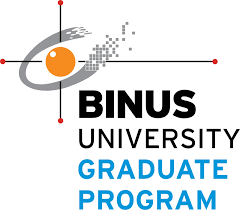
# **Transformasi Pengelolaan Data Streaming Musik Melalui Model Star Schema**



**Kelompok 10:**

Sarah Safitri - 2440027511

Vincent - 2440110886

Wanda Safira - 2440109556

**Business Intelligence and Analytics**

**UNIVERSITAS BINA NUSANTARA**

**JAKARTA**

**2024**

# **DAFTAR ISI**

[**DAFTAR ISI** i](#_Toc172132848)

[**BAB I** 1](#_Toc172132849)

[**PENDAHULUAN** 1](#_Toc172132850)

[**1.1** **Latar Belakang** 1](#_Toc172132851)

[**BAB II** 3](#_Toc172132852)

[**TINJAUAN LITERATUR** 3](#_Toc172132853)

[**2.1** **Streaming Musik** 3](#_Toc172132854)

[**2.2** **Business Intelligence (BI) dan Analytics** 3](#_Toc172132855)

[**2.3** **Structured Query Language (SQL)** 3](#_Toc172132856)

[**2.4** **Pentaho** 4](#_Toc172132857)

[**2.5** **OLTP** 4](#_Toc172132858)

[**2.6** **OLAP** 5](#_Toc172132859)

[**BAB III** 6](#_Toc172132860)

[**HASIL DAN PEMBAHASAN** 6](#_Toc172132861)

[**3.1** **Proses ETL dan Arsitektur Data Warehouse** 6](#_Toc172132862)

[**3.2** **Perancang Data Transaksi** 7](#_Toc172132863)

[**3.2.1** **Entitas** 8](#_Toc172132864)

[**3.2.2** **Relasi** 12](#_Toc172132865)

[**3.2.3** **Use Case** 12](#_Toc172132866)

[**3.3** **Script SQL Data Transaksi** 14](#_Toc172132867)

[**3.3.1** **Contoh Data untuk Setiap Tabel** 17](#_Toc172132868)

[**3.4** **Perancangan Star Schema Data Warehouse** 20](#_Toc172132869)

[**3.5** **Edit Transform Load  (ETL) Menggunakan Pentaho** 24](#_Toc172132870)

[**3.5.1** **Pembuatan Tabel Dimensi** 24](#_Toc172132871)

[**3.5.2** **Pembuatan Tabel Transaksi** 31](#_Toc172132872)

[**3.5.3** **Pembuatan Tabel Fact** 33](#_Toc172132873)

[**3.6** **Pembuatan Dashboard Menggunakan Power BI** 46](#_Toc172132874)

[**BAB IV** 48](#_Toc172132875)

[**KESIMPULAN** 48](#_Toc172132876)

[**DAFTAR PUSTAKA** 49](#_Toc172132877)

# **BAB I**

# **PENDAHULUAN**

1. **Latar Belakang**

Dalam era digital saat ini, aplikasi streaming musik telah menjadi bagian integral dari kehidupan sehari-hari masyarakat. Pengguna dapat dengan mudah mengakses jutaan lagu dari berbagai genre dan artis di seluruh dunia. Di balik kemudahan ini, terdapat kebutuhan mendesak untuk pengelolaan data yang efisien dan efektif. Pembuatan basis data yang optimal menjadi salah satu kunci utama untuk mendukung operasional aplikasi streaming musik dengan performa tinggi. Aplikasi streaming musik terbesar mengelola data dalam jumlah yang sangat besar, mencakup data pengguna, data lagu, data artis, data album, dan data aktivitas pengguna. Setiap kali pengguna melakukan pencarian, memutar lagu, membuat playlist, atau memberikan rating, semua data tersebut harus tercatat dan diolah dengan cepat dan akurat. Oleh karena itu, arsitektur basis data yang digunakan harus mampu menangani skala dan kompleksitas data tersebut.

Salah satu pendekatan yang dapat diimplementasikan untuk mengoptimalkan pengelolaan data adalah dengan menggunakan skema bintang (star schema). Star schema adalah sebuah model data yang digunakan dalam sistem gudang data (data warehouse), yang dirancang untuk memudahkan kueri analitis dan laporan bisnis. Dalam skema bintang, data diatur dalam bentuk tabel fakta yang terhubung dengan beberapa tabel dimensi. Tabel fakta menyimpan data transaksi atau peristiwa utama, sementara tabel dimensi menyediakan konteks deskriptif untuk data di tabel fakta. Penggunaan star schema pada aplikasi streaming musik memiliki beberapa keuntungan, antara lain peningkatan kinerja kueri, kesederhanaan dan intuitifitas, serta dukungan analisis data yang mendalam. Dengan struktur star schema, kueri analitis dapat dieksekusi lebih cepat karena tabel fakta dan tabel dimensi yang dirancang untuk mendukung operasi baca intensif. Model ini lebih mudah dipahami dan diimplementasikan dibandingkan dengan skema database yang lebih kompleks, seperti snowflake schema atau skema normalisasi tinggi. Selain itu, star schema memungkinkan pembuatan laporan dan analisis data yang mendalam, seperti analisis tren mendengarkan musik, preferensi pengguna, dan performa artis atau album.

Dalam konteks aplikasi streaming musik, tabel fakta dapat mencakup data tentang aktivitas mendengarkan musik, seperti waktu pemutaran, durasi pemutaran, dan lagu yang diputar. Tabel dimensi dapat mencakup data pengguna (misalnya, usia, jenis kelamin, lokasi), data lagu (misalnya, judul lagu, genre, artis), dan data waktu (misalnya, tanggal, bulan, tahun). Dengan latar belakang ini, pembuatan basis data menjadi star schema diharapkan dapat meningkatkan efisiensi operasional dan mendukung kebutuhan analitis aplikasi streaming musik, sehingga dapat memberikan pengalaman pengguna yang lebih baik dan informasi yang lebih kaya bagi pengambilan keputusan bisnis. Penerapan skema bintang dalam basis data aplikasi streaming musik juga dapat memberikan keunggulan dalam hal pemeliharaan dan skalabilitas sistem. Skema ini memungkinkan penambahan data baru tanpa perlu merombak struktur database secara keseluruhan, sehingga mendukung pertumbuhan data yang pesat dalam industri musik digital. Misalnya, penambahan artis baru, lagu baru, atau perubahan informasi pengguna dapat dilakukan dengan cepat dan efisien. Hal ini penting untuk memastikan bahwa aplikasi tetap responsif dan dapat memenuhi kebutuhan pengguna yang semakin berkembang.

Selain itu, arsitektur skema bintang juga mendukung integrasi dengan teknologi analitik dan kecerdasan buatan (AI). Dengan struktur data yang terorganisir dan terhubung dengan baik, data dari aplikasi streaming musik dapat diolah lebih lanjut untuk analisis prediktif dan personalisasi konten. Misalnya, dengan memanfaatkan machine learning, aplikasi dapat merekomendasikan lagu atau playlist yang sesuai dengan preferensi dan kebiasaan mendengarkan pengguna. Analisis tren dan pola penggunaan juga dapat membantu penyedia layanan untuk mengidentifikasi peluang bisnis baru dan meningkatkan strategi pemasaran.

Dalam jangka panjang, implementasi basis data skema bintang tidak hanya meningkatkan efisiensi operasional tetapi juga memberikan nilai tambah melalui wawasan yang lebih mendalam tentang perilaku pengguna dan performa konten. Hal ini memungkinkan pengambil keputusan di perusahaan untuk mengembangkan layanan yang lebih inovatif dan kompetitif, serta meningkatkan kepuasan dan retensi pengguna. Dengan demikian, skema bintang menjadi fondasi yang kuat untuk pengelolaan data yang efektif dan pemanfaatan informasi yang lebih optimal dalam industri streaming musik. Sebagai hasilnya, pengguna dapat menikmati pengalaman mendengarkan musik yang lebih personal dan dinamis, sementara perusahaan dapat meraih keuntungan melalui pemahaman yang lebih baik terhadap pasar dan tren musik yang sedang berkembang.

# **BAB II**

# **TINJAUAN LITERATUR**

* 1. **Streaming Musik**

Streaming musik adalah metode yang memungkinkan pengguna untuk mendengarkan musik secara langsung melalui internet tanpa perlu mengunduh file audio ke perangkat mereka (Webster, 2020). Dalam streaming musik, data audio dikirim dari server penyedia layanan ke perangkat pengguna dalam potongan kecil yang disebut "chunks," memungkinkan lagu untuk mulai diputar hampir seketika. Proses ini menggunakan teknik buffering untuk memastikan pemutaran musik tetap mulus meskipun ada gangguan dalam koneksi internet. Layanan streaming musik seperti Spotify, Apple Music, dan YouTube Music menawarkan akses ke jutaan lagu dari berbagai genre, artis, dan era. Pengguna dapat menikmati musik kapan saja dan di mana saja selama mereka memiliki koneksi internet, dengan banyak layanan yang juga menawarkan mode offline untuk mendengarkan musik tanpa koneksi internet (Lüders, 2021). Selain itu, layanan streaming ini sering menggunakan algoritma cerdas untuk merekomendasikan musik berdasarkan preferensi dan riwayat mendengarkan pengguna. Meskipun menghadapi tantangan seperti kebutuhan akan koneksi internet yang stabil dan biaya langganan, serta isu terkait distribusi royalti kepada artis, streaming musik telah merevolusi cara orang mengakses dan menikmati musik, membuatnya lebih mudah dan nyaman untuk menikmati koleksi musik yang luas kapan saja dan dimana saja.

* 1. **Business Intelligence (BI) dan Analytics**

Business Intelligence (BI) dan Analytics adalah dua konsep yang saling terkait dalam pengelolaan data untuk membantu organisasi membuat keputusan bisnis yang lebih baik dan strategis (Pabinger & Mayr, 2019). BI melibatkan penggunaan teknologi, aplikasi, dan praktik untuk mengumpulkan, mengintegrasikan, menganalisis, dan menyajikan data bisnis dengan tujuan utama mendukung pengambilan keputusan yang lebih baik. Ini mencakup berbagai alat dan teknik seperti data warehousing, reporting, OLAP (Online Analytical Processing), dan data mining. Di sisi lain, Analytics melibatkan penggunaan data, teknologi, teknik analitis, dan model untuk mendapatkan wawasan mendalam dan memprediksi hasil bisnis (Anardani et al., 2019). Ini termasuk analitik deskriptif untuk memahami data historis, analitik diagnostik untuk mengetahui penyebab suatu kejadian, analitik prediktif untuk memprediksi tren masa depan, dan analitik preskriptif untuk memberikan rekomendasi tindakan. Kedua konsep ini membantu organisasi meningkatkan efisiensi operasional, mengambil keputusan yang lebih baik, memperoleh keunggulan kompetitif, dan mengidentifikasi peluang bisnis baru. Dengan memanfaatkan BI dan Analytics, perusahaan dapat memahami dan memanfaatkan data mereka secara efektif untuk meningkatkan kinerja bisnis dan mencapai tujuan strategis.

* 1. **Structured Query Language (SQL)**

SQL atau Structured Query Language adalah bahasa pemrograman yang dirancang khusus untuk mengelola dan mengolah data dalam sistem basis data relasional (Zhang et al., 2024). SQL digunakan untuk berinteraksi dengan database, memungkinkan pengguna untuk melakukan berbagai operasi seperti mengakses, menyisipkan, memperbarui, dan menghapus data yang tersimpan dalam tabel. Dengan SQL, pengguna dapat mengeksekusi query untuk mengambil informasi spesifik dari database dengan menggunakan sintaks yang sederhana dan intuitif.

Bahasa ini terdiri dari beberapa komponen utama, termasuk Data Query Language (DQL) untuk mengambil data, Data Definition Language (DDL) untuk mendefinisikan struktur database, Data Manipulation Language (DML) untuk mengelola data, dan Data Control Language (DCL) untuk mengatur akses pengguna. Misalnya, perintah SQL seperti `SELECT` digunakan untuk mengambil data dari satu atau lebih tabel, sedangkan perintah `INSERT` digunakan untuk menyisipkan data baru. Selain itu, perintah `UPDATE` digunakan untuk memperbarui data yang sudah ada, dan `DELETE` untuk menghapus data yang tidak diperlukan. SQL juga memungkinkan pengguna untuk mendefinisikan struktur basis data dengan perintah seperti `CREATE` untuk membuat tabel baru, `ALTER` untuk mengubah struktur tabel yang sudah ada, dan `DROP` untuk menghapus tabel atau objek lainnya dari database. Selain itu, SQL mendukung penggabungan tabel (join), yang memungkinkan pengguna untuk mengambil data yang saling terkait dari beberapa tabel sekaligus, serta menyediakan fungsi agregasi seperti `SUM`, `COUNT`, `AVG`, dan lainnya untuk menganalisis data secara lebih mendalam.

Kelebihan utama SQL adalah sifatnya yang standar dan luas digunakan, menjadikannya bahasa yang penting bagi pengembang, analis data, dan profesional TI. Dengan kemampuan untuk bekerja dengan berbagai sistem manajemen basis data seperti MySQL, PostgreSQL, Microsoft SQL Server, dan Oracle Database, SQL menjadi alat yang esensial dalam pengelolaan informasi di berbagai aplikasi dan sistem. Dengan demikian, pemahaman yang baik tentang SQL sangat penting bagi siapa pun yang terlibat dalam pengelolaan data dan pengembangan aplikasi berbasis data.

* 1. **Pentaho**

Pentaho adalah platform business intelligence (BI) yang menyediakan solusi lengkap untuk analisis data, integrasi data, dan visualisasi informasi. Pentaho memungkinkan pengguna untuk mengumpulkan, mengolah, dan menganalisis data dari berbagai sumber, termasuk database, file, dan aplikasi lainnya. Dengan fitur yang komprehensif, Pentaho mencakup berbagai komponen, seperti Pentaho Data Integration (PDI) untuk ekstraksi, transformasi, dan pemuatan (ETL) data, serta Pentaho Business Analytics untuk membuat laporan, dashboard, dan visualisasi data (Hasan & Sudaryana, 2022).

Salah satu keunggulan Pentaho adalah kemampuannya dalam mengelola data secara besar-besaran dan kompleks, sehingga cocok untuk organisasi yang membutuhkan analisis mendalam. Pengguna dapat membuat model data, melakukan analisis ad-hoc, dan menghasilkan laporan yang informatif dengan antarmuka yang ramah pengguna. Pentaho juga mendukung integrasi dengan berbagai sistem dan platform, memungkinkan organisasi untuk memanfaatkan data yang sudah ada secara efektif. Dengan fleksibilitas dan kemampuannya untuk mendukung berbagai kebutuhan BI, Pentaho menjadi pilihan populer di kalangan perusahaan untuk meningkatkan pengambilan keputusan berbasis data.

* 1. **OLTP**

OLTP atau Online Transaction Processing adalah sistem manajemen basis data yang dirancang untuk mendukung transaksi yang cepat dan efisien dalam lingkungan bisnis (Sheng et al., 2019). Sistem ini digunakan untuk mengelola data dalam aplikasi yang membutuhkan pemrosesan transaksi secara real-time, seperti sistem perbankan, pemesanan tiket, dan sistem manajemen inventaris. OLTP fokus pada operasi dengan volume tinggi yang melibatkan penyisipan, pembaruan, dan penghapusan data, serta memastikan integritas dan konsistensi data melalui penggunaan transaksi atomik.

Ciri khas dari OLTP meliputi kemampuan untuk memproses banyak transaksi secara bersamaan (concurrent transactions), waktu respons yang cepat, dan dukungan untuk operasi CRUD (Create, Read, Update, Delete). Selain itu, OLTP biasanya menggunakan basis data relasional yang memiliki struktur tabel terdefinisi dengan baik. Dengan demikian, OLTP sangat penting bagi organisasi yang membutuhkan keandalan dan kecepatan dalam pemrosesan transaksi sehari-hari.

* 1. **OLAP**

OLAP atau Online Analytical Processing adalah teknologi yang digunakan untuk analisis data multidimensional dan mendukung pengambilan keputusan dalam bisnis (Patel & Sharma, 2020). Sistem OLAP memungkinkan pengguna untuk menganalisis dan mengeksplorasi data dari berbagai perspektif dengan cepat, sehingga membantu dalam memahami tren, pola, dan hubungan yang ada dalam data. OLAP biasanya digunakan dalam konteks data warehousing dan business intelligence, di mana data diorganisir dalam struktur yang mendukung analisis yang kompleks.

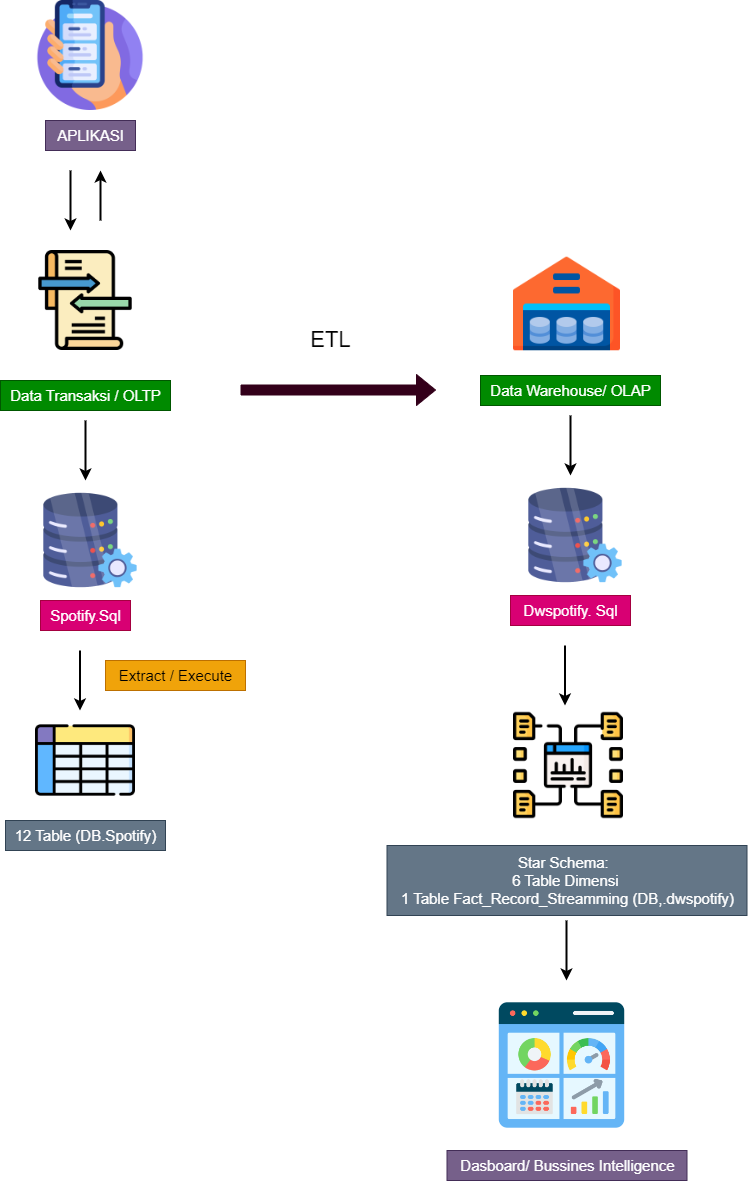
Ciri khas dari OLAP meliputi kemampuan untuk melakukan operasi analitik seperti pengelompokan (grouping), pengolahan agregasi (aggregation), dan pemotongan data (slicing and dicing). Dengan OLAP, pengguna dapat membuat laporan interaktif dan visualisasi data untuk mendalami informasi, melakukan analisis historis, dan mendukung perencanaan strategis. Sistem OLAP biasanya memanfaatkan struktur data berbentuk kubus (OLAP cube) yang memungkinkan akses cepat dan efisien terhadap data yang besar dan kompleks. Dengan demikian, OLAP sangat penting untuk organisasi yang ingin mengambil keputusan berbasis data dengan cepat dan akurat.

# **BAB III**

# **HASIL DAN PEMBAHASAN**

* 1. **Proses ETL dan Arsitektur Data Warehouse**

Gambar 3.1 di bawah menggambarkan proses ETL (Extract, Transform, Load) dan arsitektur data warehouse dalam mengelola data dari aplikasi hingga menjadi informasi yang siap dianalisis. Proses dimulai dari aplikasi di mana pengguna memasukkan data. Data ini kemudian disimpan dalam sistem OLTP (Online Transaction Processing), yang merupakan database transaksi. Langkah berikutnya adalah ekstraksi data dari OLTP menggunakan query SQL (Spotify.sql). Hasil ekstraksi disimpan dalam 12 tabel di database Spotify.

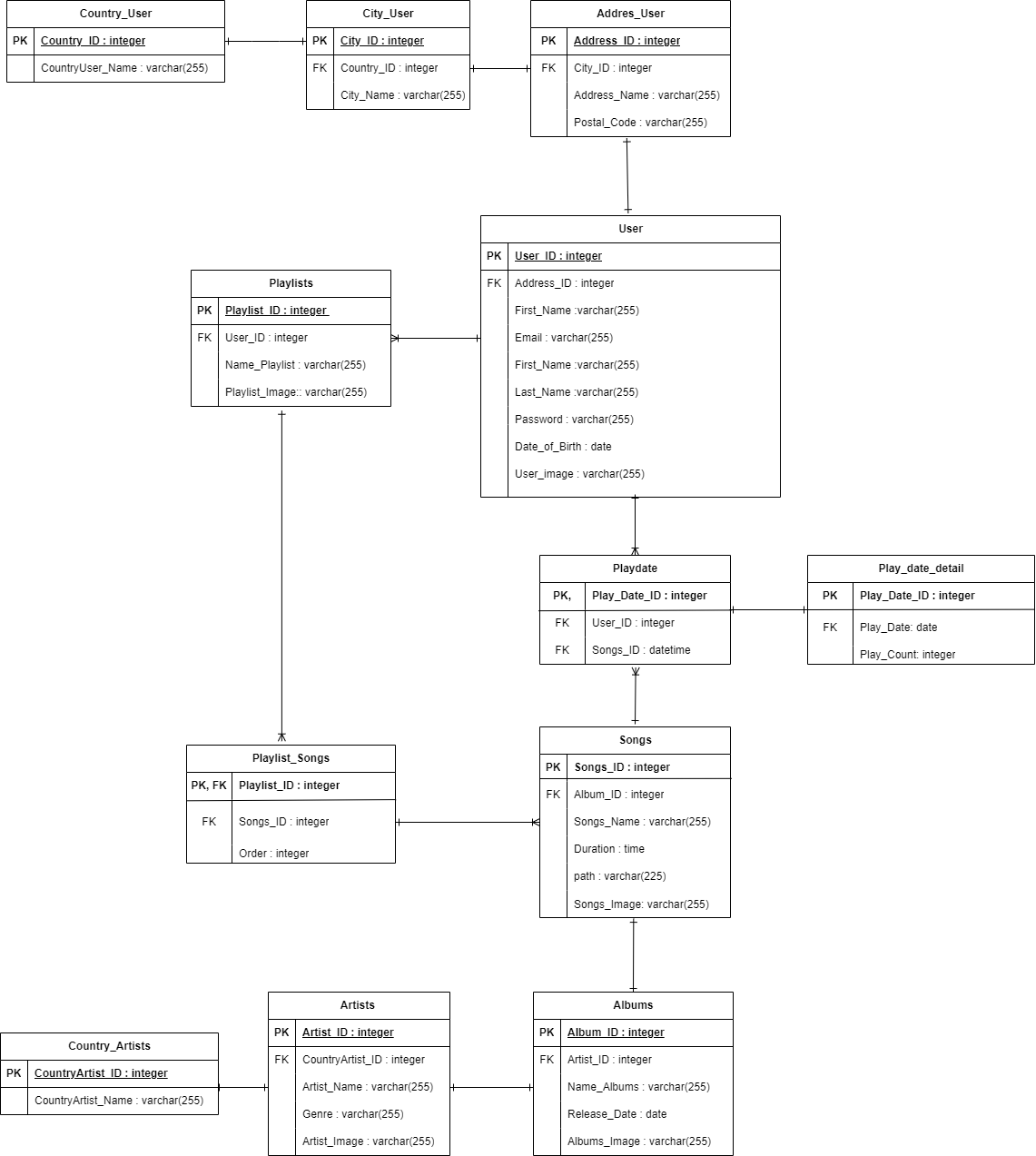


Gambar 3.1 Tahapan Proses ETL.

Selanjutnya, proses ETL dilakukan untuk mentransfer data dari database OLTP ke data warehouse atau OLAP (Online Analytical Processing). Dalam data warehouse, data yang ditransfer disimpan dalam database Dwspotify.sql dan diorganisir menggunakan skema bintang (star schema). Skema ini terdiri dari enam tabel dimensi dan satu tabel fakta yang disebut Fact\_Record\_Streaming. Akhirnya, data yang telah diolah dan disimpan dalam data warehouse digunakan untuk membangun dasbor dan laporan bisnis, memungkinkan analisis yang lebih mendalam dan pengambilan keputusan yang lebih baik. Diagram ini menyoroti pentingnya proses ETL dalam memastikan data yang akurat dan terstruktur tersedia untuk kebutuhan bisnis intelligence.

* 1. **Perancang Data Transaksi**

Aplikasi musik streaming telah mengubah cara kita mendengarkan musik, menawarkan jutaan lagu dan daftar putar yang dipersonalisasi. Untuk memahami cara kerja aplikasi-aplikasi ini, kita bisa melihat Entity relation diagram (ERD). Diagram ini menunjukkan bagaimana berbagai bagian data, seperti pengguna, lagu, dan playlist, terhubung di dalam aplikasi.



Gambar 3.2 ERD Aplikasi Streaming Musik.

Gambar 3.2 merupakan ERD yang merepresentasikan struktur dari aplikasi streaming musik yang menggambarkan bagaimana hubungan antara entitas seperti User, Songs, Playlist dan entitas lainnya saling terhubung.

* + 1. **Entitas**

Entitas dalam konteks basis data dan pemodelan data adalah suatu objek yang memiliki keberadaan yang dapat dibedakan dari objek lain dan yang memiliki atribut tertentu yang mendeskripsikan karakteristiknya. Dalam skema basis data, entitas biasanya direpresentasikan sebagai tabel, di mana setiap baris (record) dalam tabel tersebut mewakili sebuah instans dari entitas tersebut, dan setiap kolom mewakili atribut dari entitas tersebut. Berikut adalah entitas-entitas yang ada di dalam project “Transformasi Pengelolaan Data Streaming Musik Melalui Model Star Schema”:

1. User

Entitas ini berisi informasi dasar tentang pengguna aplikasi. Atributnya meliputi User\_ID (PrimaryKey), Address\_ID (Foreign Key), Nama, dan Email, Password, Tanggal lahir, dan foto profil.

Tabel 3.1 Entitas User dengan Tipe Data.

|  |  |
| --- | --- |
| Users | |
| Atribut | Tipe Data |
| Address\_ID | integer |
| Name | varchar(255) |
| Email | varchar(255) |
| Password | varchar(255) |
| Date\_of\_Birth | date |
| User\_image | varchar(255) |

1. Address\_User

Entitas ini menyimpan informasi alamat pengguna. Atributnya meliputi Address\_ID (Primary Key), City\_ID (Foreign Key), Address\_name, dan Postal\_code.

Tabel 3.2 Entitas Address\_User dengan Tipe Data.

|  |  |
| --- | --- |
| Address\_User | |
| Atribut | Tipe Data |
| City\_ID | integer |
| Address\_Name | varchar(255) |
| Postal\_Code | varchar(255) |

1. City\_User

Entitas ini menangkap informasi kota. Atributnya meliputi City\_ID (PrimaryKey), Country\_ID (Foreign Key), dan City\_Name.

Tabel 3.3 Entitas City\_User dengan Tipe data

|  |  |
| --- | --- |
| City\_User | |
| Atribut | Tipe Data |
| City\_ID | integer |
| Country\_ID | integer |
| City\_Name | varchar(255) |

1. Country\_User

Entitas ini menangkap informasi negara dengan attribut Country\_ID (PrimaryKey), dan Country\_Name.

Tabel 3.4 Entitas Country\_User dengan Tipe data

|  |  |
| --- | --- |
| Country\_User | |
| Atribut | Tipe Data |
| Country\_ID | integer |
| CountryUser\_Name | varchar(255) |

1. Playdate

Entitas ini menangkap informasi dari pemutaran lagu oleh User dengan attribut Play\_Date\_ID (PrimaryKey), User\_ID (Foreign Key), dan Songs\_ID (Foreign Key).

Tabel 3.5 Entitas Playdate dengan Tipe data

|  |  |
| --- | --- |
| Playdate | |
| Atribut | Tipe Data |
| Play\_Date\_ID | integer |
| User\_ID | integer |
| Songs\_ID | datetime |

1. Playdate\_detail

Entitas ini menangkap informasi detail dari setiap pemutaran lagu seperti banyaknya lagu diputar dan waktu dari lagu diputar User dengan attribut Country\_ID (PrimaryKey), dan Country\_Name.

Tabel 3.6 Entitas Playdate\_detail dengan Tipe data

|  |  |
| --- | --- |
| Playdate\_detail | |
| Atribut | Tipe Data |
| Play\_Date\_ID | integer |
| Play\_Date | date |
| Play\_Count | integer |

1. Playlist

Entitas ini menangkap informasi Playlist lagu dengan attribut Playlist\_ID (PrimaryKey), User\_ID (Foreign Key), Nama Playlist, dan foto dari profil Playlist.

Tabel 3.7 Entitas Playlist dengan Tipe data

|  |  |
| --- | --- |
| Playlist | |
| Atribut | Tipe Data |
| Playlist\_ID | integer |
| User\_ID | integer |
| Name\_Playlist | varchar(255) |
| Playlist\_Image | varchar(255) |

1. Playlist\_Songs

Entitas ini menangkap informasi detail dari Playlist seperti lagu beserta urutannya dengan attribut Playlist(PrimaryKey dan ForeignKey), Songs\_ID (Foreign Key) dan Order.

Tabel 3.8 Entitas Playlist\_Songs dengan Tipe data

|  |  |
| --- | --- |
| Playlist\_Songs | |
| Atribut | Tipe Data |
| Playlist\_ID | integer |
| Order | integer |

1. Songs

Entitas ini menangkap informasi dari lagu seperti judul lagu, durasi lagu serta gambar yang dengan attribut Songs\_ID, Album\_ID, Songs\_Name, Duration, path, dan Songs\_Image.

Tabel 3.9 Entitas Songs dengan Tipe data

|  |  |
| --- | --- |
| Songs | |
| Atribut | Tipe Data |
| Songs\_ID | integer |
| Album\_ID | integer |
| Songs\_Name | varhar(255) |
| Duration | time |
| path | varhar(255) |
| Songs\_Image | varhar(255) |

1. Albums

Entitas ini menangkap informasi Albums dengan attribut Album\_ID, Artist\_ID, Name\_Albums, Release\_Date, dan Albums\_Image\

Tabel 3.10 Entitas Albums dengan Tipe data

|  |  |
| --- | --- |
| Albums | |
| Atribut | Tipe Data |
| Album\_ID | integer |
| Artist\_ID | integer |
| Name\_Albums | varchar(255) |
| Release\_Date | date |
| Albums\_Image | varchar(255) |

1. Artist

Entitas ini menangkap informasi Artist dengan attribut Atribut, Artist\_ID, CountryArtist\_ID, Artist\_Name, Genre, dan Artist\_Image

Tabel 3.11 Entitas Artist dengan Tipe data

|  |  |
| --- | --- |
| Artist | |
| Atribut | Tipe Data |
| Artist\_ID | integer |
| CountryArtist\_ID | integer |
| Artist\_Name | varchar(255) |
| Genre | varchar(255) |
| Artist\_Image | varchar(255) |

1. Country\_Artist

Entitas ini menangkap informasi Country\_Artist dengan attribut CountryArtist\_ID, dan CountryArtist\_Name

Tabel 3.12 Entitas Country\_Artist dengan Tipe data

|  |  |
| --- | --- |
| Country\_Artist | |
| Atribut | Tipe Data |
| CountryArtist\_ID | integer |
| CountryArtist\_Name | varchar(255) |

* + 1. **Relasi**

Relasi adalah hubungan antara dua atau lebih entitas. Relasi menggambarkan bagaimana entitas-entitas tersebut berinteraksi satu sama lain. Dalam model relasional, relasi biasanya diimplementasikan sebagai tabel dalam basis data, dengan kunci asing (foreign key) yang digunakan untuk menghubungkan tabel-tabel yang berelasi. Berikut adalah relasi-relasi yang ada pada project “Transformasi Pengelolaan Data Streaming Musik Melalui Model Star Schema”:

1. User ke Address\_User

Relasi One to One. 1 User bisa memiliki 1 alamat.

1. Address\_User ke City\_User

Relasi One to One. 1 alamat bisa memiliki 1 kota.

1. City\_User ke Country\_User

Relasi One to One. 1 kota bisa memiliki 1 negara.

1. User ke Playdate

Relasi One to Many. 1 User bisa memiliki banyak Playdate.

1. Playdate ke Play\_date\_detail

Relasi One to One. 1 Playdate bisa memiliki 1 Play\_date\_detail.

1. User ke Playlist

Relasi One to Many. 1 User bisa memiliki banyak Playlist.

1. Playlist ke Playlist\_Songs

Relasi One to Many. 1 Playlist bisa memiliki banyak Playlist\_Songs.

1. Playlist\_Songs ke Songs

Relasi One to Many. 1 Playlist\_Songs bisa memiliki banyak Songs.

1. Songs ke Playdate

Relasi One to Many. 1 Songs bisa memiliki banyak Playdate.

1. Songs ke Albums

Relasi One to One. 1 Songs bisa memiliki 1 Albums.

1. Albums ke Artist

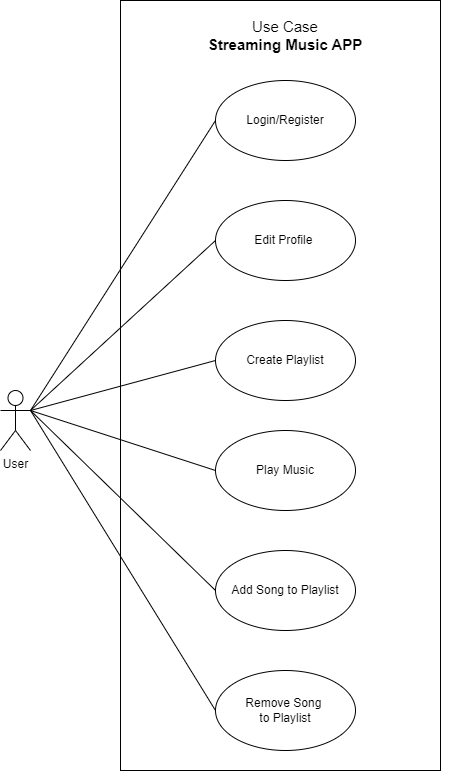
Relasi One to One. 1 Albums bisa memiliki 1 Artist.

1. Artist ke Country\_Artist

Relasi One to One. 1 Artistbisa memiliki 1 Country\_Artist.

* + 1. **Use Case**

Dalam gambar 3.3 yang ada di bawah ini, untuk memahami proses bisnis yang terjadi, kita dapat melihat terdapat enam proses bisnis utama dalam use case tersebut. Proses-proses bisnis ini meliputi:



Gambar 3.3 Diagram Proses Bisnis

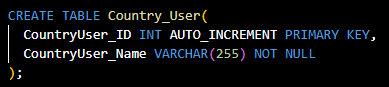
1. Login/Register: Proses ini memungkinkan user untuk membuat akun baru atau masuk ke akun yang sudah didaftarkan sebelumnya. Ini adalah langkah pertama yang harus dilakukan user untuk dapat mengakses layanan di platfrom music ini
2. Edit Profile: user dapat memperbarui informasi profil mereka, seperti nama, alamat, email, password, tanggal lahir, dan preferensi lainnya. Proses ini memungkinkan pengguna untuk menjaga informasi mereka tetap up-to-date dan sesuai dengan identitas mereka.
3. Create Playlist: Setelah user berhasil melakukan login atau registrasi, user dapat membuat playlist baru. Proses ini melibatkan penamaan playlist.
4. Play Music: user dapat memutar musik yang ada di dalam platform. Proses ini mencakup pemilihan lagu yang ingin diputar.
5. Add Song to Playlist: Proses ini memungkinkan user untuk menambahkan lagu ke dalam playlist yang sudah dibuat sebelumnya. user dapat memilih beberapa lagu dari list dan menambahkannya ke playlist tertentu sesuai keinginan mereka.
6. Remove Song from Playlist: user juga diberikan kebebasan untuk menghapus lagu dari playlist mereka. Proses ini mencakup pemilihan lagu yang ingin dihapus dan konfirmasi penghapusan dari playlist yang dipilih.

Setiap proses bisnis tersebut berinteraksi satu sama lain untuk memberikan pengalaman pengguna yang komprehensif dan terpadu di platform musik ini. Proses login/register sebagai langkah awal yang membuka akses ke fitur-fitur lainnya, sedangkan proses lainnya berfokus pada pengelolaan dan pemutaran musik, yang merupakan inti dari layanan yang ditawarkan.

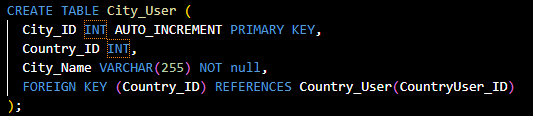
* 1. **Script SQL Data Transaksi**

Setelah membuat Entity-Relationship Diagram (ERD), langkah selanjutnya adalah mengubah diagram tersebut ke dalam skrip SQL untuk membuat tabel-tabel yang sesuai di MySQL. Proses ini melibatkan pendefinisian struktur setiap tabel berdasarkan entitas dan hubungan yang diidentifikasi dalam ERD. Skrip SQL akan menentukan nama tabel, kolom, tipe data, primary key, foreign key, dan constrain yang diperlukan untuk menjaga integritas data dan membangun hubungan antar tabel. Berikut adalah script SQL data transaksi pada project “Transformasi Pengelolaan Data Streaming Musik Melalui Model Star Schema”:

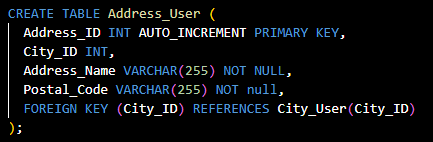
1. Buat tabel `Country\_User` untuk menyimpan informasi tentang negara pengguna.



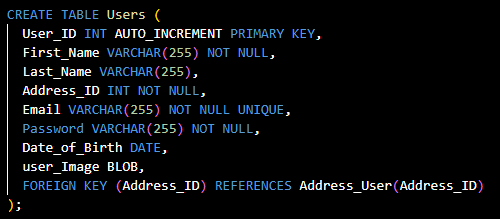
1. Buat tabel `City\_User` untuk menyimpan informasi tentang kota pengguna, termasuk referensi foreign key ke tabel `Country\_User`.



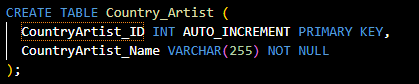
1. Buat tabel `Alamat\_Pengguna` untuk menyimpan alamat pengguna, termasuk referensi foreign key ke tabel `Kota\_Pengguna`.



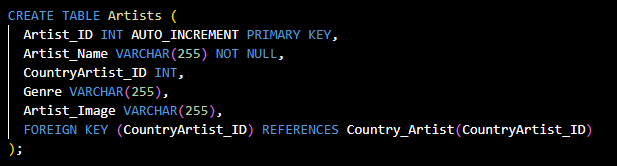
1. Buat tabel `Users` untuk menyimpan detail pengguna, termasuk referensi foreign key ke tabel `Address\_User`.



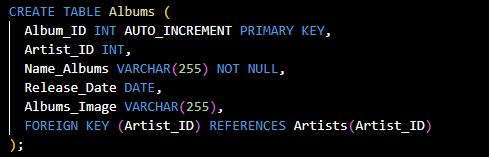
1. Buat tabel `Country\_Artist` untuk menyimpan informasi tentang negara artis.



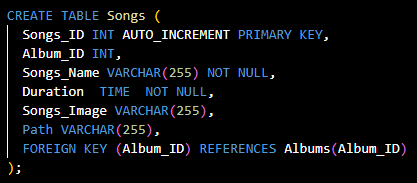
1. Buat tabel `Artists` untuk menyimpan rincian artis, termasuk referensi foreign key ke tabel `Country\_Artist`.



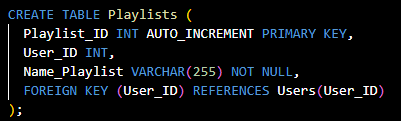
1. Buat tabel `Albums` untuk menyimpan detail album, termasuk referensi foreign key ke tabel `Artists`.



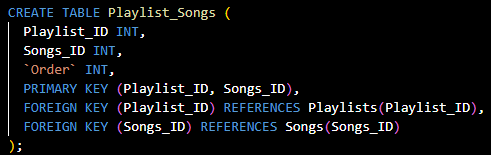
1. Buat tabel `Songs` untuk menyimpan rincian lagu, termasuk referensi foreign key ke tabel `Albums`.



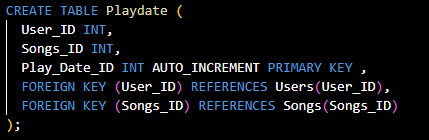
1. Buat tabel `Playlists` untuk menyimpan rincian daftar putar, termasuk referensi foreign key ke tabel `Users`.



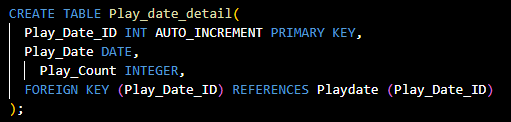
1. Buat tabel `Playlist\_Songs` untuk menautkan lagu ke daftar putar, termasuk referensi foreign key ke tabel `Playlist` dan `Songs`.



1. Buat tabel `Playdate` untuk melacak kapan pengguna memutar lagu, termasuk referensi foreign key ke tabel `Pengguna` dan `Lagu`.



1. Buat tabel `Play\_date\_detail` untuk menyimpan detail tentang tanggal pemutaran, termasuk referensi foreign key ke tabel `Playdate`.



* + 1. **Contoh Data untuk Setiap Tabel**

Contoh data untuk setiap tabel yang digunakan dalam sistem ini. Penyajian contoh data ini sangat penting untuk memberikan gambaran yang jelas tentang bagaimana struktur dan organisasi data dalam sistem berfungsi. Setiap tabel yang akan dijelaskan memiliki kolom-kolom yang berbeda, dan contoh data yang disediakan bertujuan untuk mengilustrasikan jenis informasi yang akan disimpan serta format yang digunakan. Berikut adalah contoh-contoh dari masing-masing tabel:

* + 1. Country\_User

Country User menyimpan informasi tentang pengguna berdasarkan negara, dengan kolom CountryUser\_ID sebagai pengidentifikasi unik dan CountryUser\_Name untuk nama negara. Tabel 3.13 adalah tabel untuk memberitahukan contoh data untuk tabel Country\_User.

Tabel 3.13 Contoh Data Tabel Country\_User.

|  |  |
| --- | --- |
| CountryUser\_ID | CountryUser\_Name |
| 1 | Indonesia |

* + 1. City\_User

City User adalah tabel yang berisi data pengguna berdasarkan kota. Tabel ini memiliki kolom City\_ID sebagai kunci utama, Country\_ID yang mungkin merujuk ke tabel Country User, dan City\_Name untuk nama kota. Tabel 3.14 adalah tabel untuk memberitahukan contoh data untuk tabel City\_User.

Tabel 3.14 Contoh Data Tabel City\_User.

|  |  |  |
| --- | --- | --- |
| City\_ID | Country\_ID | City\_Name |
| 1 | 1 | Jakarta |

* + 1. Address\_User

Address User merupakan tabel yang menyimpan informasi detail alamat pengguna. Terdiri dari kolom Address\_ID sebagai identifikasi unik, City\_ID yang kemungkinan terhubung dengan tabel City User, dan Address\_Name. Tabel 3.15 adalah tabel untuk memberitahukan contoh data untuk tabel Address\_User.

Tabel 3.15 Contoh Data Tabel Address\_User

|  |  |  |
| --- | --- | --- |
| Address\_ID | City\_ID | Address\_Name |
| 1 | 1 | Jl. Sudirman No. 1 |

* + 1. Users

Tabel Users adalah pusat penyimpanan informasi pengguna. Kolom-kolomnya meliputi User\_ID, First\_Name, Last\_Name, Address\_ID, Email, Password, Date\_of\_Birth, dan user\_Image. Tabel ini menyimpan data pribadi dan kredensial pengguna yang diperlukan untuk akun mereka. Tabel 3.16 adalah tabel untuk memberitahukan contoh data untuk tabel Users.

Tabel 3.16 Contoh Data Tabel Users.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| User\_ID | First\_Name | Last\_Name | Address\_ID | Email | Password | Date\_of\_Birth | user\_Image |
| 1 | Mamat | Sujito | 1 | Mamat.sujito@example.com | password123 | 1990-01-01 | https://static.cdntap.com/tap-assets-prod/wp-content/uploads/sites/24/2021/08/aktor-berdarah-minang-10-wowkeren.jpg.jpg |

* + 1. Country\_Artist

Country Artist adalah tabel yang mencatat informasi artis berdasarkan negara. Tabel ini memiliki kolom CountryArtist\_ID sebagai identifikasi unik dan CountryArtist\_Name untuk nama negara asal artis. Tabel 3.17 adalah tabel untuk memberitahukan contoh data untuk tabel Country\_Artist.

Tabel 3.17 Contoh Data Tabel Country\_Artist.

|  |  |
| --- | --- |
| CountryArtist\_ID | CountryArtist\_Name |
| 1 | United Kingdom |

* + 1. Artists

Tabel Artists menyimpan data lengkap tentang para artis. Kolomnya terdiri dari Artist\_ID, Artist\_Name, CountryArtist\_ID, Genre, dan Artist\_Image. Tabel 3.18 adalah tabel untuk memberitahukan contoh data untuk tabel Artists.

Tabel 3.18 Contoh Data Tabel Artists.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artist\_ID | Artist\_Name | CountryArtist\_ID | Genre | Artist\_Image |
| 1 | The Beatles | 1 | Rock | https://cdn.britannica.com/18/136518-050-CD0E49C6/The-Beatles-Ringo-Starr-Paul-McCartney-George.jpg |

* + 1. Albums

Albums adalah tabel yang berisi informasi tentang album musik. Tabel ini mencakup kolom Album\_ID sebagai identifikasi unik, Artist\_ID yang mungkin terhubung ke tabel Artists, Name\_Albums untuk judul album, Release\_Date untuk tanggal rilis, dan Albums\_Image untuk gambar sampul album. Tabel 3.19 adalah tabel untuk memberitahukan contoh data untuk tabel Albums.

Tabel 3.19 Contoh Data Tabel Albums.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Album\_ID | Artist\_ID | Name\_Albums | Release\_Date | Albums\_Image |
| 1 | 1 | Abbey Road | 1969-09-26 | https://upload.wikimedia.org/wikipedia/id/4/42/Beatles\_-\_Abbey\_Road.jpg |

* + 1. Songs

Tabel Songs menyimpan informasi detail tentang lagu-lagu. Kolomnya meliputi Songs\_ID sebagai identifikasi unik, Album\_ID yang mungkin terhubung ke tabel Albums, Songs\_Name untuk judul lagu, Duration untuk panjang lagu, Songs\_Image untuk gambar terkait lagu, dan Path yang mungkin merujuk pada lokasi file lagu dalam sistem. Tabel 3.20 adalah tabel untuk memberitahukan contoh data untuk tabel Songs.

Tabel 3.20 Contoh Data Tabel Songs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Songs\_ID | Album\_ID | Songs\_Name | Duration | Songs\_Image | Path |
| 1 | 1 | Come Together | 2:29 | https://www.beatlesbible.com/wp/media/italy\_something.jpg | come\_together.mp3 |

* + 1. Playlists

Tabel Playlists berisi informasi tentang daftar putar yang dibuat pengguna. Terdiri dari kolom Playlist\_ID sebagai kunci utama, User\_ID yang kemungkinan terhubung ke tabel Users, dan Name\_Playlist untuk nama daftar putar. Tabel 3.21 adalah tabel untuk memberitahukan contoh data untuk tabel Playlists.

Tabel 3.21 Contoh Data Tabel Playlists.

|  |  |  |
| --- | --- | --- |
| Playlist\_ID | User\_ID | Name\_Playlist |
| 1 | 1 | Rock Classics |

* + 1. Playlist\_Songs

Playlist\_Songs adalah tabel penghubung antara Playlists dan Songs. Kolom-kolomnya adalah Playlist\_ID, Songs\_ID, dan Order. Tabel 3.22 adalah tabel untuk memberitahukan contoh data untuk tabel Playlist\_Songs.

Tabel 3.22 Contoh Data Tabel Playlist\_Songs.

|  |  |  |
| --- | --- | --- |
| Playlist\_ID | Songs\_ID | Order |
| 1 | 1 | 1 |

* + 1. Playdate

Tabel Playdate kemungkinan digunakan untuk melacak riwayat pemutaran lagu oleh pengguna. Kolom-kolomnya meliputi User\_ID, Songs\_ID, dan Play\_Date\_ID. Tabel 3.23 adalah tabel untuk memberitahukan contoh data untuk tabel Playdate.

Tabel 3.23 Contoh Data Tabel Playdate.

|  |  |  |
| --- | --- | --- |
| User\_ID | Songs\_ID | Play\_Date\_ID |
| 2 | 1 | 1 |

* + 1. Play\_date\_detail

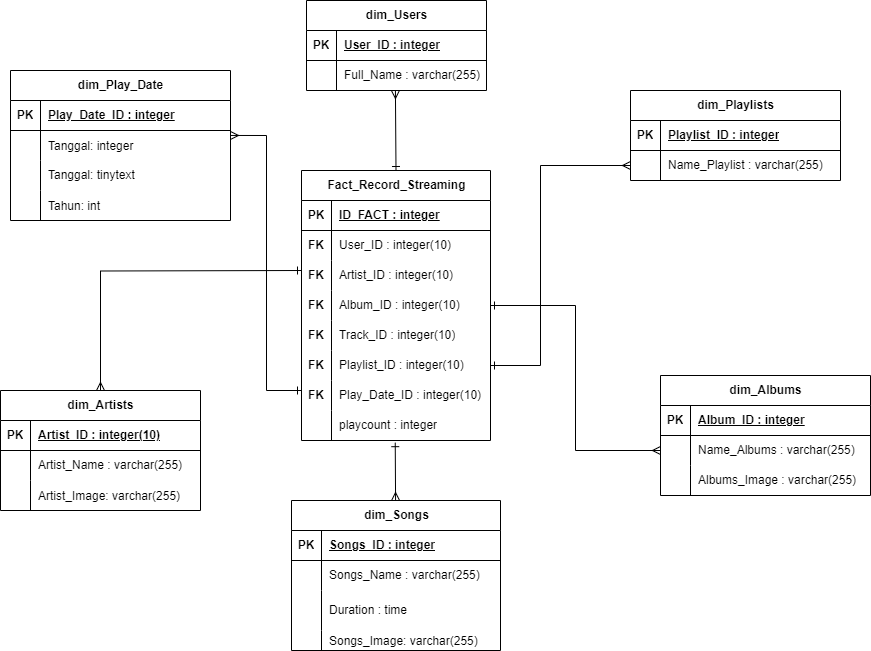
Play\_date\_detail adalah tabel yang menyimpan informasi lebih lanjut tentang pemutaran lagu. Terdiri dari kolom Play\_Date\_ID, Play\_Date untuk tanggal dan waktu pemutaran, dan Play\_Count yang mungkin menghitung berapa kali lagu diputar. Tabel ini memungkinkan analisis lebih mendalam tentang kebiasaan mendengarkan musik pengguna. Tabel 3.24 adalah tabel untuk memberitahukan contoh data untuk tabel Play\_date\_detail.

Tabel 3.24 Contoh Data Tabel Play\_date\_detail.

|  |  |  |
| --- | --- | --- |
| Play\_Date\_ID | Play\_Date | Play\_Count |
| 1 | 2023-01-01 | 2 |

* 1. **Perancangan Star Schema Data Warehouse**

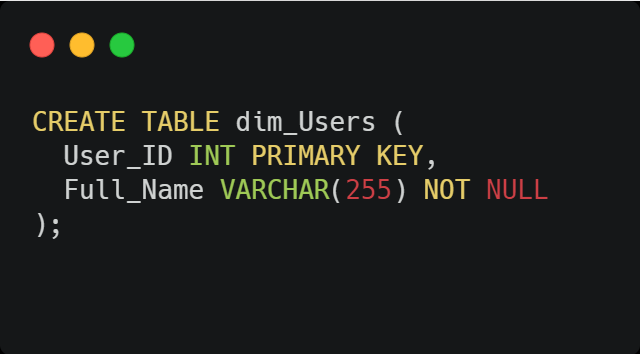
Setelah melakukan perancang sebuah database bernama db.spotify yang berisi 12 table. Maka selanjutnya adalah **peracangan Star Schema**. Dalam proses ini, kami menyusun skema bintang (star schema) yang terdiri dari 6 tabel dimensi dan 1 tabel fact.  Tabel dimensi adalah salah satu komponen utama dalam model data dimensional, yang sering digunakan dalam data warehousing dan analisis bisnis. Tabel ini berisi atribut deskriptif yang memberikan konteks pada data yang disimpan dalam tabel fact.  Sedangkan tabel fact adalah Tabel fact adalah jenis tabel dalam data warehousing dan model data dimensional yang berisi data kuantitatif yang dapat diukur dan dianalisis. Tabel ini merupakan inti dari analisis bisnis karena mencatat transaksi atau peristiwa yang terjadi dalam bisnis. Berikut merupakan gamba dari star schema yang dirancang untuk mempermudah analisis dan pelaporan data.



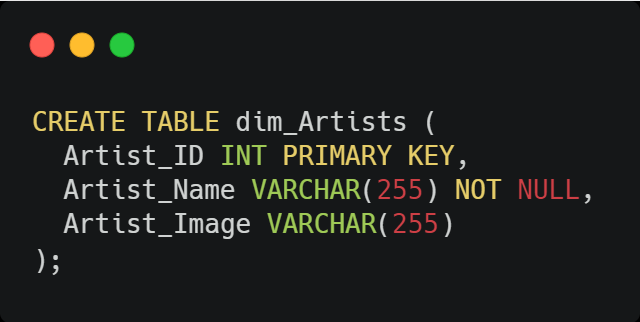
Gambar 3.4 Diagram Star Schema

Tabel dimensi adalah elemen penting dalam desain skema bintang karena mereka menyediakan konteks yang diperlukan untuk memahami data dalam tabel fact. Berikut adalah penjelasan dari masing-masing tabel dimensi dalam diagram Star Schema yang telah dirancang:

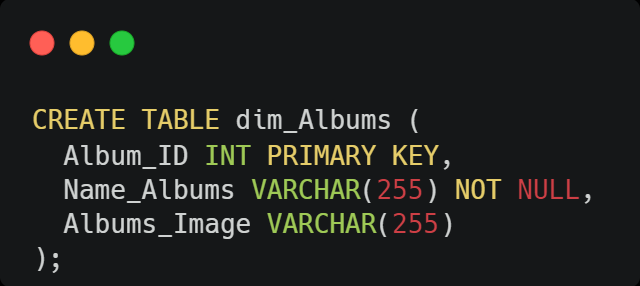
1. **dim\_Users:** Menyimpan informasi detail tentang pengguna.



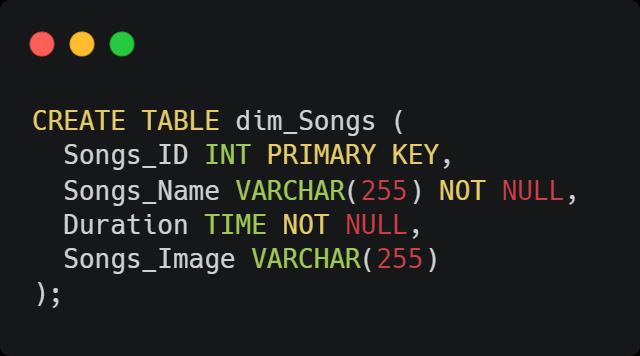
1. **dim\_Artist:** Menyimpan informasi detail tentang artis.



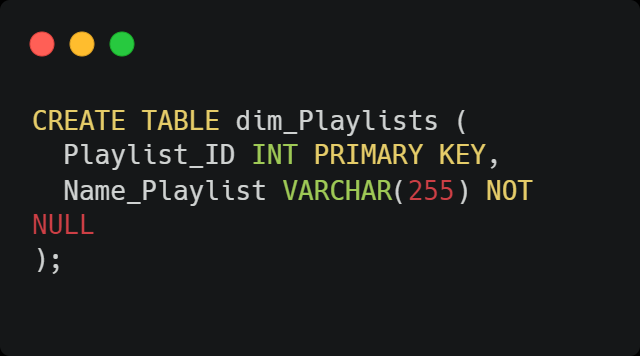
1. **dim\_Albums:** Menyimpan informasi detail tentang album.



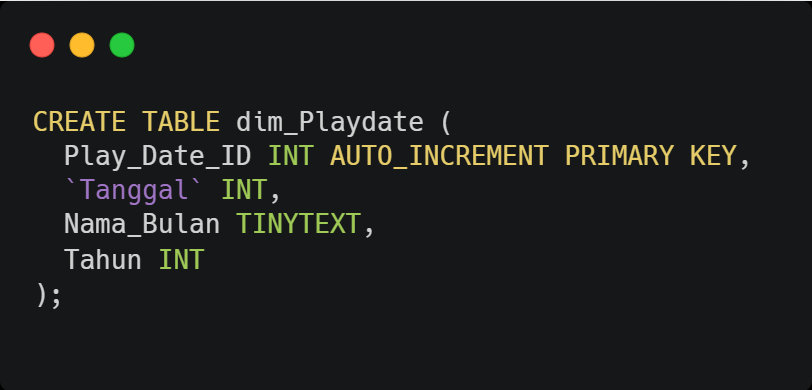
1. **dim\_Songs:** Menyimpan informasi detail tentang lagu.



1. **dim\_Playlists:** Menyimpan informasi detail playlist lagu.



1. **dim\_Playdate:** Menyimpan informasi tanggal pemutaran musik baik dalam tanngal, bulan dan tahun.

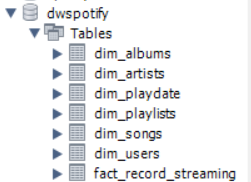


Tabel faktual adalah pusat dari skema bintang dan berisi data kuantitatif yang dapat diukur serta dianalisis. Tabel ini mencatat transaksi atau peristiwa yang terjadi dalam bisnis dan menghubungkannya dengan tabel dimensi yang relevan. Berikut adalah penjelasan tentang tabel faktual dalam diagram Star Schema yang dirancang:

1. **Tabel Fact\_Record\_Streaming:** Tabel yang digunakan untuk merekam data streaming lagu. Tabel ini dirancang untuk memungkinkan analisis data yang terkait dengan perilaku pengguna terhadap lagu-lagu dalam berbagai konteks seperti artis, album, lagu, dan playlist yang berbeda.



Untuk membangun sebuah data warehouse yang optimal, kami merancang skema bintang (star schema) yang terdiri dari enam tabel dimensi dan satu tabel fact. Skema ini akan diimplementasikan dalam sebuah database bernama “dwspotif”. Gambar 3.5 merupakan gambar dari database yang bernama “dwspotif”.



Gambar 3.5 Database dwspotify.

* 1. **Edit Transform Load (ETL) Menggunakan Pentaho**

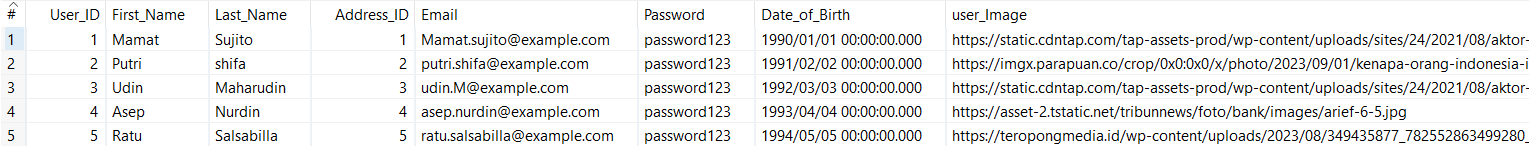
Setelah mempunyai ERD data transaksi dan ERD star schema, langkah selanjutnya adalah melakukan proses Extract, Transform, Load (ETL) menggunakan Pentaho Data Integration (PDI). Pentaho menyediakan alat yang kuat untuk mengelola proses ETL secara efisien. Pertama, data mentah dari berbagai sumber akan diekstraksi. Sumber data ini bisa berupa basis data relasional, file CSV, file Excel, atau bahkan API. Setelah data diekstraksi, langkah berikutnya adalah transformasi data. Transformasi ini meliputi pembersihan data, penggabungan data dari beberapa tabel, penghitungan nilai turunan, dan konversi tipe data sesuai kebutuhan. Di sini, peran PDI sangat penting karena menyediakan berbagai langkah dan alat bantu untuk mempermudah transformasi data. Setelah data ditransformasi sesuai dengan kebutuhan skema bintang, data kemudian dimuat ke dalam data warehouse. Proses pemuatan ini memastikan bahwa data tersedia dan siap untuk digunakan dalam analisis bisnis. Dengan demikian, ETL menggunakan Pentaho tidak hanya mempercepat proses pengolahan data tetapi juga meningkatkan akurasi dan konsistensi data yang dihasilkan.

* + 1. **Pembuatan Tabel Dimensi**

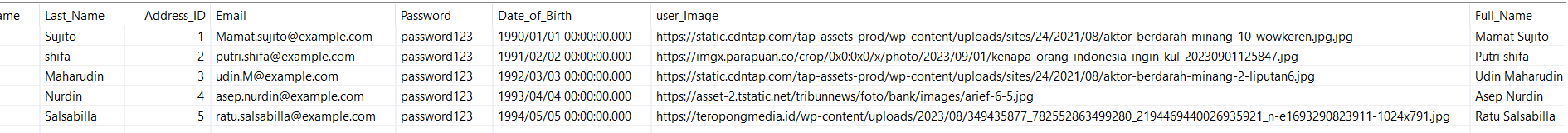
1. **A diagram of a cell phone

   Description automatically generated**Dimensi User

* Input User

Input User dilakukan dengan menggunakan Input table dengan mengkonesikan database ke pentaho dan mengambil data dengan menggunakan script SQL.

* Concat Name

A screenshot of a computer

Description automatically generatedTahapan ini menggabungkan atribut First\_Name dan Last\_Name menjadi atribut baru Full\_Name dengan menggunakan Concat fields.

* Select values

A screenshot of a computer

Description automatically generatedMemilih field yang digunakan sesuai dengan star schema yaitu User\_ID dan Full\_Name.

* Output (dim\_user)

A screenshot of a computer

Description automatically generatedOutput dari hasil transformasi disimpan pada database dwspotify dengan menggunakan table output.

1. Dimensi Albums

A diagram of a list

Description automatically generated with medium confidence

* Input Album

Input Album dilakukan dengan menggunakan Input table dengan mengkonesikan database ke pentaho dan mengambil data dengan menggunakan script SQL.

A screenshot of a computer

Description automatically generated

* Select values

Memilih field yang digunakan sesuai dengan star schema yaitu Album\_ID, Name\_Albums, dan Albums\_Image

A screenshot of a computer

Description automatically generated

* Output (dim\_album)

Output dari hasil transformasi disimpan pada database dwspotify dengan menggunakan table output.

A close-up of a text

Description automatically generated

1. Dimensi Artists

A diagram of a select values

Description automatically generated with medium confidence

* Input Artist

Input Artist dilakukan dengan menggunakan Input table dengan mengkonesikan database ke pentaho dan mengambil data dengan menggunakan script SQL.

A close-up of a text

Description automatically generated

* Select values

Memilih field yang digunakan sesuai dengan star schema yaitu Artist\_ID, Artist\_Name, Artist\_Image.

A screenshot of a computer

Description automatically generated

* Output (dim\_artist)

Output dari hasil transformasi disimpan pada database dwspotify dengan menggunakan table output.

A close-up of a text

Description automatically generated

1. A diagram of a workflow

   Description automatically generatedDimensi Playdate

* Input Playdate

Input Playdate dilakukan dengan menggunakan Input table dengan mengkonesikan database ke pentaho dan mengambil data dengan menggunakan script SQL.

A screenshot of a computer

Description automatically generated

* Select values

A screenshot of a computer

Description automatically generatedMemilih field yang dibutuhkan dalam tabel playdate yaitu Play\_Date\_ID dan Play\_Date

* Konversi waktu

A screenshot of a computer

Description automatically generatedMengubah Date menjadi atribut Tanggal, bulan dan tahun menggunakan calculator.

* Nama\_Bulan

Mengubah Bulan yang dituliskan dengan integer menjadi nama bulan dengan string seperti januari, februari dan lain-lain.

A screenshot of a calendar

Description automatically generated

* Stream Lookup

A table with numbers and numbers

Description automatically generatedMengabungkan nama bulan dengan tanggal yang sesuai menggunakan stream lookup.

* Select Values 2

A screenshot of a computer

Description automatically generatedMemilih field yang digunakan sesuai dengan star schema yaitu Play\_Date\_ID, Tanggal, Nama\_Bulan, Tahun.

* Output (dim\_playdate)

Output dari hasil transformasi disimpan pada database dwspotify dengan menggunakan table output.

A screenshot of a calendar

Description automatically generated

1. Dimensi Playlist

A diagram of a select values

Description automatically generated with medium confidence

* Input Playlist

A screenshot of a music list

Description automatically generatedInput Playlist dilakukan dengan menggunakan Input table dengan mengkonesikan database ke pentaho dan mengambil data dengan menggunakan script SQL.

* Select values

Memilih field yang digunakan sesuai dengan star schema yaitu Playlist\_ID dan Name\_Playlist.

A screenshot of a computer

Description automatically generated

* Output (dim\_playlist)

Output dari hasil transformasi disimpan pada database dwspotify dengan menggunakan table output.

A screenshot of a music list

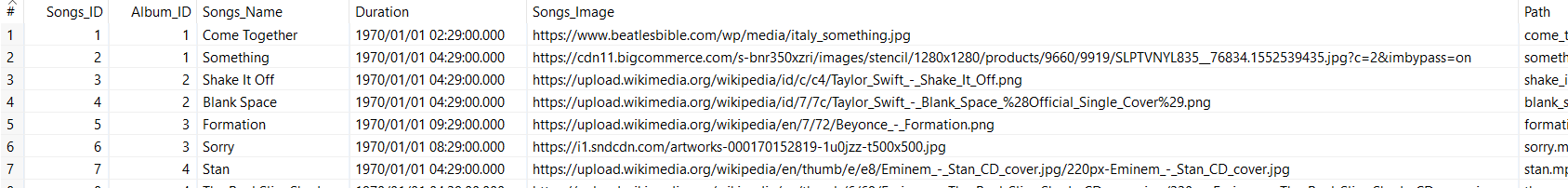
Description automatically generated

1. A diagram of a list

   Description automatically generated with medium confidenceDimensi Songs

* Input Songs

Input Playlist dilakukan dengan menggunakan Input table dengan mengkonesikan database ke pentaho dan mengambil data dengan menggunakan script SQL.



* Select values

A screenshot of a computer

Description automatically generatedMemilih field yang digunakan sesuai dengan star schema yaitu Songs\_ID, Songs\_Namw, Duration, Songs\_Image.

* Output (dim\_Songs)

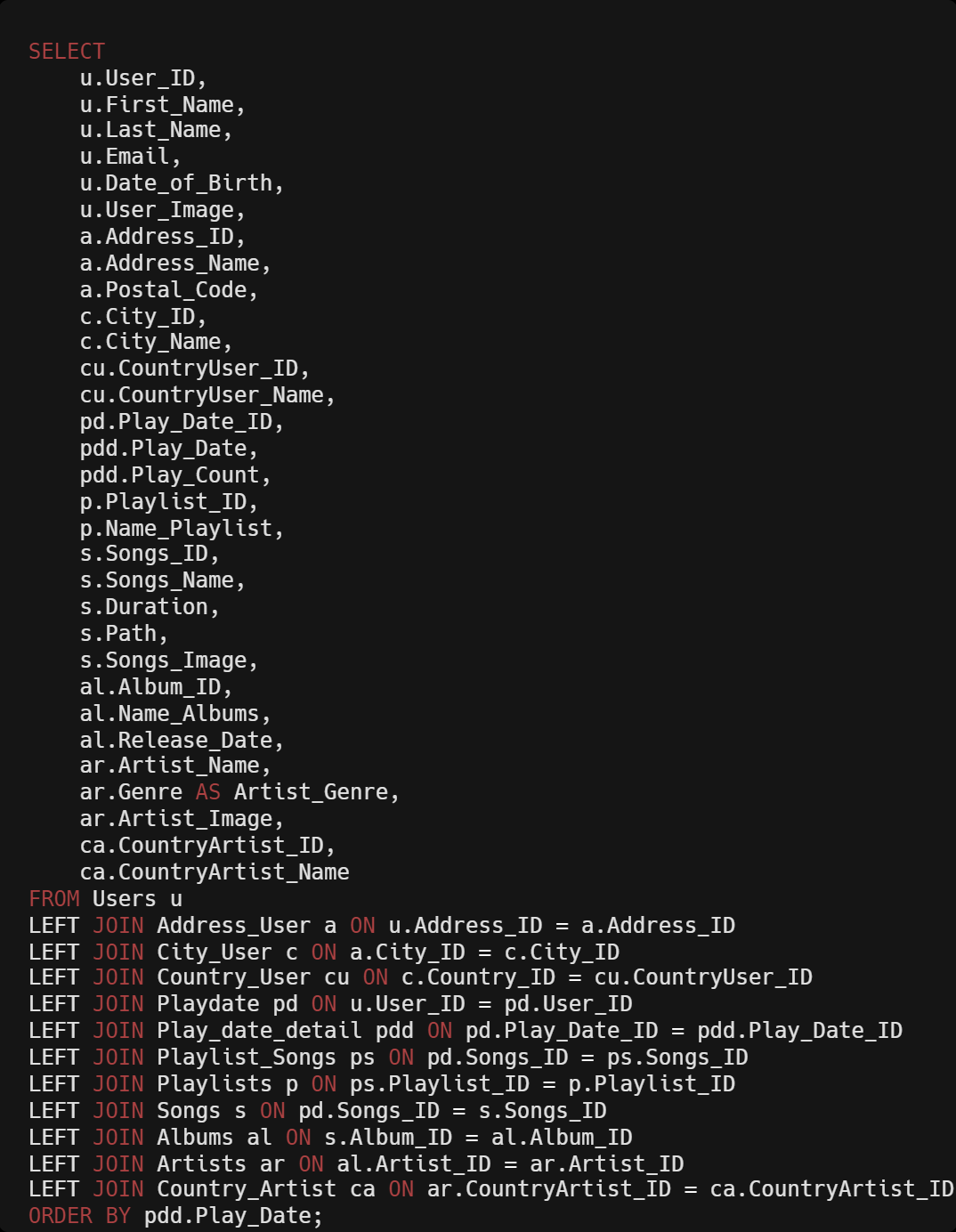
Output dari hasil transformasi disimpan pada database dwspotify dengan menggunakan table output.

A screenshot of a computer

Description automatically generated

* + 1. **Pembuatan Tabel Transaksi**

Gambar 3.6 merupakan Kueri SQL yang dirancang untuk mengambil data transaksi yang komprehensif dengan menggabungkan beberapa tabel terkait dalam database. Kueri ini mengekstrak informasi rinci tentang pengguna, alamat, tanggal pemutaran, daftar putar, lagu, album, dan artis terkait.



Gambar 3.6 Query SQL Tabel Transaksi.

Dengan melakukan serangkaian left join, kueri memastikan bahwa semua data yang relevan disertakan, bahkan jika beberapa bidang bernilai nol, sehingga memberikan tampilan lengkap untuk setiap transaksi. Kumpulan data akhir diurutkan berdasarkan tanggal pemutaran, memberikan urutan kronologis pada transaksi. Pendekatan terstruktur ini memungkinkan analisis menyeluruh terhadap aktivitas dan interaksi pengguna, menangkap konteks lengkap dari setiap tanggal pemutaran dan konten musik terkait.

* + - 1. **Implementasi Query pada Pentaho**

Untuk mereplikasi kueri SQL yang diberikan dalam Pentaho Data Integration (PDI), beberapa langkah dilakukan:

* Table Input

Konfigurasikan Table Input untuk setiap tabel yang terlibat: Users, Address\_User, City\_User, Country\_User, Playdate, Play\_date\_detail, Playlist\_Songs, Playlists, Songs, Albums, Artists, dan Country\_Artist.

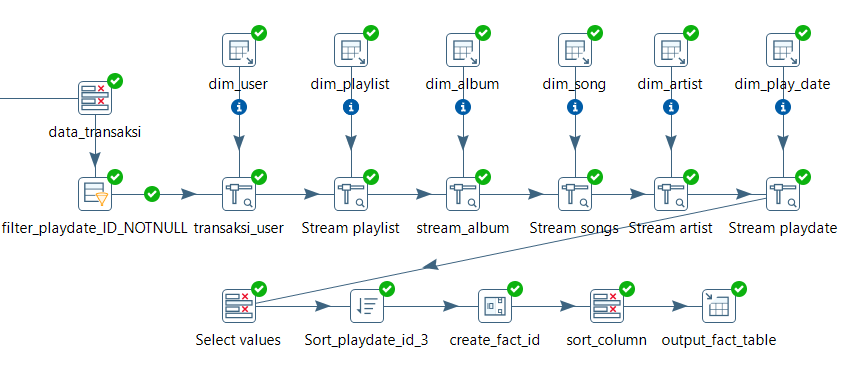
* Sort Row

Sebelum melakukan merge join dilakukan sort row pada column yang akan dipakai sebagai key pada merge join untuk memastikan hasil join yang benar.

* Merge Join

Penggunaan Merge join untuk melakukan join kiri antara tabel-tabel tersebut. Pastikan kondisi join sesuai dengan logika kueri SQL, seperti menggabungkan Users dengan Address\_User berdasarkan Address\_ID, dan seterusnya.

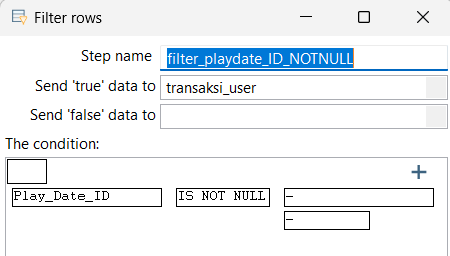
* + 1. **Pembuatan Tabel Fact**

****

Setelah mendapatkan data transaksi, langkah selanjutnya adalah melakukan proses ETL (Extract, Transform, Load) dengan Pentaho. Berikut adalah langkah-langkah umum dalam proses ETL menggunakan Pentaho Data Integration (PDI):

* Filter Playdate\_id NOT NULL

Dikarenakan ada user yang tidak melakukan transaksi (memutar lagu) terdapat row yang null oleh karena itu pada tahap ini dilakukan filter row terhadap field Playdate\_ID.

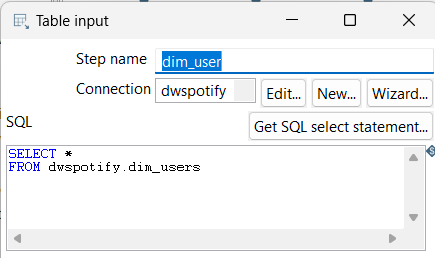


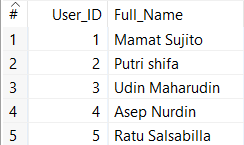
* Input Table Dimensi

Konfigurasikan Table Input untuk setiap tabel dimensi yang terlibat seperti sebagai berikut:

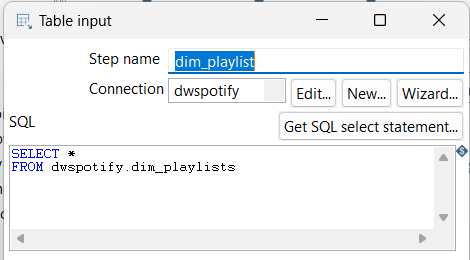
1. dim\_user

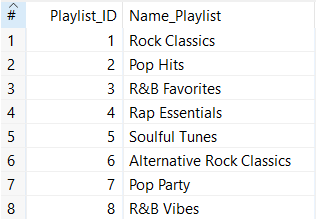
Input dim\_user dilakukan dengan menggunakan Input table dengan mengkonesikan database ke pentaho dan mengambil data dengan menggunakan script SQL.



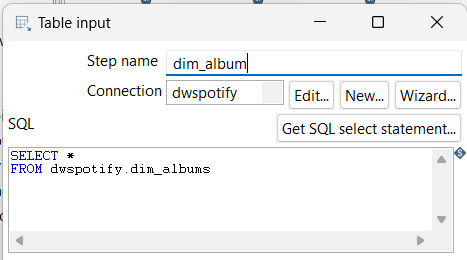


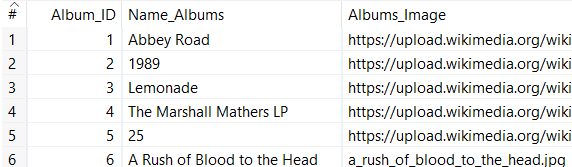
1. dim\_playlist



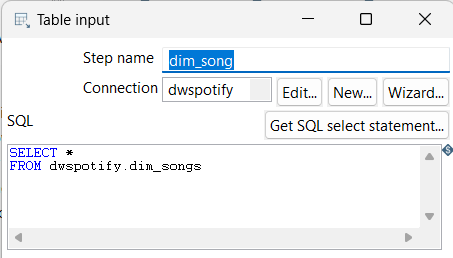


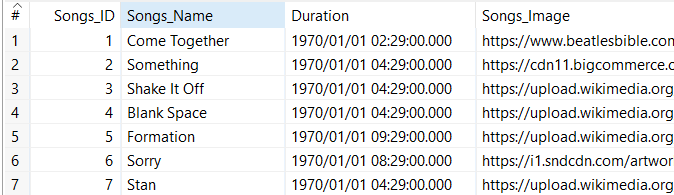
1. dim\_album



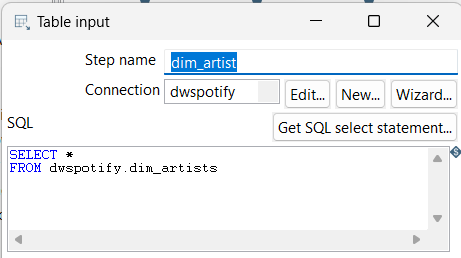


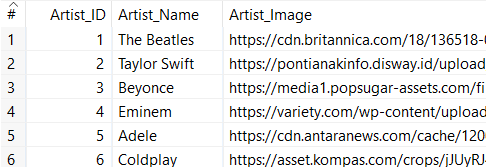
1. dim\_song



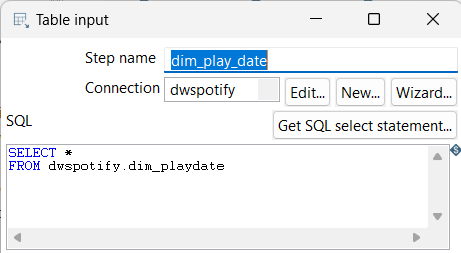


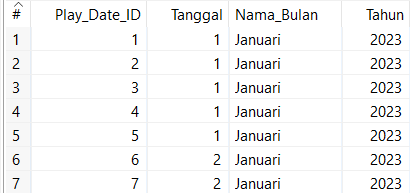
1. dim\_artist





1. dim\_playdate

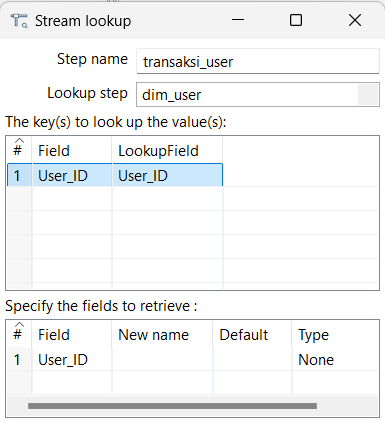




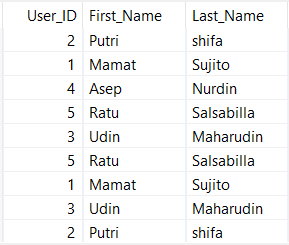
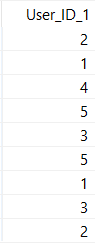
* Stream Lookup

1. Transaksi\_user

Mengintegrasikan data dari tabel dimensi seperti dim\_user ke dalam aliran data transaksi utama, dengan mencocokkan berdasarkan User\_ID

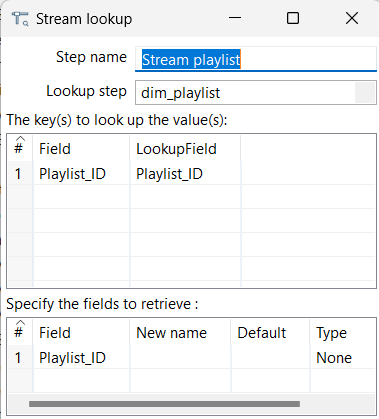


Berikut adalah hasil dari Transaksi\_user:

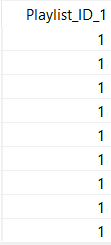
 

1. Stream Playlist

Mengintegrasikan data dari tabel dim\_playlist ke dalam aliran data transaksi, dengan mencocokkan berdasarkan Playlist\_ID

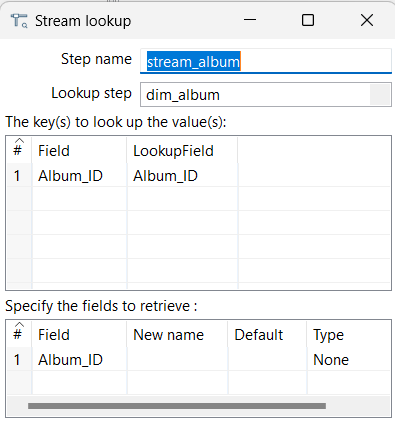


Berikut hasil dari tahap Stream playlist:

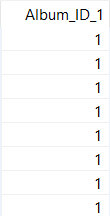
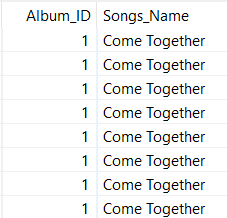
 

1. Stream album

Mengintegrasikan data dari tabel  dim\_album ke dalam aliran data, dengan mencocokkan berdasarkan Album\_ID

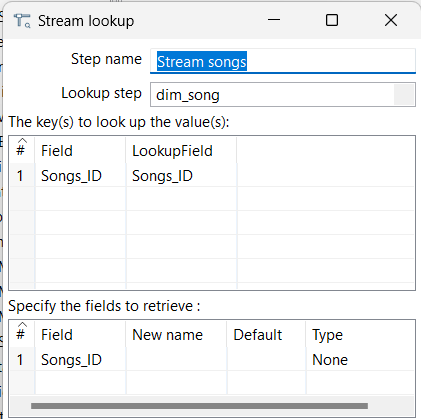


Berikut adalah hasil dari tahap Stream album:

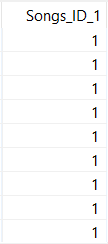
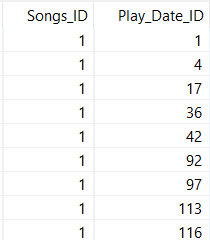


1. Stream songs

Mengintegrasikan data dari tabel  dim\_song ke dalam aliran data, dengan mencocokkan berdasarkan Songs\_ID

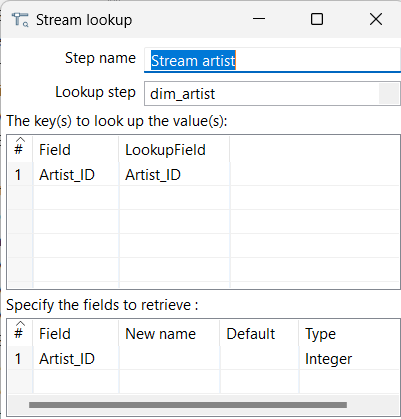


Berikut adalah hasil dari tahap Stream songs:

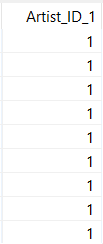


1. Stream artist

Mengintegrasikan data dari tabel  dim\_artist ke dalam aliran data, dengan mencocokkan berdasarkan dim\_artist

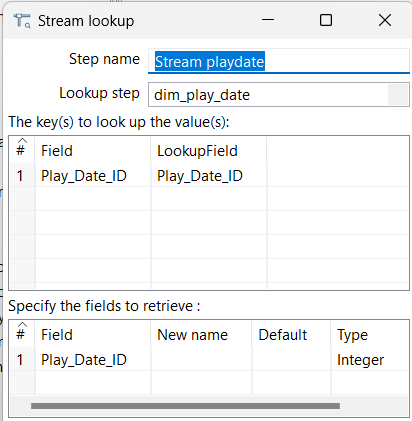


Berikut adalah hasil dari tahap Stream artist:

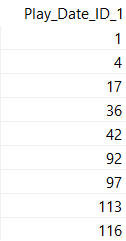
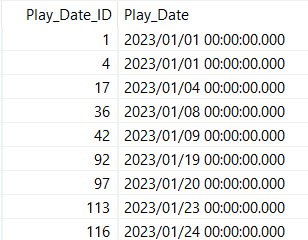


1. Stream playdate

Mengintegrasikan data dari tabel  dim\_play\_date ke dalam aliran data, dengan mencocokkan berdasarkan Play\_Date\_ID

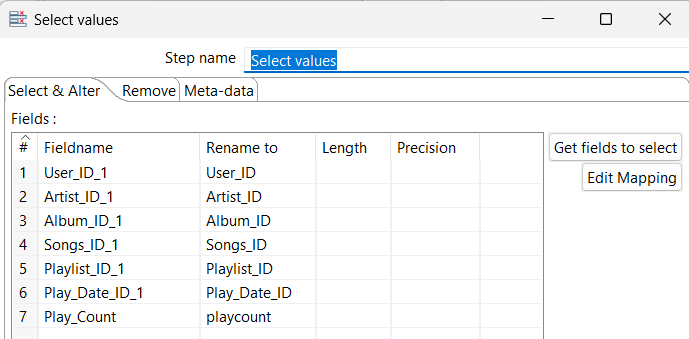


Berikut adalah hasil dari tahap Stream playdate:



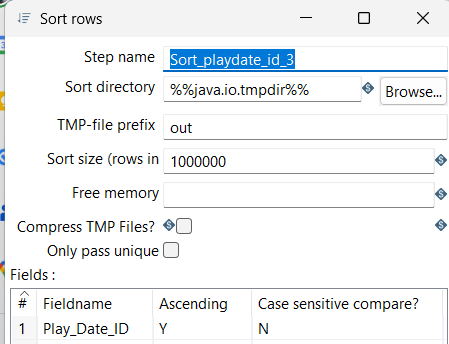
* Select Values

Pada tahapan dilakukan pemilihan field dan pengantian nama field sesuai dengan kebutuhan transformasi.

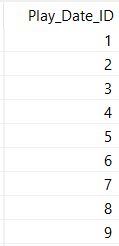
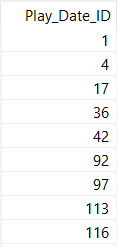


* Sort\_Playdate\_id

Dilakukan Sorting secara ascending terhadap Play\_Date\_ID untuk memastikan tahapan pemberian fact\_id dilakukan secara berurutan.

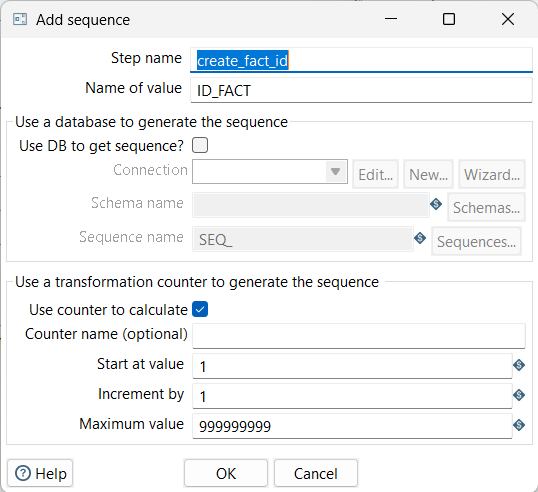


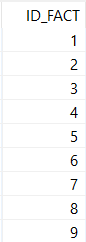
Berikut adalah hasil dari sebelum dan sesudah dilakukannya sort:



* Create fact\_ID

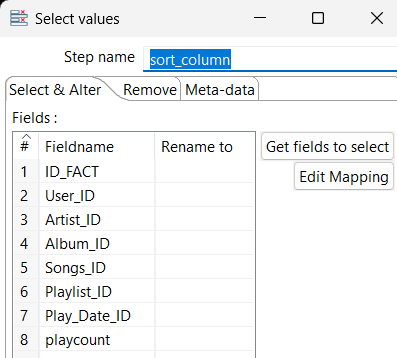
Pembuatan ID fact dengan bantuan add sequence dimulai dari angka 1 sampai 999999999.

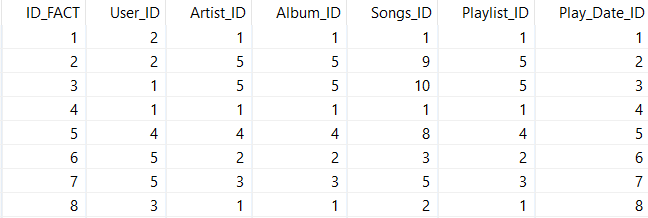




* Sort columns

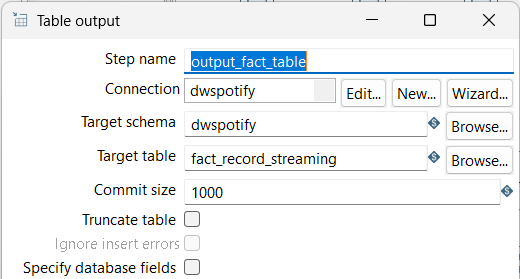
Untuk memindahkan columns ID\_FACT ke posisi paling kiri atau column pertama digunakan select values untuk mencapai hal tersebut.



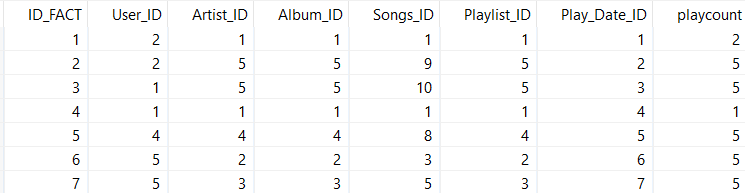


* Output fact table

Output dari hasil transformasi disimpan pada database dwspotify dengan menggunakan table output.



Berikut adalah hasil dari Output fact table

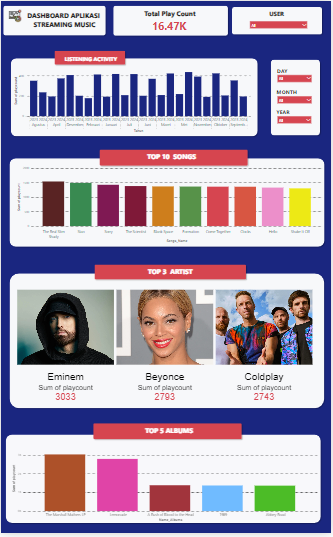


* 1. **Pembuatan Dashboard Menggunakan Power BI**

Setelah kami melakukan ETL menggunakan Pentaho, langkah terakhir adalah melakukan visualisasi menggunakan Power BI. Power BI adalah platform bisnis analitik yang memungkinkan perusahaan untuk mengolah dan memvisualisasikan data dalam bentuk yang mudah dipahami. Tujuan yang ingin dicapai melalui visualisasi ini adalah sebagai berikut:

* Perusahaan dapat mengetahui jumlah pendengar per bulan: Dengan memantau jumlah pendengar setiap bulan, perusahaan dapat mengidentifikasi tren dan pola dalam perilaku pendengar, serta merencanakan strategi pemasaran yang lebih efektif.
* Mengetahui top 10 lagu berdasarkan jumlah playcount: Dengan mengetahui lagu-lagu yang paling sering diputar, perusahaan dapat memahami preferensi musik pendengar dan dapat mengoptimalkan katalog musik mereka.
* Mengetahui top 3 artis berdasarkan jumlah playcount: Dengan informasi ini, perusahaan dapat menentukan artis yang paling populer dan mungkin fokus pada promosi atau kerjasama dengan artis-artis tersebut.
* Mengetahui top 5 album berdasarkan jumlah playcount: Data ini membantu perusahaan memahami album mana yang paling sukses dan dapat memberikan wawasan tentang preferensi album pendengar.

Nantinya, dashboard ini akan menggambarkan data baik untuk setiap pengguna individu maupun secara keseluruhan. Gambar 3.7 menunjukan dashboard dari aplikasi streaming music.



Gambar 3.7 Dashboard Aplikasi Streaming Music.

Pemilihan setiap pengguna didasarkan pada pengguna terpilih yang akan dijadikan subjek analisis atau laporan. Dengan demikian, visualisasi ini akan memberikan wawasan mendalam dan mendukung pengambilan keputusan yang lebih baik dalam strategi pemasaran dan pengelolaan katalog musik perusahaan. Untuk dashboard dapat dilihat pada link dibawah ini:

* <https://app.powerbi.com/view?r=eyJrIjoiNTAzMmQ0OGYtMTlhNi00MDFhLWIzYjQtMzdkYzdiMzIyM2YwIiwidCI6IjM0ODViOTYzLTgyYmEtNGE2Zi04MTBmLWI1Y2MyMjZmZjg5OCIsImMiOjEwfQ%3D%3D>

# **BAB IV**

# **KESIMPULAN**

Menggunakan Pentaho Data Integration (PDI) untuk proses ETL (Extract, Transform, Load) memungkinkan pengelolaan dan integrasi data transaksi secara efisien. Proses ini dimulai dengan mengekstraksi data dari database yang terhubung, kemudian mentransformasi data untuk membersihkan dan memformatnya sesuai kebutuhan analisis, dan akhirnya memuat data yang sudah ditransformasi ke dalam basis data tujuan. Setelah itu, langkah berikutnya adalah membuat tabel dimensi dan fakta. Tabel dimensi menyimpan atribut atau karakteristik data yang akan digunakan untuk analisis, seperti tabel `dim\_user` yang berisi informasi pengguna, `dim\_song` yang berisi informasi lagu, dan `dim\_artist` yang menyimpan data artis dan lainnya. Tabel fakta, seperti `fact\_streaming`, mencatat transaksi atau kejadian tertentu seperti aktivitas pemutaran lagu, dengan kolom-kolom yang mereferensikan kunci dari tabel dimensi dan data pengukuran seperti jumlah pemutaran (play\_count). Setelah tabel dimensi dan fakta siap, langkah selanjutnya adalah membuat dashboard menggunakan Power BI. Dashboard ini dapat menampilkan metrik penting seperti lima lagu teratas berdasarkan jumlah pemutaran, aktivitas pengguna mendengarkan lagu, artis teratas beserta lagu-lagu mereka, serta jumlah total pemutaran untuk setiap artis. Dengan Power BI, visualisasi data menjadi menarik dan interaktif, memberikan wawasan yang berharga bagi pengguna, dan memungkinkan mereka memahami tren musik serta preferensi mendengarkan dengan lebih baik. Integrasi antara PDI dan Power BI memastikan bahwa data yang ditampilkan selalu akurat dan up-to-date, mendukung pengambilan keputusan yang lebih baik.

# **DAFTAR PUSTAKA**

Anardani, S., STT, L. S., & Maghfur, A. (2019). Analysis of business intelligence system design for student performance monitoring. *Journal of Physics: Conference Series*, *1381*(1), 012015. https://doi.org/10.1088/1742-6596/1381/1/012015

Hasan, F. N., & Sudaryana, I. K. (2022). Penerapan Business Intelligence & Online Analytical Processing untuk Data-Data Penelitian dan Luarannya pada Perguruan Tinggi Menggunakan Pentaho. *Infotech: Journal of Technology Information*, *8*(2), Article 2. https://doi.org/10.37365/jti.v8i2.143

Lüders, M. (2021). Ubiquitous tunes, virtuous archiving and catering for algorithms: The tethered affairs of people and music streaming services. *Information, Communication & Society*, *24*(15), 2342–2358. https://doi.org/10.1080/1369118X.2020.1758742

Pabinger, D., & Mayr, S. (2019). *Controlling und Business Intelligence & Analytics*. https://doi.org/10.1007/978-3-658-27723-9\_4

Patel, J. A., & Sharma, P. (2020). Online Analytical Processing for Business Intelligence in Big Data. *Big Data*, *8*(6), 501–518. https://doi.org/10.1089/big.2020.0045

Sheng, Y., Tomasic, A., Zhang, T., & Pavlo, A. (2019). Scheduling OLTP transactions via learned abort prediction. *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, 1–8. https://doi.org/10.1145/3329859.3329871

Webster, J. (2020). Taste in the platform age: Music streaming services and new forms of class distinction. *Information, Communication & Society*, *23*(13), 1909–1924. https://doi.org/10.1080/1369118X.2019.1622763

Zhang, Q., Dong, J., Chen, H., Li, W., Huang, F., & Huang, X. (2024). *Structure Guided Large Language Model for SQL Generation* (arXiv:2402.13284). arXiv. https://doi.org/10.48550/arXiv.2402.13284