga adalah pembuatan data dummy sebagai simulasi transaksi penjualan musik digital. Data ini digunakan untuk memvalidasi proses ETL dan memastikan integritas data selama migrasi ke data warehouse.

##### 2.1.2 Skema Perancangan

Proyek data warehouse ini menggunakan arsitektur pipeline data untuk mengintegrasikan data dari *Chinook Database*. Data dari *Chinook Database* yang berfungsi sebagai sistem OLTP diekstrak menggunakan DBeaver. Pada tahap ini, proses ETL (Extract, Transform, Load) dilakukan dengan memanfaatkan Pentaho Data Integration (PDI). Data yang diambil diekstrak dari tabel-tabel utama seperti *Customer*, *Invoice*, dan *Track*. Kemudian, data tersebut ditransformasikan dengan membersihkan nilai kosong atau duplikat, menormalkan struktur data, dan menggabungkan tabel terkait sesuai skema yang dirancang. Selanjutnya, data yang telah diproses dimuat ke PostgreSQL sebagai data warehouse untuk mendukung analisis lebih lanjut.

Data warehouse dirancang menggunakan model skema bintang (*star schema*) yang terdiri dari satu tabel fakta dan beberapa tabel dimensi. Tabel fakta utama adalah tabel *Faktur Penjualan* yang menyimpan data transaksi seperti *InvoiceID*, total penjualan (*Total*), serta referensi ke tabel dimensi. Tabel dimensi meliputi dimensi *Pelanggan* yang mencatat data pelanggan seperti *CustomerID*, nama, email, dan lokasi; dimensi *Lagu* yang mencatat detail lagu seperti *TrackID*, judul, genre, komposer, dan album; serta dimensi *Waktu* yang mencakup informasi tanggal, bulan, tahun, dan kuartal.

Teknologi yang digunakan dalam perancangan ini mencakup PostgreSQL untuk penyimpanan data warehouse, Pentaho Data Integration untuk proses ETL, dan Tableau untuk visualisasi hasil analisis. Dengan struktur ini, data warehouse dapat digunakan untuk menganalisis pola penjualan, perilaku pelanggan, dan performa lagu, sehingga mendukung pengambilan keputusan bisnis yang berbasis data.

### 2.1.3 Ekstraksi Data

Proses ekstraksi data dimulai dengan mengambil data mentah dari *Chinook Database*, yang berfungsi sebagai sistem OLTP. Data diekstrak menggunakan DBeaver, sebuah alat yang memungkinkan eksekusi query SQL untuk mengambil informasi dari tabel-tabel yang relevan. Tabel utama yang diekstrak meliputi *Customer*, *Invoice*, dan *Track*, yang masing-masing berisi data pelanggan, transaksi, dan lagu. Pada tahap ini, data diekstrak dalam format yang sesuai untuk proses selanjutnya tanpa mengubah struktur atau isi data aslinya. Ekstraksi dilakukan secara sistematis dengan mempertimbangkan integritas data, memastikan bahwa semua data yang diambil bebas dari error dan sesuai dengan kebutuhan analisis.

Hasil ekstraksi kemudian disimpan sementara di staging area untuk proses transformasi lebih lanjut. Staging area memungkinkan pemisahan antara data mentah yang diekstraksi dan data yang telah diproses, sehingga memastikan setiap langkah dalam pipeline ETL dapat diaudit dan dimonitor dengan baik. Proses ini bertujuan untuk menyediakan data awal yang konsisten dan terstruktur sebagai dasar pembangunan data warehouse.

### 2.1.4 Pembuatan CUBE Mondrian Di Tomcat

Pembuatan CUBE Mondrian di Tomcat dimulai dengan mempersiapkan data warehouse yang telah dirancang menggunakan PostgreSQL, di mana tabel fakta dan tabel dimensi yang berbasis skema bintang (*star schema*) menjadi dasar untuk analisis data multidimensi. Selanjutnya, file konfigurasi *schema.xml* dibuat untuk mendefinisikan struktur CUBE, termasuk fakta utama, dimensi, hierarki, dan ukuran (*measures*). Tabel fakta *Faktur Penjualan* digunakan untuk menyimpan informasi transaksi, sementara dimensi seperti Pelanggan, Lagu, dan Waktu dirancang untuk mendukung analisis dari berbagai sudut pandang.

File konfigurasi *schema.xml* ini kemudian ditempatkan di direktori *webapps* server Tomcat, di dalam folder *tomcat/webapps/mondrian/WEB-INF/*. Selain itu, file *web.xml* diperbarui untuk memastikan Mondrian dapat berjalan dan terhubung dengan data warehouse di PostgreSQL. Setelah konfigurasi selesai, pengujian dilakukan dengan menggunakan query MDX (Multidimensional Expressions) melalui alat seperti Pentaho BI Server untuk memastikan struktur CUBE berfungsi dengan baik dan data dapat diakses sesuai kebutuhan.

CUBE Mondrian yang telah berhasil dibuat memungkinkan eksplorasi data multidimensi melalui *dashboard* atau laporan interaktif. Pengguna dapat melakukan analisis mendalam, seperti *drill-down* data pelanggan, mengevaluasi tren penjualan berdasarkan waktu, atau memeriksa kontribusi pendapatan per genre musik. Dengan CUBE Mondrian, data warehouse dapat dioptimalkan untuk mendukung pengambilan keputusan yang berbasis data secara efisien.

### 2.1.5 Perancangan Dashboard

Perancangan dashboard dilakukan dengan tujuan untuk menyediakan visualisasi data yang interaktif, estetik, dan efisien. Dashboard ini dirancang agar mudah digunakan oleh pengguna dengan fokus pada informasi utama yang relevan untuk mendukung analisis dan pengambilan keputusan berbasis data. Proses perancangan dimulai dengan identifikasi kebutuhan pengguna, yang melibatkan analisis data dari *data mart* dan penentuan metrik penting seperti pendapatan, volume penjualan, dan segmentasi pelanggan. Selanjutnya, dibuat *mockup* dashboard untuk merancang tata letak visual dan elemen-elemen utama, seperti peta, grafik garis, *treemap*, dan tabel. Desain mengadopsi tema futuristik dengan palet warna minimalis (#000000, #f1eee9, #59504e) dan penggunaan font profesional (*Aetrina*, *Apple Garamond*, dan *Poppins Medium*) dengan ukuran yang bervariasi sesuai elemen.

Dashboard yang dihasilkan terdiri dari dua bagian utama, yaitu dashboard utama dan dashboard tambahan. Dashboard utama menampilkan peta distribusi pendapatan per negara, *treemap* untuk pendapatan berdasarkan genre musik, serta statistik penting seperti total pendapatan, transaksi, dan penjualan. Selain itu, terdapat grafik segmentasi pelanggan berdasarkan negara dan *donut chart* untuk pendapatan berdasarkan genre. Dashboard tambahan memberikan detail lebih lanjut, seperti tren pelanggan dan transaksi melalui grafik garis, analisis harga unit bulanan dengan *highlight table*, serta grafik penjualan tertinggi berdasarkan pelanggan. Kedua dashboard dilengkapi dengan navigasi interaktif, termasuk tombol untuk mengunduh data dalam format PDF, mengakses profil LinkedIn tim, dan melihat video presentasi proyek di YouTube.

Pengembangan dashboard menggunakan Tableau untuk visualisasi, Pentaho Data Integration untuk memproses data mentah, dan PostgreSQL untuk manajemen database. Dashboard ini dirancang untuk memberikan gambaran makro yang jelas sekaligus memungkinkan eksplorasi mendalam terhadap data, sehingga dapat mendukung kebutuhan bisnis dan pengambilan keputusan strategis.



## BAB III

### HASIL DAN Pengerjaan

#### 3.1 Tahapan Project

##### 3.1.1 Metodologi Pengumpulan Data

*Chinook Database* menggunakan data dari berbagai sumber untuk membuat lingkungan simulasi bisnis yang realistis. Data ini mencakup informasi tentang pelanggan, produk, transaksi, dan manajemen katalog musik. Data tentang pelanggan dan transaksi dibuat secara manual dengan menggunakan nama, alamat, dan informasi kontak fiktif dari sumber terbuka seperti *Google Maps*. Sementara itu, data media dan katalog musik diambil dari data asli yang menunjukkan struktur bisnis toko musik digital.

Database ini terdiri dari beberapa tabel utama, termasuk “*Album*”, “*Artist*”, “*Customer*”, “*Employee*”, “*Genre*”, “*Invoice*”, “*Invoice Line*”, “*Media Type*”, “*Playlist*”, “*Playlist Track*”, dan “*Track*”. Setiap tabel menyimpan data khusus yang saling berhubungan, seperti informasi album di tabel album, rincian pelanggan di tabel pelanggan, dan detail transaksi di tabel *invoiceline*. Struktur tabel ini memungkinkan analisis data yang menyeluruh dan terorganisir.

#### 1. Tabel Album

Pada tabel ini berisi data album musik, mencakup *AlbumId*, *Title*, *ArtistId* (relasi ke tabel *Artist*).

album_id	title	artist_id
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
4	Let There Be Rock	1
5	Big Ones	3
6	Jagged Little Pill	4
7	Facelift	5
8	Warner 25 Anos	6
9	Plays Metallica By Four Cellos	7
10	Audioslave	8
11	Out Of Exile	8
12	BackBeat Soundtrack	9
13	The Best Of Billy Cobham	10
14	Alcohol Fueled Brewtality Live! [Disc 1]	11
15	Alcohol Fueled Brewtality Live! [Disc 2]	11
16	Black Sabbath	12
17	Black Sabbath Vol. 4 (Remaster)	12
18	Body Count	13
19	Chemical Wedding	14
20	The Best Of Buddy Guy - The Millenium C	15
21	Prenda Minha	16
22	Sozinho Remix Ao Vivo	16
23	Minha Historia	17

## 2. Tabel Artist

Pada table ini menyimpan informasi artis, seperti *ArtistId*, *Name*.

artist_id	name
1	AC/DC
2	Accept
3	Aerosmith
4	Alanis Morissette
5	Alice In Chains
6	Antônio Carlos Jobim
7	Apocalyptica
8	Audioslave
9	BackBeat
10	Billy Cobham
11	Black Label Society
12	Black Sabbath
13	Body Count
14	Bruce Dickinson
15	Buddy Guy
16	Caetano Veloso
17	Chico Buarque
18	Chico Science & Nação Zumbi
19	Cidade Negra
20	Claudio Zoli
21	Various Artists
22	Led Zeppelin
23	Frank Zappa & Captain Beefheart
24	Marcos Valle

## 3. Tabel Customer

Pada table ini berisi informasi pelanggan, seperti *CustomerId*, *FirstName*, *LastName*, *Company*, *Address*, *City*, *State*, *Country*, *PostalCode*, *Phone*, *Fax*, *Email*, *SupportRepId* (relasi ke tabel *Employee*).

customer_id	first_name	last_name	company	address	city
1	Luis	Gonzales	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José dos Ca
2	Leonie	Köhler	[NUL]	Theodor-Heuss-Straße 34	Stuttgart
3	François	Pierrel	[NUL]	1498 rue Bélanger	Montréal
4	Sven	Hansen	[NUL]	Ullevøien 14	Oslo
5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague
6	Helen	Holý	[NUL]	Růska 3174/6	Prague
7	Astrid	Gruber	[NUL]	Richterumstraße 4, 1010 Innere Stadt	Vienna
8	Daan	Pieters	[NUL]	Grifstraat 63	Brussels
9	Kara	Nielsen	[NUL]	Sander Boulevard 51	Copenhagen
10	Eduardo	Martins	Woodstock Discos	Rua Dr. Falcão Filho, 155	São Paulo
11	Alexandre	Rocha	Banco do Brasil S.A.	Av. Paulista, 2022	São Paulo
12	Roberto	Almeida	Riotur	Praça Pio X, 119	Rio de Janeiro
13	Fernanda	Ramos	[NUL]	Qe 7 Bloco G	Brasília
14	Mark	Philips	Telus	8210 111 St NW	Edmonton
15	Jennifer	Peterson	Rogers Canada	700 W Pender Street	Vancouver
16	Frank	Harris	Google Inc.	1600 Amphitheatre Parkway	Mountain View
17	Jack	Smith	Microsoft Corporation	1 Microsoft Way	Redmond
18	Michelle	Brooks	[NUL]	627 Broadway	New York
19	Tim	Goyer	Apple Inc.	1 Infinite Loop	Cupertino
20	Dun	Miller	[NUL]	541 Del Medio Avenue	Mountain View
21	Kathy	Chase	[NUL]	801 W 4th Street	Reno
22	Heather	Leacock	[NUL]	120 S Orange Ave	Orlando
23	John	Gordon	[NUL]	69 Salem Street	Boston

## 4. Tabel Employee

Pada table ini berisi informasi karyawan, termasuk *EmployeeId*, *LastName*, *FirstName*, *Title*, *ReportsTo*, *BirthDate*, *HireDate*, *Address*, *City*, *State*, *Country*, *PostalCode*, *Phone*, *Fax*, *Email*.

employee_id	last_name	first_name	title	reports_to	birth_date	hire_date	address
1	Adams	Andrew	General Manager	[NUL]	1962-02-18 00:00:00.000	2002-08-14 00:00:00.000	11120 Jasper
2	Edwards	Nancy	Sales Manager	1	1958-12-08 00:00:00.000	2002-05-01 00:00:00.000	825 8 Ave SW
3	Peacock	Jane	Sales Support Agent	2	1973-08-29 00:00:00.000	2002-04-01 00:00:00.000	1111 6 Ave SW
4	Park	Margaret	Sales Support Agent	2	1947-09-19 00:00:00.000	2003-05-03 00:00:00.000	683 10 Street
5	Johnson	Steve	Sales Support Agent	2	1965-03-03 00:00:00.000	2003-10-17 00:00:00.000	77278 41 Ave
6	Mitchell	Michael	IT Manager	1	1973-07-01 00:00:00.000	2003-10-17 00:00:00.000	5827 Bowness
7	King	Robert	IT Staff	6	1970-05-29 00:00:00.000	2004-01-02 00:00:00.000	590 Columbia
8	Callahan	Laura	IT Staff	6	1968-01-09 00:00:00.000	2004-03-04 00:00:00.000	923 7 ST NW

## 5. Tabel Genre

Pada table ini berisi kategori genre musik seperti Rock, Jazz, dan Pop yang mencakup GenreId, Name.

genre_id	name
1	Rock
2	Jazz
3	Metal
4	Alternative & Punk
5	Rock And Roll
6	Blues
7	Latin
8	Reggae
9	Pop
10	Soundtrack
11	Bossa Nova
12	Easy Listening
13	Heavy Metal
14	R&B/Soul
15	Electronica/Dance
16	World
17	Hip Hop/Rap
18	Science Fiction
19	TV Shows
20	Sci Fi & Fantasy
21	Drama
22	Comedy
23	Alternative
24	Classical

## 6. Tabel Invoice

Pada table ini menyimpan data transaksi pelanggan, seperti *InvoiceId*, *CustomerId*, *InvoiceDate*, *BillingAddress*, *BillingCity*, *BillingState*, *BillingCountry*, *BillingPostalCode*, *Total*.

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	bill
1	2	2021-01-01 00:00:00.000	Theodor-Heuss-Straße 34	Stuttgart	[NULL]	Germany	70174
2	4	2021-01-02 00:00:00.000	Ullevålsveien 14	Oslo	[NULL]	Norway	0171
3	8	2021-01-03 00:00:00.000	Grêtrystraat 63	Brussels	[NULL]	Belgium	1000
4	14	2021-01-06 00:00:00.000	8210 111 ST NW	Edmonton	AB	Canada	T6G 2
5	23	2021-01-11 00:00:00.000	69 Salem Street	Boston	MA	USA	2113
6	37	2021-01-19 00:00:00.000	Berger Straße 10	Frankfurt	[NULL]	Germany	60316
7	38	2021-02-01 00:00:00.000	Barbarossastraße 19	Berlin	[NULL]	Germany	10775
8	40	2021-02-01 00:00:00.000	8, Rue Hanovre	Paris	[NULL]	France	75002
9	42	2021-02-02 00:00:00.000	9, Place Louis Barthou	Bordeaux	[NULL]	France	33006
10	46	2021-02-03 00:00:00.000	3 Chatham Street	Dublin	Dublin	Ireland	[NULL]
11	52	2021-02-06 00:00:00.000	202 Hoxton Street	London	[NULL]	United Kingdom	N1 5L
12	2	2021-02-11 00:00:00.000	Theodor-Heuss-Straße 34	Stuttgart	[NULL]	Germany	70174
13	16	2021-02-19 00:00:00.000	1600 Amphitheatre Parkway	Mountain View	CA	USA	94043
14	17	2021-03-04 00:00:00.000	1 Microsoft Way	Redmond	WA	USA	98052
15	19	2021-03-04 00:00:00.000	1 Infinite Loop	Cupertino	CA	USA	95014
16	21	2021-03-05 00:00:00.000	801 W 4th Street	Reno	NV	USA	89503
17	25	2021-03-06 00:00:00.000	319 N. Frances Street	Madison	WI	USA	53703
18	31	2021-03-09 00:00:00.000	194A Chain Lake Drive	Halifax	NS	Canada	B3S 1
19	40	2021-03-14 00:00:00.000	8, Rue Hanovre	Paris	[NULL]	France	75002
20	54	2021-03-22 00:00:00.000	110 Raeburn Pl	Edinburgh	[NULL]	United Kingdom	EH4 1
21	55	2021-04-04 00:00:00.000	421 Bourke Street	Sidney	NSW	Australia	2010
22	57	2021-04-04 00:00:00.000	Calle Lira, 198	Santiago	[NULL]	Chile	[NULL]
23	59	2021-04-05 00:00:00.000	3,Raj Bhavan Road	Bangalore	[NULL]	India	56006

## 7. Tabel Invoice Line

Pada table ini menyimpan rincian setiap item pada transaksi tertentu yang meliputi *InvoiceLineId*, *InvoiceId*, *TrackId*, *UnitPrice*, *Quantity*.

121 invoice_line_id	120 invoice_id	120 track_id	123 unit_price	123 quantity
1	1	2	0.99	1
2	1	4	0.99	1
3	2	6	0.99	1
4	2	8	0.99	1
5	2	10	0.99	1
6	2	12	0.99	1
7	3	16	0.99	1
8	3	20	0.99	1
9	3	24	0.99	1
10	3	28	0.99	1
11	3	32	0.99	1
12	3	36	0.99	1
13	4	42	0.99	1
14	4	48	0.99	1
15	4	54	0.99	1
16	4	60	0.99	1
17	4	66	0.99	1
18	4	72	0.99	1
19	4	78	0.99	1
20	4	84	0.99	1
21	4	90	0.99	1
22	5	99	0.99	1
23	5	108	0.99	1
24	5	117	0.99	1

## 8. Tabel Media Type

Pada table ini menyimpan informasi tentang format media, seperti MP3 atau AAC yang mencakup *MediaTypeId*, *Name*.

124 media_type_id	A-Z name
1	MPEG audio file
2	Protected AAC audio file
3	Protected MPEG-4 video file
4	Purchased AAC audio file
5	AAC audio file

## 9. Tabel Playlist

Pada table ini berisi data tentang *playlist*, termasuk *PlaylistId*, *Name*

playlist_id	A-Z name
1	Music
2	Movies
3	TV Shows
4	Audiobooks
5	90's Music
6	Audiobooks
7	Movies
8	Music
9	Music Videos
10	TV Shows
11	Brazilian Music
12	Classical
13	Classical 101 - Deep Cuts
14	Classical 101 - Next Steps
15	Classical 101 - The Basics
16	Grunge
17	Heavy Metal Classic
18	On-The-Go 1

## 10. Tabel Playlist Track

Pada table ini menyimpan hubungan antara playlist dan lagu-lagu yang ada di dalamnya dengan meliputi *PlaylistId*, *TrackId* (relasi ke tabel *Playlist* dan *Track*).

playlist_id	track_id
1	3,402
1	3,389
1	3,390
1	3,391
1	3,392
1	3,393
1	3,394
1	3,395
1	3,396
1	3,397
1	3,398
1	3,399
1	3,400
1	3,401
1	3,336
1	3,478
1	3,375
1	3,376
1	3,377
1	3,378
1	3,379
1	3,380
1	3,381
1	3,382

## 11. Track

Pada table ini menyimpan data lagu, termasuk *TrackId*, *Name*, *AlbumId*, *MediaTypeId*, *GenreId*, *Composer*, *Milliseconds*, *Bytes*, *UnitPrice*

track_id	name	album_id	media_type_id	genre_id	composer
1	For Those About To Rock (We Salute You)	1	1	1	Angus Young, Malcolm Young, Brian Johnson
2	Balls to the Wall	2	2	1	U. Dirkschneider, W. Hoffmann, H. Frank, P. Bal
3	Fast As a Shark	3	2	1	F. Baltes, S. Kaufman, U. Dirkschneider & W. Hof
4	Restless and Wild	3	2	1	F. Baltes, R.A. Smith-Diesel, S. Kaufman, U. Dirk
5	Princess of the Dawn	3	2	1	Deaffy & R.A. Smith-Diesel
6	Put The Finger On You	1	1	1	Angus Young, Malcolm Young, Brian Johnson
7	Let's Get It Up	1	1	1	Angus Young, Malcolm Young, Brian Johnson
8	Inject The Venom	1	1	1	Angus Young, Malcolm Young, Brian Johnson
9	Snowballed	1	1	1	Angus Young, Malcolm Young, Brian Johnson
10	Evil Walks	1	1	1	Angus Young, Malcolm Young, Brian Johnson
11	C.O.D.	1	1	1	Angus Young, Malcolm Young, Brian Johnson
12	Breaking The Rules	1	1	1	Angus Young, Malcolm Young, Brian Johnson
13	Night Of The Long Knives	1	1	1	Angus Young, Malcolm Young, Brian Johnson
14	Spellbound	1	1	1	Angus Young, Malcolm Young, Brian Johnson
15	Go Down	4	1	1	AC/DC
16	Dog Eat Dog	4	1	1	AC/DC
17	Let There Be Rock	4	1	1	AC/DC
18	Bad Boy Boogie	4	1	1	AC/DC
19	Problem Child	4	1	1	AC/DC
20	Overdose	4	1	1	AC/DC
21	Hell Ain't A Bad Place To Be	4	1	1	AC/DC
22	Whole Lotta Rosie	4	1	1	AC/DC
23	Walk On Water	5	1	1	Steven Tyler, Joe Perry, Jack Blades, Tommy Shi

### 3.2 Perancangan Data Warehouse

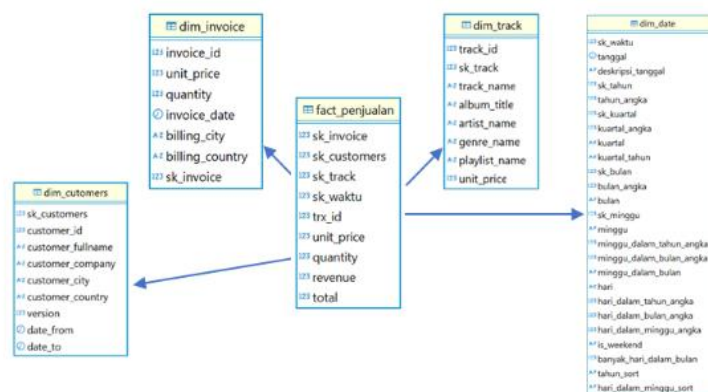
Penelitian ini menggunakan metodologi *Five – Step Design*, yang terdiri dari beberapa tahap utama

#### 1. Analisis Kebutuhan dan Pemilihan Proses Bisnis

Proses dimulai dengan menentukan kebutuhan informasi dari data yang tersedia dari Chinook Database, seperti data pelanggan, transaksi, produk, dan lainnya. Fokus utama adalah analisis *business intelligence* pada data chinook untuk optimalisasi penjualan music.

#### 2. Penentuan Struktur Data dan Hubungan Dimensi Fakta

Perancangan tabel fakta, seperti harga satuan produk, bergantung pada dimensi grain atau detail terkecil data. Dimensi yang relevan, seperti pelanggan, produk, faktur, dan waktu, diidentifikasi untuk membangun hubungan dalam model skema bintang untuk meningkatkan efisiensi data gudang.



*Model Tabel Fakta*

#### 3. Pemilihan Fakta dan Pre – Calculation

Tabel fakta penjualan dipilih berdasarkan gandum yang telah ditentukan. Kalkulasi sebelumnya, seperti total penjualan dan pendapatan, dibuat untuk membantu menggabungkan data dalam rentang waktu tertentu, seperti hari, bulan, atau tahun.

#### 4. Menambah Atribut Tabel Dimensi

Setiap tabel dimensi memiliki atribut yang relevan untuk mendukung analisis yang lebih mendalam. Misalnya, dimensi pelanggan mencakup data pelanggan seperti identitas, lokasi, dan perusahaan, sedangkan dimensi waktu mencakup detail kalender seperti tingkat harian.

#### 5. Waktu Penyimpanan Data dan Pemantauan Perubahan Dimensi

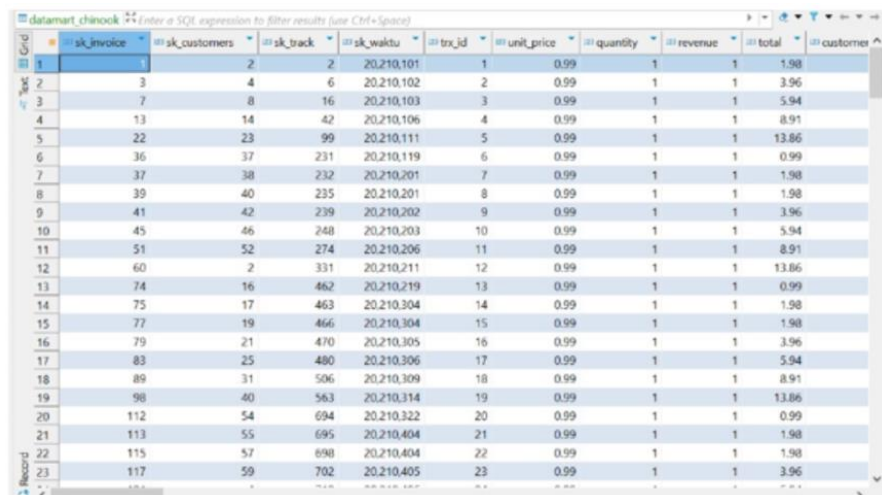
Waktu penyimpanan data diatur selama lima tahun untuk memastikan ketersediaan data sejarah. Perubahan atribut tabel dimensi juga dipantau melalui proses ETL yang berkelanjutan untuk mengakomodasi perubahan data.

### 3.3 Datamart Dan Datamart OLAP

#### 1. Datamart

Subset data yang disebut Datamart ditunjukkan pada gambar tersebut dan dimaksudkan untuk membantu analisis data pada tingkat tertentu, seperti penjualan. Dalam proses ini, fakta penjualan digabungkan dengan berbagai dimensi, seperti pelanggan (dim customers), faktur (dim invoice), jalur/item (dim track), dan tanggal (dim date).

Hasil akhirnya adalah tabel datamart yang menggabungkan data dengan atribut penting seperti kunci surrogate (sk\_customers, sk\_invoice, sk\_track, sk\_time), harga satuan (unit\_price), dan pendapatan (revenue). Analisis bisnis atau pelaporan berbasis data menjadi lebih mudah dengan bantuan datamart ini.



id	sk_invoice	sk_customers	sk_track	sk_waktu	trx_id	unit_price	quantity	revenue	total	customer
1	2	2	2	20.210.101	1	0.99	1	1	1.98	
2	3	4	6	20.210.102	2	0.99	1	1	3.96	
3	7	8	16	20.210.103	3	0.99	1	1	5.94	
4	13	14	42	20.210.106	4	0.99	1	1	8.91	
5	22	23	99	20.210.111	5	0.99	1	1	13.86	
6	36	37	231	20.210.119	6	0.99	1	1	0.99	
7	37	38	232	20.210.201	7	0.99	1	1	1.98	
8	39	40	235	20.210.201	8	0.99	1	1	1.98	
9	41	42	239	20.210.202	9	0.99	1	1	3.96	
10	45	46	248	20.210.203	10	0.99	1	1	5.94	
11	51	52	274	20.210.206	11	0.99	1	1	8.91	
12	60	2	331	20.210.211	12	0.99	1	1	13.86	
13	74	16	462	20.210.219	13	0.99	1	1	0.99	
14	75	17	463	20.210.304	14	0.99	1	1	1.98	
15	77	19	466	20.210.304	15	0.99	1	1	1.98	
16	79	21	470	20.210.305	16	0.99	1	1	3.96	
17	83	25	480	20.210.306	17	0.99	1	1	5.94	
18	89	31	506	20.210.309	18	0.99	1	1	8.91	
19	98	40	563	20.210.314	19	0.99	1	1	13.86	
20	112	54	694	20.210.322	20	0.99	1	1	0.99	
21	113	55	695	20.210.404	21	0.99	1	1	1.98	
22	115	57	698	20.210.404	22	0.99	1	1	1.98	
23	117	59	702	20.210.405	23	0.99	1	1	3.96	

#### 2. Datamart OLAP

Datamart OLAP adalah bagian penting dari *business intelligence (BI)* yang memungkinkan analisis data berbagai dimensi secara cepat dan efisien. Data ini disusun dalam kubus OLAP dan memiliki berbagai dimensi, seperti produk dan waktu, serta fakta, seperti penjualan. Sebelum disimpan untuk analisis, data mentah diubah melalui proses pembersihan dan pengayaan.

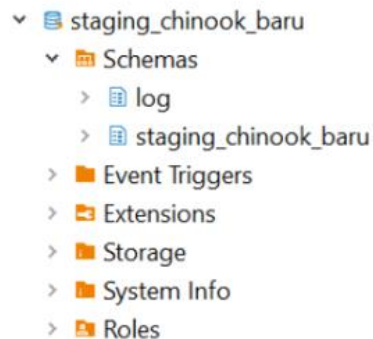
Struktur ini memberi pengguna kemampuan untuk mengeksplorasi data secara interaktif, melihat pola besar, atau mendalami detail. Datamart OLAP menjadi

alat strategis untuk analisis bisnis, membantu pengambilan keputusan berbasis data, dan menemukan tren tersembunyi.

tahun_angka	kuartal	bulan	customer_city	sk_customers	sk_invoice	sk_track	sk_waktu	revenue
2,021	Q1	February	Berlin	1	1	1	1	1
2,021	Q1	February	Bordeaux	1	1	1	1	1
2,021	Q1	February	Dublin	1	1	1	1	1
2,021	Q1	February	London	1	1	1	1	1
2,021	Q1	February	Mountain View	1	1	1	1	1
2,021	Q1	February	Paris	1	1	1	1	1
2,021	Q1	February	Stuttgart	1	1	1	1	1
2,021	Q1	January	Boston	1	1	1	1	1
2,021	Q1	January	Brussels	1	1	1	1	1
2,021	Q1	January	Edmonton	1	1	1	1	1
2,021	Q1	January	Frankfurt	1	1	1	1	1
2,021	Q1	January	Oslo	1	1	1	1	1
2,021	Q1	January	Stuttgart	1	1	1	1	1
2,021	Q1	March	Cupertino	1	1	1	1	1
2,021	Q1	March	Edinburgh	1	1	1	1	1
2,021	Q1	March	Halifax	1	1	1	1	1
2,021	Q1	March	Madison	1	1	1	1	1
2,021	Q1	March	Paris	1	1	1	1	1
2,021	Q1	March	Redmond	1	1	1	1	1
2,021	Q1	March	Reno	1	1	1	1	1
2,021	Q2	April	Bangalore	1	1	1	1	1
2,021	Q2	April	Cupertino	1	1	1	1	1
2,021	Q2	April	Oslo	1	1	1	1	1
2,021	Q2	April	Santiago	1	1	1	1	1

### 3.4 Pembuatan Staging Area

#### 3.4.1 Migrasi dan Pembuatan Log Untuk Database Staging Chinook



#### Database Staging Chinook

Langkah pertama dalam proses ini adalah membuat database baru yang bernama *Staging\_Chinook* di DBeaver. Setelah itu, untuk membuat kerangka tabel di dalam database baru tersebut, kita menggunakan schema *Public*. Data dan struktur tabel dari database *Chinook* akan dipindahkan ke database *Staging\_Chinook* dengan cara membuka database *Chinook*, kemudian membuka schema *Public*, dan memilih *Tables*. Semua tabel kemudian diblok, dan DDL (Data Definition Language) untuk semua tabel tersebut diambil dengan cara klik kanan dan memilih *Generate SQL* lalu memilih *DDL* dan menyalin script tersebut. Setelah itu, beralih ke database *Staging\_Chinook\_Baru*, membuka schema *Public*, dan mengklik kanan pada *SQL Editor* untuk membuat skrip SQL baru, kemudian mem-paste DDL yang

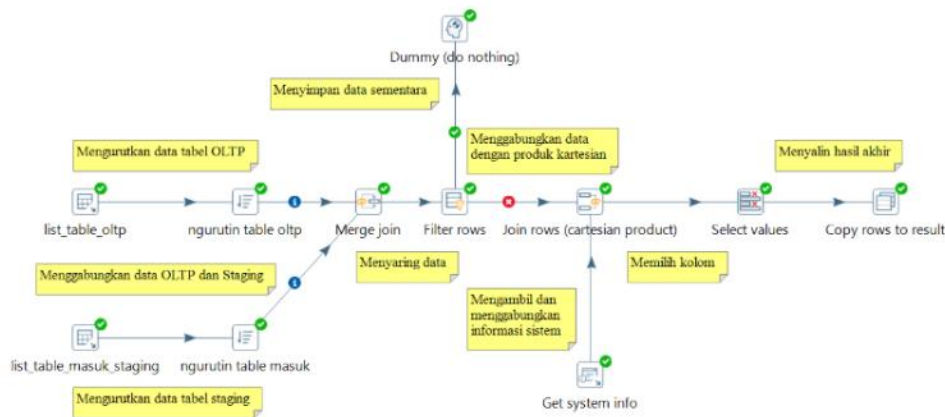


sudah disalin. Setelah itu, dengan menekan *CTRL+A* dan *CTRL+Enter*, skrip akan dijalankan dan tabel-tabel tersebut akan terbentuk di schema *Public*. Untuk memverifikasi, buka kembali schema *Public* dan lakukan *Refresh* untuk memastikan tabel-tabel sudah ada meskipun belum ada data di dalamnya.

Selanjutnya, langkah kedua adalah membuat *log table* untuk mencatat proses pemindahan data. Tabel ini berisi kolom-kolom seperti *table\_name*, *source\_date*, *source\_records*, *target\_date*, dan *target\_records*. Kemudian, untuk merapikan struktur database, schema *Public* diubah menjadi *Staging* dengan cara mengklik kanan pada schema *Public*, memilih *Rename*, dan mengeksekusi perubahan tersebut. Setelah itu, schema baru bernama *Log* dibuat dengan cara mengklik kanan pada schema dan memilih *Create New Schema*, memberikan nama *Log*, dan menjalankan skrip SQL untuk pembuatan schema tersebut.

Langkah ketiga adalah menggunakan Pentaho untuk membuat *Job* baru. Setelah menyimpan job tersebut, sebuah folder bernama *Case\_Staging\_Chinook* dibuat, dan job tersebut diberi nama *Main Job*. Proses job dimulai dengan langkah *Start*, kemudian langkah berikutnya adalah menambahkan step Transformasi dan menghubungkannya. Setelah itu, kita membuat transformasi baru sesuai kebutuhan pemindahan dan pengolahan data.

### 3.4.2 Transformasi List Tabel



#### *Transformasi List Table*

Dalam proses transformasi data, langkah pertama adalah mengurutkan data yang diambil dari *list\_table\_oltp* berdasarkan kriteria tertentu menggunakan langkah mengurutkan *table oltp*, yang memudahkan pemrosesan data lebih lanjut. Selanjutnya, data dari *list\_table\_masuk\_staging* juga diurutkan dalam langkah mengurutkan *table masuk* untuk mempersiapkan data sebelum melanjutkan ke langkah berikutnya.

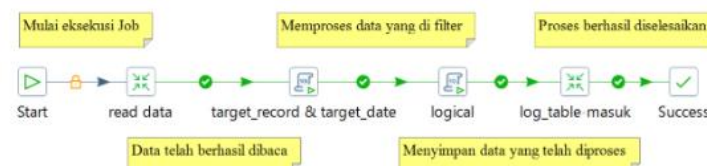
Setelah kedua dataset diurutkan, data dari mengurutkan *table oltp* digabungkan dengan dataset lain melalui langkah *Merge join*, menghasilkan dataset yang lebih lengkap dan siap untuk diproses lebih lanjut. Begitu juga, data yang diurutkan dari *ngurutin table masuk* digabungkan dengan data dari OLTP dalam langkah *Merge join* untuk memastikan kedua sumber data terintegrasi dengan baik.

Setelah penggabungan, data yang telah digabungkan kemudian difilter menggunakan langkah *Filter rows*, untuk mendapatkan subset data yang relevan sesuai dengan kriteria yang ditentukan. Selanjutnya, data yang sudah difilter diteruskan ke langkah *Dummy (do nothing)*, yang berfungsi untuk tidak mengolah data lebih lanjut, mungkin digunakan untuk pengujian atau sebagai placeholder dalam alur transformasi data.

Setelah melalui langkah *Dummy (do nothing)*, data kemudian digabungkan lagi dengan dataset lain melalui langkah *Join rows (cartesian product)*, memperluas dataset lebih lanjut. Pada langkah *Select values*, kolom dan nilai yang relevan dipilih untuk menyiapkan data sesuai kebutuhan, memfokuskan data pada informasi yang dibutuhkan untuk analisis atau tujuan lainnya.

Akhirnya, data yang sudah dipilih dan disaring diteruskan ke langkah *Copy rows to result*, yang merupakan output akhir dari proses transformasi data. Di sisi lain, informasi sistem yang relevan juga diambil dan digabungkan dengan dataset melalui langkah *Join rows (cartesian product)* untuk melengkapi informasi yang diperlukan dalam proses ini, sehingga menghasilkan dataset akhir yang lengkap dan siap digunakan.

### 3.4.3 Job Execute Data



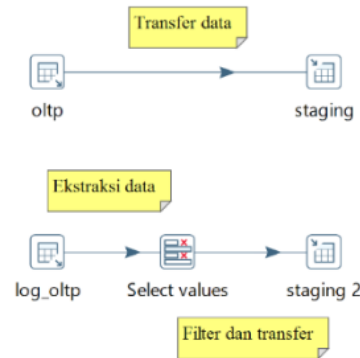
#### *Job Execute Data*

Proses dimulai dengan langkah pertama, yaitu *read data*, di mana data diambil atau dibaca dari sumber yang telah ditentukan. Setelah data berhasil dibaca, pekerjaan berlanjut ke langkah *target\_record & target\_date*, di mana data difilter atau diproses berdasarkan catatan target dan tanggal target yang relevan. Selanjutnya, setelah data difilter, langkah berikutnya melibatkan penerapan operasi logis pada data, seperti validasi atau pemeriksaan kondisi tertentu untuk memastikan bahwa data memenuhi kriteria yang diinginkan.

Setelah operasi logis diterapkan, data kemudian disimpan ke dalam tabel log yang disebut *log\_table-masuk*. Langkah ini bertujuan untuk menulis data ke dalam basis data atau tabel log, yang digunakan untuk tujuan audit atau pelacakan proses. Setelah data berhasil disimpan, pekerjaan berlanjut ke

langkah terakhir, yaitu *Success*, yang menandakan bahwa seluruh proses eksekusi pekerjaan telah selesai dengan sukses tanpa kendala.

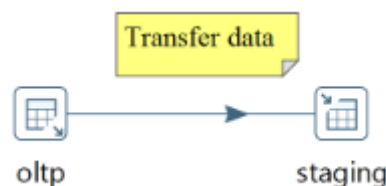
#### 3.4.4 Transformasi Read Data



##### *Transformasi Read Data*

Proses ini dimulai dengan mentransfer data dari sumber OLTP ke area *staging*, yang berfungsi sebagai tempat penyimpanan sementara sebelum data diproses lebih lanjut atau dipindahkan ke penyimpanan akhir. Setelah data berada di area *staging*, langkah selanjutnya adalah mengekstrak data dari sumber *log\_oltp*, yang berisi data log dari sistem OLTP. Data ini perlu diproses atau difilter agar sesuai dengan kebutuhan sebelum digunakan lebih lanjut. Setelah itu, data yang telah diekstraksi melalui langkah *Select values* akan disaring atau ditransformasikan, dengan memilih kolom-kolom tertentu atau menerapkan transformasi pada data sesuai dengan tujuan yang diinginkan. Setelah langkah ini, data siap untuk dipindahkan ke *staging 2* atau diproses lebih lanjut.

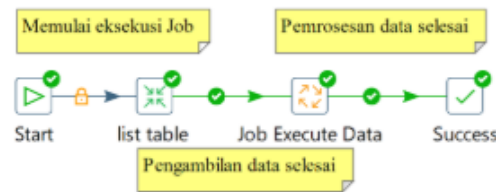
#### 3.4.5 Transformasi Log Table Masuk



##### *Transformasi Log Tabel Masuk*

Proses pemindahan data dari sistem OLTP (*Online Transaction Processing*) ke Staging. Di sini, data yang berasal dari transaksi harian di OLTP, seperti pemesanan atau penjualan, dipindahkan ke Staging untuk diproses lebih lanjut. Staging berfungsi sebagai area penyimpanan sementara, memungkinkan data disaring dan dipersiapkan sebelum dimuat ke dalam sistem data permanen, seperti data warehouse atau tabel output. Proses ini memastikan data siap untuk analisis atau pemrosesan lebih lanjut.

### 3.4.6 Main Job



### Transformasi Main Job

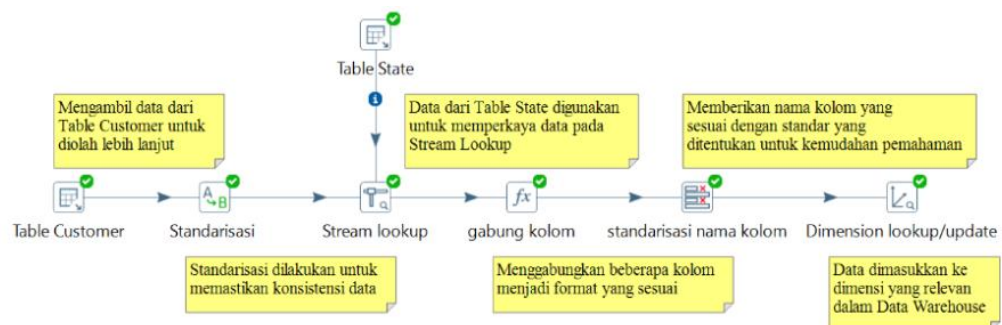
Proses ini dimulai dengan langkah pertama, yaitu *list table*, yang bertujuan untuk mengambil atau mencatat data dari tabel yang relevan. Setelah data berhasil diambil, pekerjaan berlanjut ke langkah *Job Execute Data*, yang merupakan tahap pemrosesan atau transformasi data yang telah dikumpulkan. Setelah *Job Execute Data* selesai dieksekusi dengan sukses, proses dilanjutkan ke langkah terakhir, yaitu *Success*, yang menandakan bahwa seluruh rangkaian pekerjaan telah berhasil dieksekusi tanpa ada kendala.

## 3.5 Implementasi Model Multidimensi

### 3.5.1 Skema dw\_sales

Pada tahap perancangan skema, kami menentukan model Star Schema untuk mendesain data warehouse. Dalam model ini, database *dw\_sales* memiliki beberapa tabel dimensi yang mendukung analisis data. Berikut adalah tabel dimensi dari *dw\_sales*:

#### 1. Dimensi Customer



### Transformasi Dim Customer

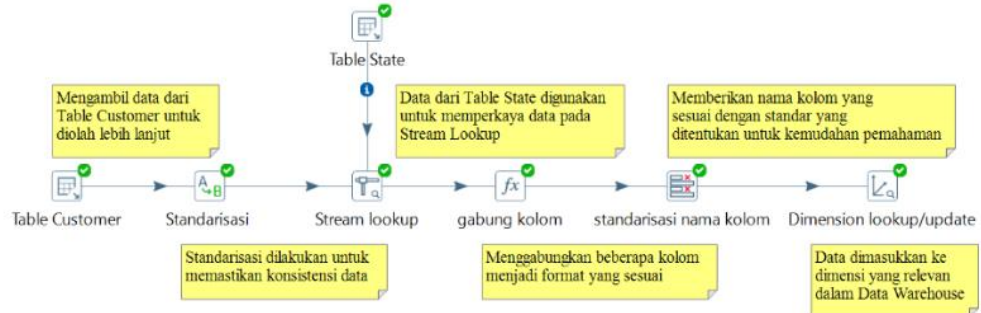
Proses ini dimulai dengan mengambil data dari sumber *Table Customer* dan meneruskannya ke langkah berikutnya untuk melakukan standarisasi region. Setelah data distandarisasi, proses berlanjut ke langkah *Stream lookup*, di mana data dari sumber lain ditambahkan berdasarkan key tertentu. Proses lookup ini kemudian menggabungkan data tersebut untuk menyusun dataset sesuai dengan kebutuhan analisis.

Selanjutnya, kolom-kolom yang dihasilkan dari proses lookup disusun atau digabungkan untuk memastikan standarisasi nama kolom. Setelah nama kolom distandarisasi, data diteruskan ke langkah *Dimension lookup/update*, yang

bertujuan untuk memetakan data ke dimensi yang sesuai di dalam data warehouse.

Pada langkah terakhir, data dimasukkan ke dalam tabel dimensi di data warehouse, dengan proses pembaruan (update) yang dilakukan jika terdapat data yang sudah ada sebelumnya.

## 2. Dimensi Invoice



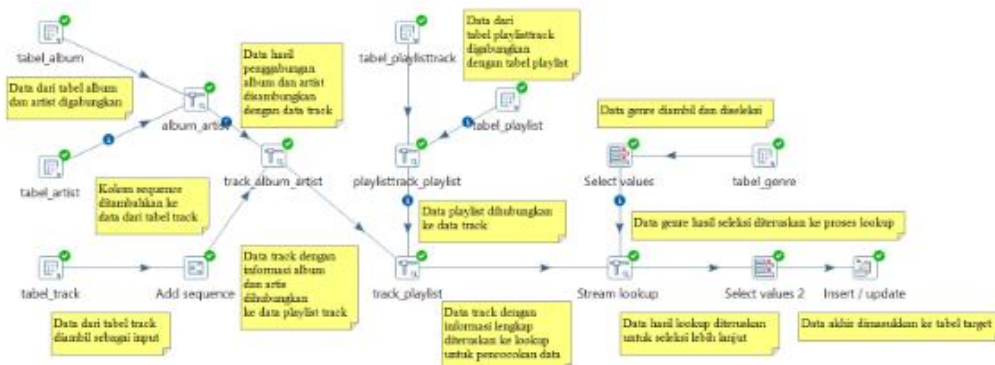
### Transformasi Dim Invoice

Proses dimulai dengan meneruskan data dari tabel *invoice* untuk digabungkan dengan data lain pada langkah *Merge join*, yang biasanya dilakukan berdasarkan key yang sama, seperti *invoice\_id*. Setelah itu, data dari tabel *invoice line*, yang berisi rincian item pada faktur, diteruskan dan digabungkan dengan data dari tabel *invoice* menggunakan *Merge join*.

Hasil gabungan data dari langkah *Merge join* selanjutnya diteruskan ke langkah *Add sequence*, yang memberikan nomor urut unik pada setiap baris data. Kolom tambahan yang berisi urutan (sequence) kemudian diteruskan ke langkah *Select values*, di mana kolom yang dibutuhkan akan dipilih atau disusun ulang.

Terakhir, setelah kolom yang diperlukan telah dipilih, data diteruskan ke *Table output*, tempat data disimpan ke dalam database atau target lainnya. Proses ini memastikan bahwa data yang digabungkan, diberi nomor urut, dan disusun sesuai kebutuhan akan disimpan dengan rapi untuk analisis atau pemrosesan lebih lanjut.

## 3. Dimensi Track



### Transformasi Dim Track

Proses dimulai dengan menyiapkan data dari tabel track, di mana kolom sequence akan ditambahkan untuk menjadi kunci unik (surrogate key) pada setiap baris data. Setelah itu, hasil penambahan sequence ini dihubungkan ke

Setelah memastikan data memiliki format dan kolom yang sesuai, data hasil seleksi akhirnya dimasukkan atau diperbarui dalam database target. Jika data sudah ada, proses ini akan memperbarui informasi yang sudah ada; jika data belum ada, maka data baru akan ditambahkan. Proses ini memastikan data yang lengkap dan terstruktur dengan baik siap untuk digunakan atau dianalisis lebih lanjut.

Pada langkah berikutnya, nama hari disesuaikan agar formatnya konsisten melalui normalisasi string, memastikan nama hari seperti "Monday" memiliki format yang seragam. Selain itu, data pelanggan digunakan untuk



menghubungkan dan mengidentifikasi track yang terkait dengan setiap transaksi pelanggan, sehingga analisis yang dilakukan dapat mencakup informasi tentang pelanggan serta lagu yang dibeli.

Selanjutnya, format nama bulan distandarkan agar seragam, misalnya dengan memastikan semua huruf kapital pertama atau mengikuti format lainnya. Data nama hari yang sudah dinormalisasi diproses lebih lanjut untuk dimasukkan ke kolom hasil akhir, demikian pula dengan data nama bulan yang telah dinormalisasi, yang juga disiapkan untuk langkah berikutnya atau disimpan di kolom hasil akhir.

Proses kemudian berlanjut dengan seleksi data dari langkah-langkah sebelumnya, termasuk nama hari dan bulan yang sudah diolah, untuk diproses lebih lanjut. Pada langkah ini, perhitungan atau manipulasi tambahan dapat dilakukan, seperti menambahkan kolom khusus dengan formula tertentu. Hanya kolom-kolom yang diperlukan yang diteruskan ke langkah berikutnya, memastikan struktur data final yang tepat.

Akhirnya, data hasil akhir disimpan ke dalam tabel *dim\_date*, yang berfungsi sebagai dimensi waktu yang akan digunakan dalam analisis Business Intelligence.

## 5. Tabel Fakta Penjualan



### *Transformasi Table Fakta*

Proses dimulai dengan memproses data dari tabel *Invoice* dan memuatnya ke langkah *Database Invoice* untuk memperoleh informasi rinci terkait faktur yang dibutuhkan dalam transformasi data. Selanjutnya, data pelanggan digunakan untuk menghubungkan dan mengidentifikasi track yang terkait dengan setiap transaksi, sehingga analisis dapat mencakup informasi mengenai pelanggan dan lagu yang dibeli.

Pada tahap ini, identifikasi pelanggan dari setiap transaksi berdasarkan faktur memungkinkan analisis yang lebih mendalam terkait pelanggan. Selain itu, data track diperluas dengan informasi waktu, seperti tanggal transaksi atau pembelian terkait, yang mendukung analisis berbasis waktu.

Data pelanggan kemudian digunakan dalam perhitungan pendapatan (*revenue*) berdasarkan transaksi yang dilakukan oleh masing-masing pelanggan.

Informasi waktu dimanfaatkan untuk membuat analisis pendapatan lebih terstruktur, seperti analisis pendapatan per periode tertentu.

Selanjutnya, data pendapatan diberi urutan unik menggunakan langkah *Add Sequence*, yang membantu dalam identifikasi atau pengelolaan data lebih lanjut. Setelah data diurutkan, hanya kolom yang relevan yang dipilih dan

difilter melalui langkah *Select Values* untuk memastikan hanya informasi penting yang diteruskan ke langkah selanjutnya.  
Akhirnya, data yang telah difilter dimasukkan ke dalam *Table Output*, yang merupakan hasil akhir dari transformasi data dan siap digunakan untuk analisis atau pelaporan lebih lanjut..

### 3.6 Implementasi Mondrian OLAP CUBE

#### 1. fact\_penjualan (Menganalisis Dimensi Invoice)

Mendefinisikan struktur CUBE yang menghubungkan tabel fakta (fact\_penjualan) dengan dimensi Invoice.

- Hasil

Query SALES using Mondrian OLAP



				Measures			
Invoice	Customers	Track	Date	TotalPrice	TotalQuantity	TotalRevenue	TotalHasilPenjualan
-All Invoice	+All Customers	+All Track	+All Date	432	412	432	2,329
-0.99	+All Customers	+All Track	+All Date	392	392	392	2,152
-1	+All Customers	+All Track	+All Date	392	392	392	2,152
- Amsterdam	+All Customers	+All Track	+All Date	6	6	6	32
- Netherlands	+All Customers	+All Track	+All Date	6	6	6	32
(Anesthesia) Pulling Teeth	+All Customers	+All Track	+All Date				
Anthem	+All Customers	+All Track	+All Date				
Avisa	+All Customers	+All Track	+All Date				
Burn Away	+All Customers	+All Track	+All Date				
Cold Turkey	+All Customers	+All Track	+All Date				
Desaforo	+All Customers	+All Track	+All Date				
Falamansa Song	+All Customers	+All Track	+All Date				
Fica	+All Customers	+All Track	+All Date				
Get On Top	+All Customers	+All Track	+All Date				
If You Don't Know Me By Now	+All Customers	+All Track	+All Date				
If You Have To Ask	+All Customers	+All Track	+All Date				
Instant Karma	+All Customers	+All Track	+All Date				
Jump In The Fire	+All Customers	+All Track	+All Date	1	1	1	2
Just Ain't Good Enough	+All Customers	+All Track	+All Date				
La Puesta Del Sol	+All Customers	+All Track	+All Date				
Let Me Love You Baby	+All Customers	+All Track	+All Date	1	1	1	4
O Bra√bo Da Minha Guitarra	+All Customers	+All Track	+All Date				
Oh, My Love	+All Customers	+All Track	+All Date				
Os Exilados	+All Customers	+All Track	+All Date				
Quem Mata A Mulher Mata O Melhor	+All Customers	+All Track	+All Date	1	1	1	2
Right On Time	+All Customers	+All Track	+All Date				
Riviera Paradise	+All Customers	+All Track	+All Date				
Rock And Roll All Nite	+All Customers	+All Track	+All Date	1	1	1	1
Sapato 36	+All Customers	+All Track	+All Date	1	1	1	14
Scratch-N-Sniff	+All Customers	+All Track	+All Date				
Tear	+All Customers	+All Track	+All Date				
The Last Song	+All Customers	+All Track	+All Date				
This Is The Place	+All Customers	+All Track	+All Date				

#### 2. fact\_penjualan (Menganalisis Dimensi Customer)

Mendefinisikan struktur CUBE yang menghubungkan tabel fakta (fact\_penjualan) dengan dimensi Customer.

- Hasil



### Query SALES using Mondrian OLAP



Invoice	Customers	Track	Date	Measures			
				TotalPrice	TotalQuantity	TotalRevenue	TotalHasilPenjualan
+All Invoice	+All Customers	+All Track	+All Date	432	412	432	2,329
	+ #null	+All Track	+All Date				
	+Aaron Mitchell	+All Track	+All Date	7	7	7	38
	+ #null	+All Track	+All Date	7	7	7	38
	+Winnipeg	+All Track	+All Date	7	7	7	38
	+Canada	+All Track	+All Date	7	7	7	38
	+Alexandre Rocha	+All Track	+All Date	7	7	7	38
	+Astrid Gruber	+All Track	+All Date	8	7	8	43
	+Bjv/rn Hansen	+All Track	+All Date	7	7	7	40
	+Camille Bernard	+All Track	+All Date	8	7	8	39
	+Daan Peeters	+All Track	+All Date	7	7	7	38
	+Dan Miller	+All Track	+All Date	8	7	8	40
	+Diego Gutiv@rrez	+All Track	+All Date	7	7	7	38
	+Dominique Lefebvre	+All Track	+All Date	8	7	8	39
	+Eduardo Martins	+All Track	+All Date	7	7	7	38
	+Edward Francis	+All Track	+All Date	7	7	7	38
	+Ellie Sullivan	+All Track	+All Date	7	7	7	38
	+Emma Jones	+All Track	+All Date	7	7	7	38
	+Enrique MuvToz	+All Track	+All Date	7	7	7	38
	+Fernanda Ramos	+All Track	+All Date	7	7	7	38
	+Frank Harris	+All Track	+All Date	7	7	7	38
	+Frank Ralston	+All Track	+All Date	9	7	9	44
	+František Wichterlov	+All Track	+All Date	7	7	7	41
	+François Tremblay	+All Track	+All Date	8	7	8	40
	+Fynn Zimmermann	+All Track	+All Date	7	7	7	44
	+Hannah Schneider	+All Track	+All Date	7	7	7	38
	+Heather Leacock	+All Track	+All Date	8	7	8	40
	+Helena HolvĹ	+All Track	+All Date	7	7	7	50
	+Hugh O'Reilly	+All Track	+All Date	8	7	8	46
	+Isabelle Mercier	+All Track	+All Date	7	7	7	41
	+Jack Smith	+All Track	+All Date	7	7	7	40
	+Jennifer Peterson	+All Track	+All Date	7	7	7	39
	+Joakim Johansson	+All Track	+All Date	7	7	7	39

- fact\_penjualan (Menganalisis Dimensi Track)  
Mendefinisikan struktur CUBE yang menghubungkan tabel fakta (fact\_penjualan) dengan dimensi Track.

- Hasil

Query SALES using Mondrian OLAP



Invoice	Customers	Track	Date	Measures			
				TotalPrice	TotalQuantity	TotalRevenue	TotalHasilPenjualan
+All Invoice	+All Customers	+All Track	+All Date	432	412	432	2,329
		+ "40"	+All Date				
		+War	+All Date				
		+U2	+All Date				
		+Rock	+All Date				
		+Music	+All Date				
		+ 0.99	+All Date				
		+ "2"	+All Date				
		+ "Eine Kleine Nachtmusik" Serenade In G, K. 525: I. Allegro	+All Date				
		+ #1 Zero	+All Date				
		+ #9 Dream	+All Date				
		+ Round Midnight	+All Date				
		+ (Anesthesia) Pulling Teeth	+All Date	1	1	1	2
		+ (Da Le) Yaeo	+All Date				
		+ (I Can't Help) Falling In Love With You	+All Date				
		+ (Oh) Pretty Woman	+All Date				
		+ (There Is) No Greater Love (Too Licks)	+All Date				
		+ (We Are) The Road Crew	+All Date				
		+ (White Man) In Hammersmith Palais	+All Date				
		+ (Wish I Could) Hideaway	+All Date	1	1	1	6
		+...And Found	+All Date				
		+...And Justice For All	+All Date				
		+...In Translation	+All Date				
		+ 0.7%	+All Date				
		+01 - Prowler	+All Date				
		+02 - Sanctuary	+All Date				
		+03 - Remember Tomorrow	+All Date				
		+04 - Running Free	+All Date				
		+05 - Phantom of the Opera	+All Date				
		+06 - Transylvania	+All Date				
		+07 - Strange World	+All Date				
		+08 - Charlotte the Harlot	+All Date				
		+09 - Iron Maiden	+All Date				

- fact\_penjualan (Menganalisis Dimensi Date)  
Mendefinisikan struktur CUBE yang menghubungkan tabel fakta (fact\_penjualan) dengan dimensi Date.

- Hasil

Query SALES using Mondrian OLAP



				Measures			
Invoice	Customers	Track	Date	• TotalPrice	• TotalQuantity	• TotalRevenue	• TotalHasilPenjualan
+All Invoice	+All Customers	+All Track	-All Date	432	412	432	2,329
			01-01-1946				
			01-01-1947				
			01-01-1948				
			01-01-1949				
			01-01-1950				
			01-01-1951				
			01-01-1952				
			01-01-1953				
			01-01-1954				
			01-01-1955				
			01-01-1956				
			01-01-1957				
			01-01-1958				
			01-01-1959				
			01-01-1960				
			01-01-1961				
			01-01-1962				
			01-01-1963				
			01-01-1964				
			01-01-1965				
			01-01-1966				
			01-01-1967				
			01-01-1968				
			01-01-1969				
			01-01-1970				
			01-01-1971				
			01-01-1972				
			01-01-1973				
			01-01-1974				
			01-01-1975				
			01-01-1976				
			01-01-1977				

## LAMPIRAN

### 1. fact.xml

```
<?xml version="1.0"?>
<Schema name="fact">
<!--
    == This software is subject to the terms of the Eclipse Public License v1.0
    == Agreement, available at the following URL:
    == http://www.eclipse.org/legal/epl-v10.html.
    == You must accept the terms of that agreement to use this software.
    ==
    == Copyright (C) 2000-2005 Julian Hyde
    == Copyright (C) 2005-2011 Pentaho and others
    == All Rights Reserved.
    -->

<!-- Shared dimensions -->

<!-- Sales -->
<!-- Sales -->
<Cube name="Fact" defaultMeasure="Revenue">
  <Table name="fact_penjualan"/>

  <Dimension name="Invoice" foreignKey="sk_invoice">
    <Hierarchy hasAll="true" allMemberName="All Invoice" primaryKey="sk_invoice">
      <Table name="dim_invoice" />

      <Level name="Price" column="unit_price"/>
      <Level name="Quantity" column="quantity"/>
      <Level name="City" column="billing_city"/>
      <Level name="Country" column="billing_country"/>
      <Level name="Track" column="name"/>

    </Hierarchy>
  </Dimension>

  <Dimension name="Customers" foreignKey="sk_customers">
    <Hierarchy hasAll="true" allMemberName="All Customers" primaryKey="sk_customers">
      <Table name="dim_cutomers" />

      <Level name="CustomerName" column="customer_fullname"/>
      <Level name="CustomerCompany" column="customer_company"/>
      <Level name="CustomerCity" column="customer_city"/>
      <Level name="CustomerCountry" column="customer_country"/>
    </Hierarchy>
  </Dimension>

  <Dimension name="Track" foreignKey="sk_track">
    <Hierarchy hasAll="true" allMemberName="All Track" primaryKey="sk_track">
      <Table name="dim_track" />

      <Level name="TrackName" column="track_name"/>
      <Level name="AlbumTitle" column="album_title"/>
      <Level name="Artis" column="artist_name"/>
      <Level name="Genre" column="genre_name"/>
      <Level name="PlaylistName" column="playlist_name"/>
      <Level name="UnitPrice" column="unit_price"/>
    </Hierarchy>
  </Dimension>
</Cube>
```

```

<Dimension name="Date" foreignKey="sk_waktu">
  <Hierarchy hasAll="true" allMemberName="All Date" primaryKey="sk_waktu">
    <Table name="dim_date" />

    <Level name="Tanggal" column="deskripsi_tanggal"/>
  </Hierarchy>
</Dimension>

<!-- Measures -->
  <Measure name="TotalPrice" column="unit_price" aggregator="sum"
formatString="Standard"/>
  <Measure name="TotalQuantity" column="quantity" aggregator="sum"
formatString="Standard"/>
  <Measure name="TotalRevenue" column="revenue" aggregator="sum"
formatString="Standard"/>
  <Measure name="TotalHasilPenjualan" column="total" aggregator="sum"
formatString="Standard"/>

</Cube>

</Schema>

```

## 2. fact.jsp

```

<%@ 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/dwh?user=root&password"
  catalogUri="/WEB-INF/queries/fact.xml"
>
  select
  {[Measures].[TotalPrice],[TotalQuantity],[TotalRevenue],[TotalHasilPenjualan]}
  ON COLUMNS, {[Invoice],[Customers],[Track],[Date]} ON ROWS from [Fact]
</jp:mondrianQuery>

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

```

<https://github.com/sayananda11/Chinook.git>