



## **LAPORAN AKHIR *DATA WAREHOUSE***

### **Perancangan Data Warehouse untuk Industri Manufaktur : Tesla**

#### **Kelompok 7**

Elia Meylani Simanjuntak	(122450026)
Chalifia Wananda	(122450076)
Sahid Maulana	(122450109)
Muhammad Rafif Vivaldi	(122140026)

**PROGRAM STUDI SAINS DATA  
FAKULTAS SAINS  
INSTITUT TEKNOLOGI SUMATERA**

**Mei 2025**

## DAFTAR ISI

<b>1. Pendahuluan.....</b>	<b>3</b>
<b>1.1 Latar Belakang.....</b>	<b>3</b>
<b>1.2 Tujuan.....</b>	<b>3</b>
<b>1.3 Lingkup Sistem.....</b>	<b>3</b>
<b>2. Metodologi.....</b>	<b>4</b>
<b>2.1 Tools.....</b>	<b>5</b>
<b>2.2 Metodologi Pengembangan.....</b>	<b>5</b>
<b>2.3 Analisis Kebutuhan.....</b>	<b>6</b>
<b>3. Desain.....</b>	<b>10</b>
<b>3.1 Desain Konseptual.....</b>	<b>10</b>
<b>3.1.1 Desain Tabel Dimensi.....</b>	<b>10</b>
<b>3.1.2 Desain Tabel Fakta.....</b>	<b>11</b>
<b>3.2 Desain Logikal.....</b>	<b>13</b>
<b>3.2.1 Design and Implement Indeks.....</b>	<b>13</b>
<b>3.2.2 Desain Storage.....</b>	<b>14</b>
<b>3.2.3 Design and implement partitioned tables and views.....</b>	<b>16</b>
<b>4. Implementasi.....</b>	<b>18</b>
<b>4.1 Proses.....</b>	<b>18</b>
<b>4.2 Hasil.....</b>	<b>23</b>
<b>5. Evaluasi.....</b>	<b>24</b>
<b>6. Rencana Pengembangan.....</b>	<b>24</b>
<b>7. Tim Proyek.....</b>	<b>25</b>

## **1. Pendahuluan**

### **1.1 Latar Belakang**

Industri manufaktur adalah sektor yang memproduksi barang dari bahan mentah melalui proses mekanis, kimia, atau teknologi. Di era modern, industri ini bergantung pada otomatisasi, efisiensi, dan pemanfaatan big data untuk pengambilan keputusan cepat dan akurat. Manufaktur modern mengintegrasikan sistem produksi fisik dengan sistem informasi digital. Dalam manufaktur otomotif, kompleksitas meningkat karena melibatkan ribuan komponen yang saling terhubung, serta rantai pasok global dan regulasi yang ketat.

Tesla Inc. adalah contoh perusahaan yang menggabungkan manufaktur otomotif dengan teknologi tinggi, memproduksi kendaraan listrik, baterai, sistem autopilot, perangkat lunak, dan infrastruktur pengisian daya. Dengan kompleksitas data yang tinggi, Tesla membutuhkan sistem Data Warehouse (DW) untuk mengintegrasikan, menyimpan, dan menganalisis data guna mendukung keputusan strategis seperti :

- a. Mengoptimalkan produksi dan distribusi dengan menganalisis data historis tentang keterlambatan produksi dan pengiriman kendaraan, Tesla dapat mengidentifikasi pola dan tren, serta merencanakan kapasitas produksi dan distribusi kendaraan secara lebih efisien.
- b. Menganalisis kepuasan pelanggan dengan menggunakan Data Warehouse untuk menganalisis umpan balik pelanggan dan tingkat kepuasan terhadap berbagai model kendaraan memungkinkan perusahaan untuk meningkatkan kualitas produk dan layanan mereka.
- c. Pengelolaan Sumber Daya dan Layanan Purna Jual, dengan data mengenai penggantian baterai dan kunjungan servis dapat dianalisis untuk memprediksi kebutuhan pemeliharaan, serta memberikan informasi strategis bagi tim pengembangan produk.

### **1.2 Tujuan**

Tujuan dirancangnya sistem Data Warehouse yaitu ;

- a. Meningkatkan efisiensi produksi dan mengurangi keterlambatan dengan menyediakan data historis dan analisis tren Production Delay berdasarkan model, pabrik, dan waktu produksi.
- b. Mengoptimalkan distribusi kendaraan ke wilayah yang paling membutuhkan, berdasarkan tren Units Delivered, rating pelanggan, dan tingkat servis per wilayah.
- c. Memprediksi kebutuhan layanan dan perawatan dengan menganalisis data Service Visits dan Battery Replacement Rate untuk setiap model dan software version.
- d. Mendukung pengembangan produk dan perangkat lunak dengan menganalisis Avg Customer Rating, Number of Recalls, dan Software Version untuk mengidentifikasi titik lemah produk.
- e. Meningkatkan kepuasan dan loyalitas pelanggan melalui analitik terhadap rating, efisiensi kendaraan, dan nilai jual kembali (Resale Value (%)) berdasarkan kombinasi model dan wilayah.

### **1.3 Lingkup Sistem**

Tesla, sebagai produsen kendaraan listrik berteknologi tinggi, menghadapi kompleksitas data dari berbagai sumber dan unit bisnis. Oleh karena itu, kebutuhan utama misi ini adalah:

- a. Integrasi data lintas fungsi bisnis (produksi, distribusi, teknis, dan layanan pelanggan) untuk mendukung pengambilan keputusan strategis.
- b. Pemantauan efisiensi operasional, seperti mengurangi keterlambatan produksi dan mendistribusikan kendaraan tepat waktu ke berbagai wilayah.
- c. Analisis kualitas produk dan layanan, seperti memahami frekuensi servis dan penggantian baterai.

- d. Menganalisis performa teknis kendaraan, termasuk efisiensi, fitur autopilot, dan update software.
- e. Evaluasi pengalaman pelanggan, termasuk kepuasan, nilai jual kembali, dan loyalitas.

Kebutuhan ini mendorong perancangan data warehouse multidimensi yang mampu menjawab pertanyaan analitis lintas waktu, wilayah, model kendaraan, dan segmentasi pelanggan.

Dalam perancangan Data Warehouse untuk Tesla, pendekatan yang digunakan adalah Star Schema, yaitu suatu model multidimensi di mana satu tabel fakta utama berada di pusat dan dihubungkan secara langsung ke beberapa tabel dimensi. Hubungan antar tabel bersifat one-to-many, yaitu satu entri pada tabel dimensi bisa berhubungan dengan banyak entri pada tabel fakta. Star schema dipilih karena:

- a. Mudah dipahami dan intuitif bagi pengguna teknis maupun non-teknis.
- b. Sangat cocok untuk query analitis (OLAP) seperti drill-down, roll-up, slice, dan dice.
- c. Mendukung hierarki dalam dimensi, misalnya Date → Quarter → Year.
- d. Mempercepat performa query agregat yang sering digunakan dalam pelaporan.
- e. Fleksibel dan scalable, bisa ditambah entitas baru tanpa mengubah struktur inti. Struktur ini selaras dengan kebutuhan bisnis Tesla dalam memahami performa produk, efisiensi produksi, distribusi regional, dan kepuasan pelanggan dari berbagai dimensi.

## 2. Metodologi

Pembangunan sistem Data Warehouse (DW) untuk industri manufaktur Tesla dilaksanakan dengan pendekatan yang sistematis dan terstruktur. Tujuan utama dari metodologi ini adalah untuk menjamin bahwa seluruh proses pembangunan sistem — dari pengumpulan kebutuhan, perancangan, implementasi, hingga evaluasi — dilakukan secara logis dan saling terhubung. Sistem DW yang dirancang ditujukan untuk mendukung analisis multidimensi terhadap data operasional Tesla, termasuk produksi kendaraan, pengiriman, performa teknis, dan kepuasan pelanggan.

Dalam proyek ini, metodologi pengembangan yang digunakan adalah Waterfall, yaitu pendekatan linier dan sekuensial yang mengharuskan setiap tahap diselesaikan secara penuh sebelum melanjutkan ke tahap berikutnya. Waterfall dianggap cocok untuk proyek ini karena ruang lingkup proyek telah ditentukan dengan jelas sejak awal, struktur data yang digunakan bersifat terdefinisi, dan kebutuhan pengguna telah diidentifikasi dengan baik.

Pendekatan ini juga dipadukan dengan penggunaan arsitektur tiga lapis (Three-Tier Architecture) yang umum digunakan dalam pengembangan sistem data warehouse modern. Tiga lapisan tersebut terdiri dari:

1. Staging Layer – tempat penyimpanan awal data mentah hasil ekstraksi dari sumber eksternal (CSV, API).
2. Warehouse Layer – tempat dilakukannya proses transformasi dan penyimpanan data ke dalam struktur skema bintang (Star Schema) dengan optimalisasi performa melalui indexing dan partisi.
3. Reporting/Analysis Layer – menyediakan data yang telah terstruktur untuk dianalisis melalui query analitis, visualisasi dashboard, atau pembuatan data mart dan indexed views.

Dengan mengombinasikan pendekatan metodologis Waterfall dan arsitektur teknis Three-Tier, sistem yang dibangun diharapkan mampu menyediakan data yang akurat, terintegrasi, dan siap dianalisis oleh seluruh departemen strategis dalam organisasi Tesla.

## 2.1 Tools

Dalam mendukung pengembangan sistem DW ini, berbagai alat bantu (*tools*) digunakan untuk memfasilitasi proses ekstraksi, transformasi, pemuatan data, dan analisis. Berikut adalah tools utama yang digunakan:

- a. Python (dengan library pandas, psycopg2):  
Digunakan dalam proses *Extract* dan *Transform*. Python memudahkan pembacaan file sumber (CSV), pembersihan data (cleaning), standarisasi format, serta transformasi nilai atribut agar sesuai dengan tabel dimensi dan fakta.
- b. PostgreSQL / SQL Server:  
Merupakan sistem manajemen basis data yang digunakan untuk menyimpan data warehouse. Tools ini mendukung pembuatan skema bintang, implementasi indexing (clustered dan non-clustered), partisi tabel, serta pengelolaan filegroup penyimpanan.
- c. Airflow atau cronjob:  
Digunakan untuk mengatur workflow ETL secara otomatis dan terjadwal. Automasi ini penting agar proses pemrosesan data rutin (harian/kuartalan) dapat dilakukan tanpa intervensi manual.

Kombinasi tools tersebut memastikan bahwa proses pengolahan data berjalan efisien, hasilnya akurat, serta mudah dipahami oleh pemangku kepentingan bisnis dari berbagai level.

## 2.2 Metodologi Pengembangan

Proyek ini dibagi ke dalam empat tahap utama (misi) berdasarkan pendekatan Waterfall. Setiap tahap menyelesaikan satu bagian kritical dari sistem Data Warehouse dan menjadi dasar bagi tahapan berikutnya. Berikut adalah penjelasan tiap tahapan:

Tahapan (Misi)	Deskripsi Kegiatan	Output
Misi 1: Analisis Kebutuhan	Identifikasi stakeholder utama, pengumpulan kebutuhan informasi dari tiap departemen Tesla (produksi, distribusi, layanan pelanggan, pengembangan produk,	Dokumen kebutuhan sistem, daftar stakeholder, atribut data penting.
Misi 2: Desain Konseptual	Perancangan model multidimensi berbasis Star Schema yang terdiri dari satu tabel fakta dan lima tabel dimensi (waktu, model, wilayah, performa, pelanggan).	Skema konseptual, ERD, definisi tabel fakta dan dimensi.
	Implementasi struktur logikal database dengan perintah	Struktur SQL database, indeks, skema partisi, staging table,

	CREATE TABLE, indexing (clustered, non-clustered, composite), partisi tabel fakta berdasarkan tahun, serta pemisahan filegroup penyimpanan.	dokumentasi storage (Three-Tier Architecture).
Misi 4: Implementasi dan Evaluasi	Proses ETL: ekstraksi data dari sumber (CSV), transformasi data ke format warehouse, dan pemuatan ke tabel dimensi dan fakta. Pembuatan indexed view dan data mart untuk analitik. Evaluasi performa sistem.	Data warehouse terisi, view agregat, data mart, laporan evaluasi performa dan fungsionalitas.

## 2.3 Analisis Kebutuhan

### Stakeholder

Departemen	Peran dan Kebutuhan Informasi
Produksi	Memantau keterlambatan produksi berdasarkan model dan waktu.
Distribusi & Logistik	Menganalisis jumlah pengiriman kendaraan per wilayah dan kuartal.
Customer Service	Mengidentifikasi model dengan angka kunjungan servis dan penggantian baterai tertinggi.
Tim Pengembangan Produk & Software	Mengevaluasi hubungan antara versi software, rating pelanggan, dan recall.
Manajemen Regional Penjualan	Menganalisis tren penjualan dan efisiensi kendaraan di berbagai wilayah.
Eksekutif Strategis / CEO	Melihat performa bisnis secara keseluruhan untuk pengambilan keputusan strategis.

Untuk merespons masalah bisnis di atas, maka sistem Data Warehouse dirancang untuk membantu pencapaian tujuan-tujuan bisnis strategis berikut:

- Meningkatkan efisiensi produksi dan mengurangi keterlambatan dengan menyediakan data historis dan analisis tren Production Delay berdasarkan model, pabrik, dan waktu produksi.
- Mengoptimalkan distribusi kendaraan ke wilayah yang paling membutuhkan, berdasarkan tren Units Delivered, rating pelanggan, dan tingkat servis per wilayah.
- Memprediksi kebutuhan layanan dan perawatan dengan menganalisis data Service Visits dan Battery Replacement Rate untuk setiap model dan software version.
- Mendukung pengembangan produk dan perangkat lunak dengan menganalisis Avg Customer Rating, Number of Recalls, dan Software Version untuk mengidentifikasi titik lemah produk.
- Meningkatkan kepuasan dan loyalitas pelanggan melalui analitik terhadap rating, efisiensi kendaraan, dan nilai jual kembali (Resale Value (%)) berdasarkan kombinasi model dan

wilayah.

#### Tabel Fakta dan Tabel Dimensi

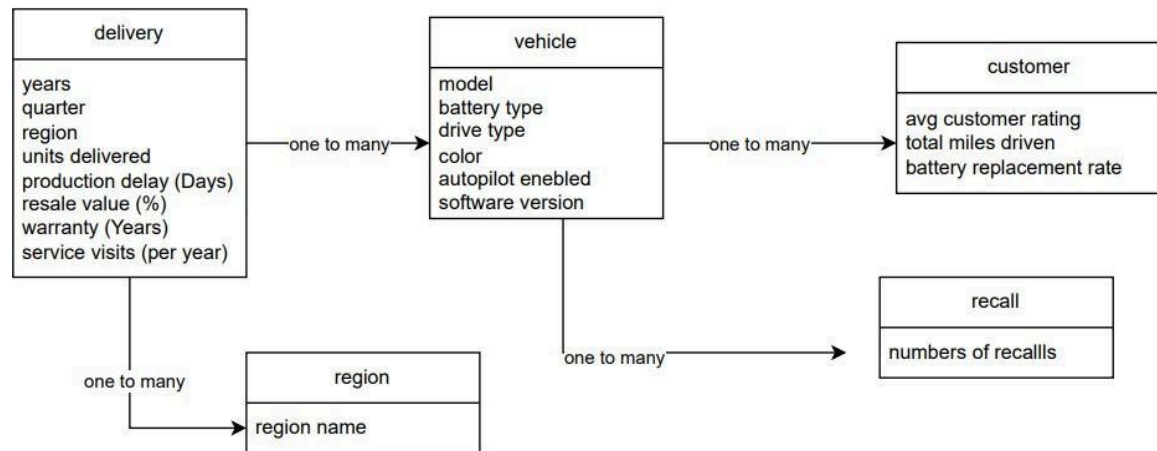
##### - Tabel Fakta

Nama Kolom	Deskripsi
Units Delivered	Jumlah kendaraan Tesla yang dikirim ke wilayah tertentu dalam satu kuartal.
Production Delay (Days)	Jumlah hari keterlambatan dalam proses produksi kendaraan.
Average Price (USD)	Harga rata-rata kendaraan yang dikirim.
Service Visits (Per Year)	Frekuensi kunjungan servis tahunan untuk model tertentu.
Battery Replacement Rate	Persentase kendaraan yang mengganti baterai dalam satu tahun.
Total Miles Driven	Total jarak tempuh yang dicapai kendaraan yang dikirim.
Number of Recalls	Jumlah recall yang terjadi pada model kendaraan.
CO2 Saved (Tons)	Jumlah emisi CO2 yang dihemat oleh kendaraan listrik.
Resale Value (%)	Persentase nilai jual kembali kendaraan.
Avg Customer Rating	Rata-rata rating pelanggan untuk model tertentu.

##### - Tabel Dimensi

Nama Dimensi	Kolom	Deskripsi
Dim_Waktu	Year, Quarter	Menjelaskan periode waktu pengiriman kendaraan.
Dim_Model	Model, Battery Type, Drive Type, Color	Detail tipe kendaraan yang diproduksi dan dikirim.
Dim_Wilayah	Region	Wilayah geografis tempat kendaraan dikirim.
Dim_Performa	Range, Efficiency, Charging Time, Autopilot Enabled, Software Version	Metrik teknis performa kendaraan.
Dim_Pelanggan	Warranty, Update Frequency, Customer Rating	Informasi terkait layanan purna jual dan pengalaman pengguna.

## ERD (Entity Relationship Diagram)



## Sumber Data

Data berasal dari dataset di Kaggle dengan judul “Tesla Vehicle Deliveries Dataset (2012-2024)” pada link :

<https://www.kaggle.com/datasets/khushikyad001/tesla-vehicle-deliveries-dataset-2012-2024/data>

## Metadata

Nama Kolom	Tipe Data	Deskripsi
Year	Integer	Tahun pengiriman kendaraan.
Quarter	String	Kuartal (Q1, Q2, Q3, Q4).
Region	String	Wilayah geografis pengiriman (misal: North America, Asia).
Model	String	Jenis Kendaraan Tesla (Model S, X, Y, Semi, dll).
Units Delivered	Integer	Jumlah kendaraan yang dikirim.
Production Delay (Days)	Integer	Jumlah hari keterlambatan produksi.
Average Price (USD)	Float	Harga rata-rata kendaraan yang dikirim dalam USD.
Battery Type	String	Jenis baterai yang digunakan (Standard, Long Range, Performance).
Autopilot Enabled	Boolean	Status apakah fitur autopilot aktif.



Avg Customer Rating	Float	Skor rata-rata dari pelanggan untuk model tersebut (skala 1–5).
---------------------	-------	---

### 3. Desain

#### 3.1 Desain Konseptual

Desain konseptual data warehouse yang diterapkan dalam studi ini dibentuk dengan mempertimbangkan kebutuhan bisnis Tesla yang menuntut integrasi data dari berbagai lini operasional seperti produksi, distribusi, layanan pelanggan, dan inovasi produk. Pemilihan *star schema* karena kesederhanaan strukturnya, tetapi karena pendekatan ini dapat mendukung pengambilan keputusan strategis secara cepat dan tepat. Dengan bentuk relasi langsung antara tabel fakta dan tabel dimensi, setiap entitas yang telah dirancang sebelumnya seperti waktu, wilayah, model kendaraan, performa teknis, dan pelanggan dapat dengan mudah dikaitkan ke metrik utama dalam bisnis, seperti jumlah pengiriman kendaraan, keterlambatan produksi, atau rating pelanggan. Struktur ini secara langsung mencerminkan bagaimana Tesla perlu menganalisis data secara operasional dan historis.

Pendekatan tersebut juga kompatibel dengan berbagai alat Business Intelligence modern yang digunakan untuk pelaporan dan visualisasi. Hal ini akan mempermudah implementasi dashboard yang bersifat operasional maupun eksekutif, serta memungkinkan pelaporan yang lebih interaktif dan berkelanjutan. Dengan kata lain, desain konseptual yang dibangun tidak hanya menjawab kebutuhan integrasi data lintas fungsi bisnis Tesla, tetapi juga mempertimbangkan efisiensi operasional, kemudahan analisis multidimensi, serta kesiapan sistem.

Rancangan data warehouse yang telah disusun dalam studi ini didasarkan pada data yang tersedia dari dataset publik bertema pengiriman kendaraan Tesla selama periode 2012–2024. Setelah ditelaah, struktur dan isi dari dataset tersebut menunjukkan kesesuaian yang tinggi dengan model konseptual yang dirancang, baik dalam hal cakupan informasi, format data, maupun kejelasan atribut yang dibutuhkan untuk membentuk tabel fakta dan tabel dimensi.

Dataset memuat kolom-kolom penting seperti jumlah kendaraan yang dikirim, estimasi harga rata-rata per model, versi perangkat lunak, efisiensi kendaraan, serta rating pelanggan. Semua atribut ini secara langsung dapat dikonversi menjadi bagian dari tabel fakta, khususnya untuk menyimpan metrik utama yang menggambarkan performa operasional dan pengalaman pelanggan Tesla. Selain itu, informasi deskriptif seperti model kendaraan, jenis baterai, sistem penggerak, wilayah penjualan (negara atau kawasan), serta rentang waktu (kuartal dan tahun) tersedia dalam bentuk terstruktur dan dapat digunakan untuk membentuk tabel-tabel dimensi yang diperlukan.

##### 3.1.1 Desain Tabel Dimensi

###### a. Tabel Dimensi

Dimensi adalah komponen deskriptif yang menyediakan konteks bagi data dalam tabel fakta. Dimensi memungkinkan pengguna melakukan analisis berdasarkan waktu, lokasi, karakteristik produk, performa teknis, dan tipe pelanggan.

###### 1) Dim\_Waktu

- a. Atribut: Date, Month, Quarter, Year
- b. Hierarki: Date → Quarter → Year
- c. Fungsi: Mendukung analisis tren jangka pendek dan panjang; sangat berguna untuk laporan kuartalan, analisis musiman, dan pertumbuhan tahunan.

###### 2) Dim\_Model

- a. Atribut: Model, Battery Type, Drive Type, Color
  - b. Hierarki: Model → Battery Type → Drive Type → Color
  - c. Fungsi: Memungkinkan eksplorasi performa berdasarkan tipe kendaraan, preferensi pasar terhadap warna/model, dan efisiensi spesifikasi.
- 3) Dim\_Wilayah
- a. Atribut: City, Region, Country
  - b. Hierarki: City → Region → Country
  - c. Fungsi: Berguna untuk pelacakan distribusi kendaraan, efektivitas jaringan logistik, serta analisis preferensi konsumen berdasarkan geografi.
- 4) Dim\_Performa
- a. Atribut: Software Version, Efficiency, Charging Time, Autopilot Enabled
  - b. Hierarki: Software Version → Efficiency → Autopilot Enabled
  - c. Fungsi: Untuk mengevaluasi pengaruh teknologi digital terhadap pengalaman dan performa kendaraan; termasuk inovasi software Tesla.
- 5) Dim\_Pelanggan
- a. Atribut: Warranty Period, Update Frequency, Customer Rating
  - b. Hierarki: Warranty → Update Frequency → Customer Rating
  - c. Fungsi: Mendukung segmentasi pengguna berdasarkan loyalitas, frekuensi pembaruan, dan tingkat kepuasan.

Nama Dimensi	Kolom	Deskripsi
Dim_Waktu	Year, Quarter	Menjelaskan periode waktu pengiriman kendaraan.
Dim_Model	Model, Battery Type, Drive Type, Color	Detail tipe kendaraan yang diproduksi dan dikirim.
Dim_Wilayah	Region	Wilayah geografis tempat kendaraan dikirim.
Dim_Performa	Range, Efficiency, Charging Time, Autopilot Enabled, Software Version	Metrik teknis performa kendaraan.
Dim_Pelanggan	Warranty, Update Frequency, Customer Rating	Informasi terkait layanan purna jual dan pengalaman pengguna.

### 3.1.2 Desain Tabel Fakta

- a. Tabel Fakta: Fakta\_Pengiriman\_Kendaraan

Tabel fakta menyimpan data numerik dan terukur yang merupakan hasil dari proses bisnis operasional Tesla. Data pada tabel ini akan dihubungkan dengan dimensi-dimensi kontekstual seperti waktu, model, wilayah, performa, dan pelanggan.

Kolom Fakta	Fungsi dan Kegunaan
-------------	---------------------

Units Delivered	Mengukur volume output distribusi kendaraan; metrik utama performa produksi dan logistik.
Production Delay (Days)	Indikator efisiensi jalur produksi; mendeteksi bottleneck dan masalah manufaktur.
Average Price (USD)	Untuk menganalisis strategi harga per model, wilayah, atau waktu tertentu.
Service Visits	Mewakili keandalan kendaraan; frekuensi tinggi bisa menunjukkan cacat desain atau masalah komponen tertentu.
Battery Replacement Rate	Menilai daya tahan dan kualitas baterai; juga berperan dalam estimasi garansi dan biaya servis jangka panjang.
Total Miles Driven	Metrik pemakaian aktual oleh pelanggan; menunjukkan performa dan ketahanan kendaraan di lapangan.
Number of Recalls	Menunjukkan skala penarikan produk akibat cacat produksi atau update keamanan; terkait langsung dengan kualitas manufaktur.
CO2 Saved (Tons)	Estimasi dampak lingkungan positif dari kendaraan listrik; mendukung inisiatif ESG (Environmental, Social, Governance).
Resale Value (%)	Metrik penting untuk menilai nilai ekonomis jangka panjang kendaraan dan persepsi pasar terhadap kualitas produk Tesla.
Avg Customer Rating	Rata-rata kepuasan pelanggan berdasarkan survei atau feedback digital; digunakan untuk segmentasi pasar dan strategi perbaikan produk.

## 3.2 Desain Logikal

### 3.2.1 Desain Logikal Data Warehouse: Pendekatan 3-Tier

Desain penyimpanan dalam proyek ini mengikuti *Three-Tier Architecture* yang umum digunakan dalam pengembangan sistem data warehouse. Pendekatan ini memastikan data diproses secara terstruktur dan efisien, dari sumber mentah hingga penyajian analisis. Berikut adalah tiga lapisan yang diimplementasikan:

#### a. Tier 1: Data Source - Collection Layer

Pada lapisan ini, data berasal dari berbagai sumber eksternal yang mungkin termasuk file CSV, sistem operasional, dan API lainnya. Dalam kasus ini, data tentang pengiriman kendaraan Tesla berasal dari file CSV yang memuat berbagai informasi transaksi kendaraan.

Data yang diekstraksi dari sumber eksternal dimuat ke dalam tabel staging di Data Staging Layer. Tabel staging adalah tempat penyimpanan sementara untuk data mentah yang belum sepenuhnya diproses atau ditransformasikan. Data yang ada di tabel staging kemungkinan besar tidak dalam format yang sesuai untuk analisis, sehingga transformasi diperlukan.

```
1. CREATE TABLE Staging_Pengiriman_Kendaraan (  
2.     raw_id INT PRIMARY KEY AUTO_INCREMENT,  
3.     year INT,  
4.     quarter TEXT,  
5.     region TEXT,  
6.     model TEXT,  
7.     units_delivered INT,  
8.     average_price_usd FLOAT,  
9.     production_delay_days INT,  
10.    battery_type TEXT,  
11.    drive_type TEXT,  
12.    color TEXT,  
13.    range_km INT,  
14.    charging_time_min INT,  
15.    efficiency_kwh_per_100km FLOAT,  
16.    co2_saved_tons FLOAT,  
17.    customer_rating FLOAT,  
18.    warranty_years INT,  
19.    number_of_recalls INT,  
20.    software_version TEXT,  
21.    autopilot_enabled BOOLEAN,  
22.    total_miles_driven FLOAT,  
23.    update_frequency_per_year INT,  
24.    battery_replacement_rate FLOAT,  
25.    service_visits_per_year INT  
26. );  
27.
```

**Proses Mengimpor Data ke Staging Layer:**

```

1. LOAD DATA INFILE
   'C:\\path\\to\\tesla_vehicle_deliveries.csv'
2. INTO TABLE Staging
3. FIELDS TERMINATED BY ','
4. ENCLOSED BY '"'
5. LINES TERMINATED BY '\n'
6. IGNORE 1 ROWS;

```

Tabel Staging\_Pengiriman menyimpan data mentah yang diekstrak dari sumber luar. Tipe data masih umum (TEXT) karena belum dibersihkan. Digunakan sementara sebelum ETL.

b. Tier 2: Data Warehouse Layer (Lapisan Data Warehouse)

Setelah data dimuat ke lapisan staging dan dibersihkan serta ditransformasikan, data kemudian dipindahkan ke Data Warehouse Layer. Di lapisan ini, data akan dimasukkan ke dalam tabel fakta dan tabel dimensi yang telah dibangun menggunakan star schema (yang telah dijelaskan sebelumnya).

Fungsi Utama:

- Transformasi dan Pemetaan Data: Pada tahap ini, data dari staging diubah menjadi format yang lebih terstruktur, dihubungkan dengan ID dimensi untuk memastikan konsistensi dan keterkaitan.
- Memuat Data ke dalam Data Warehouse: Data dimasukkan ke dalam tabel dimensi (seperti Dim\_Model, Dim\_Wilayah, dll.) dan tabel fakta (seperti Fakta\_Pengiriman\_Kendaraan).
- Agregasi dan Penyimpanan Data: Data yang dimasukkan ke dalam tabel fakta siap untuk dianalisis menggunakan query OLAP.

**ETL untuk Tabel Fakta:** Merekam transaksi pengiriman kendaraan beserta nilai-nilai metrik dan referensi ke tabel-tabel dimensi.

```

1. CREATE TABLE Fakta_Pengiriman_Kendaraan (
2.   fakta_id INT AUTO_INCREMENT PRIMARY KEY,
3.   model_id INT,
4.   wilayah_id INT,
5.   waktu_id INT,
6.   performa_id INT,
7.   pelanggan_id INT,
8.   units_delivered INT,
9.   production_delay_days INT,
10.   average_price NUMERIC,
11.   service_visits INT,
12.   FOREIGN KEY (model_id) REFERENCES
    Dim_Model(model_id),
13.   FOREIGN KEY (wilayah_id) REFERENCES
    Dim_Wilayah(wilayah_id),
14.   FOREIGN KEY (waktu_id) REFERENCES

```

```

    Dim_Waktu(waktu_id),
15.     FOREIGN KEY (performa_id) REFERENCES
    Dim_Performa(performa_id),
16.     FOREIGN KEY (pelanggan_id) REFERENCES
    Dim_Pelanggan(pelanggan_id)
17. );
18.
19.

```

- Data transaksi dari tabel staging digabungkan dengan data dari tabel dimensi berdasarkan atribut yang sama.
- Nilai metrik seperti **units\_delivered**, **production\_delay\_days**, **average\_price**, dan **service\_visits** disimpan sebagai data numerik untuk analisis lebih lanjut.
- Semua entitas seperti model, wilayah, waktu, performa, dan pelanggan direpresentasikan oleh foreign key ID agar data lebih terstruktur dan efisien.

**ETL untuk Tabel Dimensi :**Menyusun data entitas utama dalam bentuk tabel dimensi agar dapat digunakan sebagai referensi dalam tabel fakta. Proses ini memastikan bahwa nilai-nilai unik dari masing-masing atribut disimpan hanya sekali.

```

1. INSERT INTO Dim_Waktu (year, quarter)
2. SELECT DISTINCT year, quarter
3. FROM Staging;
4.
5. INSERT INTO Dim_Model (model, battery_type,
   drive_type, color)
6. SELECT DISTINCT model, `Battery Type`, `Drive Type`,
   color
7. FROM Staging;
8.
9. INSERT INTO Dim_Wilayah (region)
10. SELECT DISTINCT region
11. FROM Staging;
12.
13. INSERT INTO Dim_Performa (software_version,
   efficiency, charging_time, autopilot_enabled)
14. SELECT DISTINCT `software version`, `Efficiency
   (kWh/100km)`, `Charging Time (Min)`, `Autopilot
   Enabled`
15. FROM Staging;
16.
17. INSERT INTO Dim_Pelanggan (warranty_period,
   update_frequency, customer_rating)
18. SELECT DISTINCT `Warranty (Years)`, `Update
   Frequency (Per Year)`, `Customer Rating`
19. FROM Staging;
20.

```

c. Tier 3: Analysis and Reporting Layer (OLAP)

Pada Analysis and Reporting Layer, kita melakukan OLAP untuk menganalisis data yang telah dimuat ke dalam Data Warehouse Layer. Proses ini melibatkan pengolahan data agregat menggunakan fungsi **SUM**, **AVG**, dan **COUNT** untuk menghasilkan laporan analitik.

Query berikut menghitung total penjualan berdasarkan tahun dan kuartal.

```
1. SELECT t.year, t.quarter, SUM(s.`Units Delivered` *  
   s.`Average Price (USD)`) AS total_sales  
2. FROM Staging_Pengiriman_Kendaraan s  
3. JOIN Dim_Waktu t ON s.year = t.year AND s.quarter =  
   t.quarter  
4. GROUP BY t.year, t.quarter  
5. ORDER BY t.year, t.quarter;
```

Query berikut menghitung rata-rata transaksi per pelanggan dengan menggunakan **AVG**.

```
1. SELECT c.pelanggan_id, AVG(s.`Units Delivered` *  
   s.`Average Price (USD)`) AS  
   avg_transaction_per_customer  
2. FROM Staging_Pengiriman_Kendaraan s  
3. JOIN Dim_Pelanggan c ON s.`Customer Rating` =  
   c.customer_rating  
4. GROUP BY c.pelanggan_id  
5. ORDER BY avg_transaction_per_customer DESC;
```

Query berikut mengidentifikasi produk terlaris per wilayah dengan menghitung jumlah unit yang terjual.

```
1. SELECT w.region, s.model, SUM(s.`Units Delivered`) AS  
   total_units_sold  
2. FROM Staging_Pengiriman_Kendaraan s  
3. JOIN Dim_Wilayah w ON s.region = w.region  
4. GROUP BY w.region, s.model  
5. ORDER BY total_units_sold DESC;
```

Query ini menghitung **persentase pertumbuhan penjualan** dari tahun ke tahun.

```
1. SELECT t.year,  
2.     SUM(s.`Units Delivered` * s.`Average Price  
   (USD)`) AS total_sales,  
3.     (SUM(s.`Units Delivered` * s.`Average Price  
   (USD)`) -  
4.     COALESCE(LAG(SUM(s.`Units Delivered` *  
   s.`Average Price (USD)`) OVER (ORDER BY t.year), 0)) /  
5.     COALESCE(LAG(SUM(s.`Units Delivered` *  
   s.`Average Price (USD)`) OVER (ORDER BY t.year), 1) *  
   100 AS year_growth  
6. FROM Staging_Pengiriman_Kendaraan s
```



```
7. JOIN Dim_Waktu t ON s.year = t.year
8. GROUP BY t.year
9. ORDER BY t.year;
```

#### Indexed View untuk Analitik

```
1. CREATE VIEW vw_AvgRatingPerModel
2. WITH SCHEMABINDING AS
3. SELECT
4.     m.model,
5.     AVG(f.avg_customer_rating) AS avg_rating
6. FROM dbo.Fakta_Pengiriman_Kendaraan f
7. JOIN dbo.Dim_Model m ON f.model_id = m.model_id
8. GROUP BY m.model;
9.
10. CREATE UNIQUE CLUSTERED INDEX idx_rating_model
11. ON vw_AvgRatingPerModel (model);
```

View vw\_AvgRatingPerModel menyediakan rata-rata rating pelanggan per model kendaraan. Indexed view ini mempercepat kueri pelaporan tanpa harus membaca tabel besar setiap saat.

#### Data Mart

```
1. CREATE TABLE DataMart_KepuasanPelanggan AS
2. SELECT
3.     f.delivery_date,
4.     m.model,
5.     r.region,
6.     f.avg_customer_rating,
7.     f.service_visits
8. FROM Fakta_Pengiriman_Kendaraan f
9. JOIN Dim_Model m ON f.model_id = m.model_id
10. JOIN Dim_Wilayah r ON f.wilayah_id = r.wilayah_id
11. WHERE f.avg_customer_rating IS NOT NULL;
```

DataMart\_KepuasanPelanggan adalah subset warehouse yang fokus pada analisis loyalitas dan kepuasan pelanggan. Disusun untuk mempermudah akses cepat oleh user tanpa query kompleks.

### 3.2.3 Design and implement partitioned tables and views

Untuk memastikan performa dan skalabilitas sistem data warehouse seiring dengan pertumbuhan data historis, proyek ini menerapkan partisi tabel dan view yang terindeks. Pendekatan ini memungkinkan eksekusi kueri yang lebih cepat dan manajemen data historis yang efisien, terutama untuk data skala besar seperti pengiriman kendaraan per tahun.

#### a. Partisi Tabel Fakta

Partisi dilakukan secara horizontal (row partitioning) pada tabel Fakta\_Pengiriman\_Kendaraan berdasarkan atribut delivery\_year, yang berasal dari kolom

delivery\_date. Ini mempermudah pelacakan dan agregasi berdasarkan waktu. Tujuan Partisi ini untuk :

- Mempercepat kueri waktu-spesifik seperti analisis tahunan.
- Mempermudah manajemen historis (arsip dan pemeliharaan).
- Mendukung optimalisasi performa saat digunakan bersama dengan clustered index.

```
1. -- Membuat fungsi partisi berdasarkan tahun
2. CREATE PARTITION FUNCTION pf_DeliveryYear (INT)
3. AS RANGE LEFT FOR VALUES (2018, 2020, 2022, 2024);
4.
5. -- Skema partisi
6. CREATE PARTITION SCHEME ps_DeliveryYear
7. AS PARTITION pf_DeliveryYear ALL TO ([DW_Fakta]);
8.
9. -- Menerapkan indeks terpartisi pada tabel fakta
10. CREATE CLUSTERED INDEX idx_delivery_date
11. ON Fakta_Pengiriman_Kendaraan (delivery_date)
12. ON ps_DeliveryYear (delivery_year);
```

Penjelasan :

- Data dikelompokkan ke dalam partisi berdasarkan delivery\_year.
- Partisi dikombinasikan dengan clustered index untuk mempercepat pencarian berdasarkan tanggal.
- DW\_Fakta digunakan sebagai target filegroup untuk partisi, sesuai dengan struktur storage.

b. Perancangan Indexed Views (Zona Analitik/Gold Layer)

Untuk mendukung agregasi yang sering digunakan dalam laporan dan analisis dashboard, sistem menggunakan indexed views yang menggabungkan fakta dan dimensi, serta sudah terindeks agar siap pakai.

- View Agregat : Total kendaraan per Model per Tahun

```
1. CREATE VIEW vw_TotalKendaraanPerTahun
2. WITH SCHEMABINDING AS
3. SELECT
4.     m.model,
5.     w.year,
6.     SUM(f.units_delivered) AS total_units
7. FROM dbo.Fakta_Pengiriman_Kendaraan f
8. JOIN dbo.Dim_Model m ON f.model_id = m.model_id
9. JOIN dbo.Dim_Waktu w ON f.waktu_id = w.waktu_id
10. GROUP BY m.model, w.year;
11.
12. CREATE UNIQUE CLUSTERED INDEX idx_units_per_year
13. ON vw_TotalKendaraanPerTahun (model, year);
```

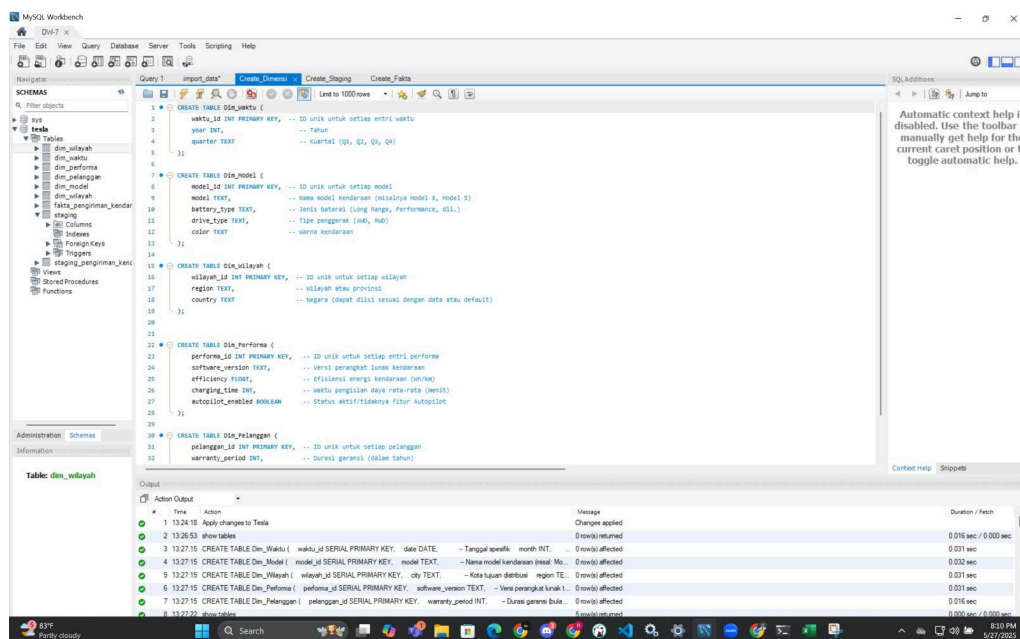
Penjelasan :

- View ini merangkum jumlah kendaraan yang dikirim per model setiap tahun.
- WITH SCHEMABINDING diperlukan untuk indexed views.
- View ini ditempatkan di Tier 3 untuk akses cepat oleh pelaporan bisnis.

## 4. Implementasi

### 4.1 Proses

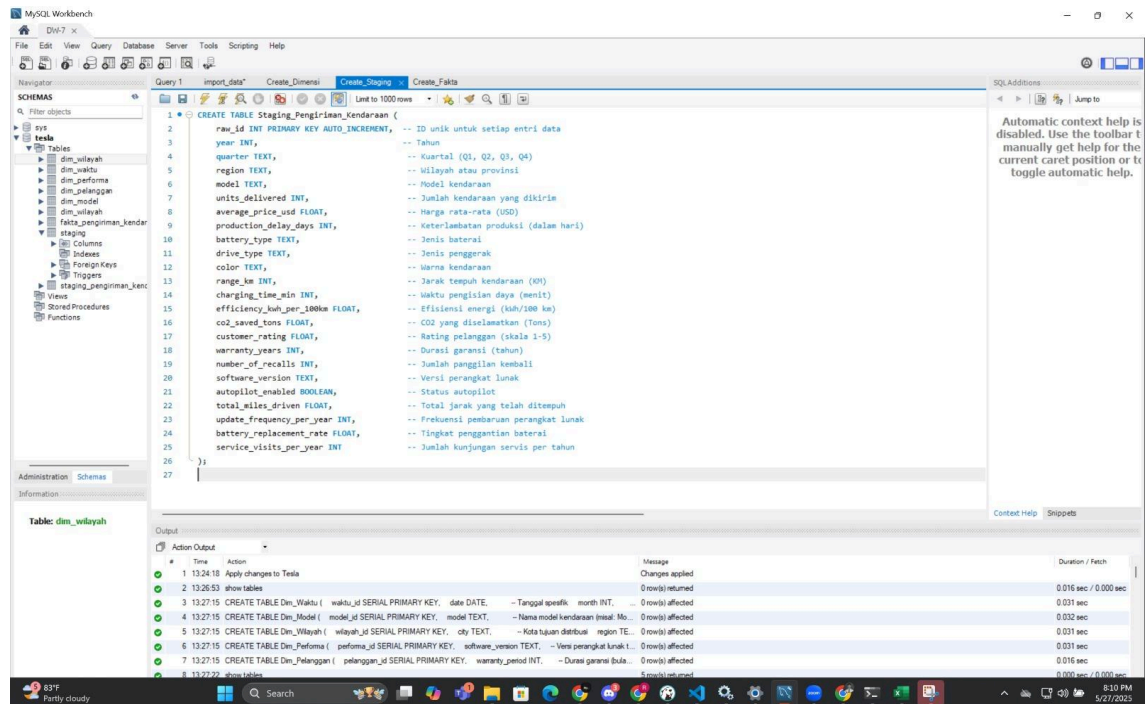
#### 1. Membuat Tabel Dimensi



Tampilan gambar di atas merupakan query yang digunakan untuk membuat setiap tabel dimensi. Karena menggunakan skema bintang (*star-scheme*), tabel-tabel dimensi ini akan memiliki keterhubungan berupa dengan mengelilingi Tabel Fakta. Dalam hal ini, dibuat 5 tabel dimensi yang meliputi :

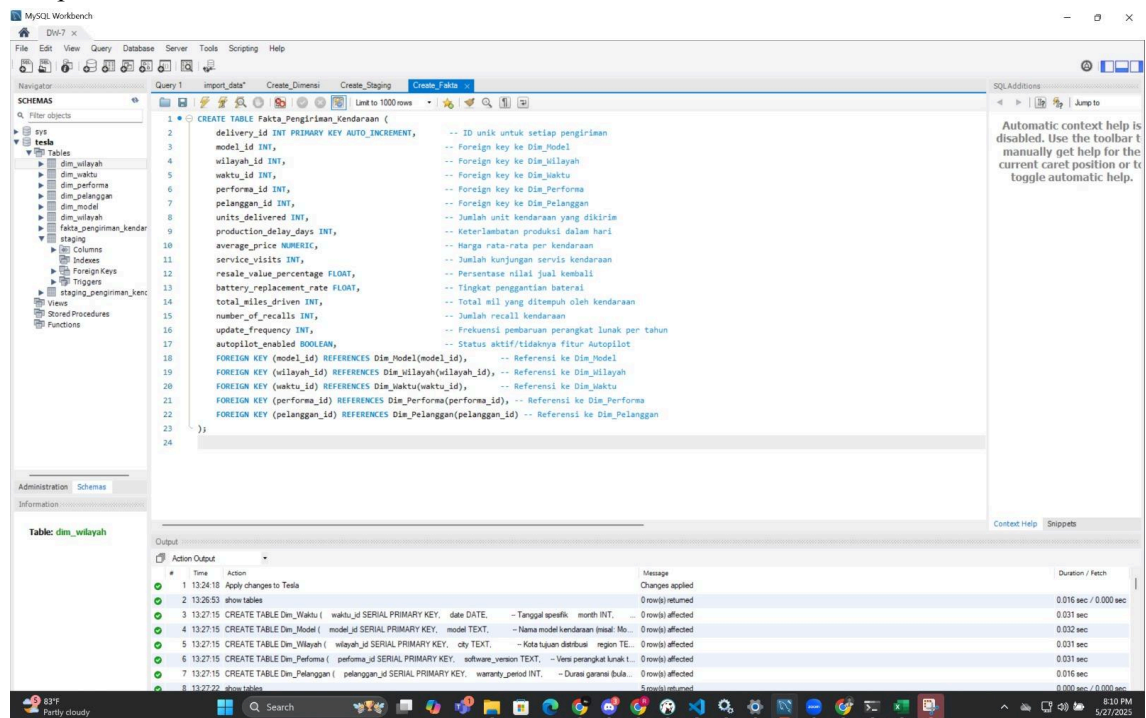
- Dim\_Waktu : tabel dimensi waktu
- Dim\_Model : tabel dimensi Model
- Dim\_Wilayah : tabel dimensi Wilayah
- Dim\_Performa : tabel dimensi Performa
- Dim\_Pelanggan : tabel dimensi Pelanggan

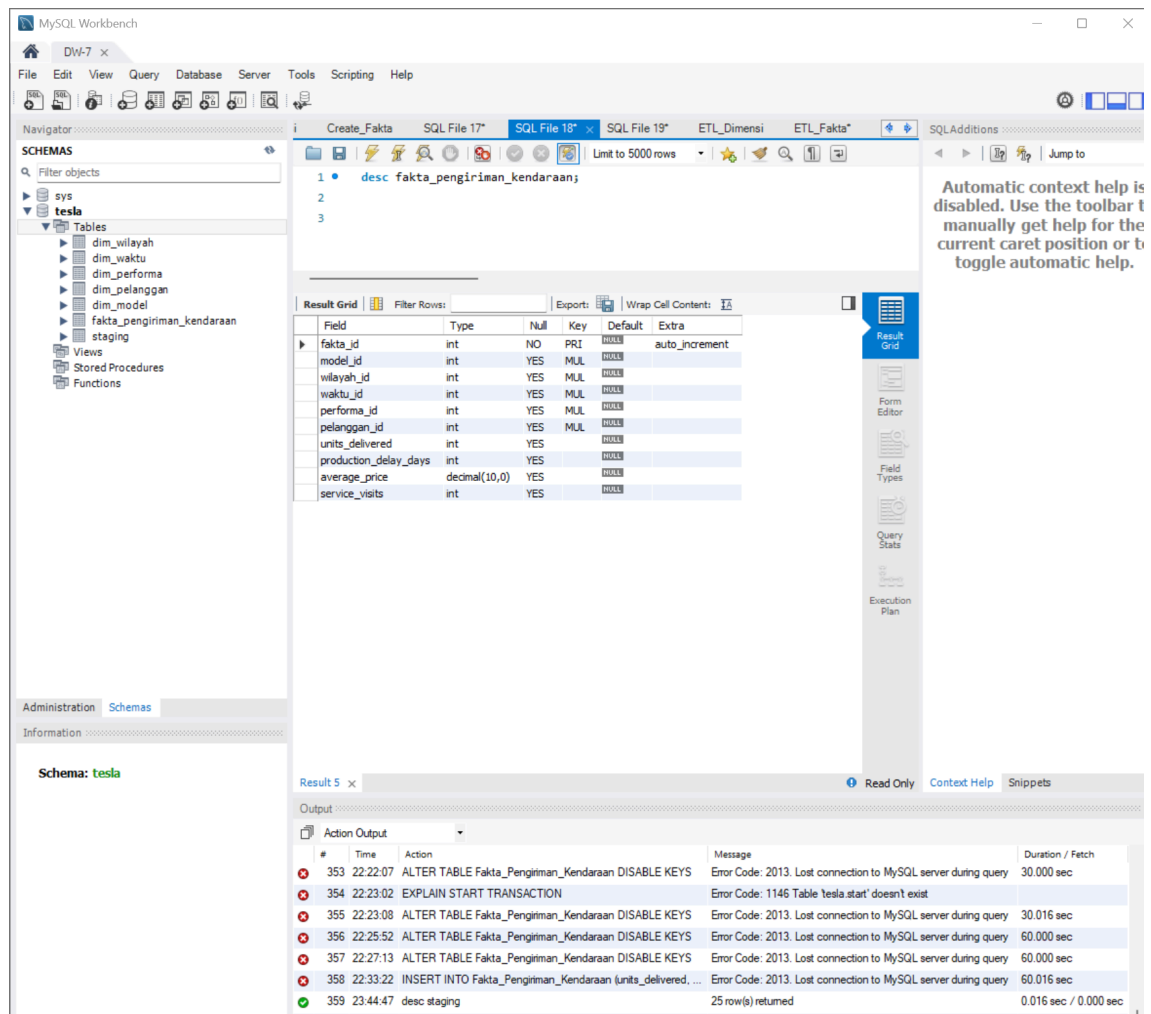
## 2. Buat Staging



Proses yang terjadi dalam staging area meliputi beberapa tahapan penting. Tahap ekstraksi melibatkan pengambilan data dari berbagai sistem sumber dengan format dan struktur yang berbeda-beda. Selanjutnya dilakukan pembersihan data untuk menangani missing values, duplikasi, dan inkonsistensi format. Transformasi data kemudian dilakukan untuk mengkonversi data ke format yang sesuai dengan skema data warehouse, termasuk perhitungan kolom derived dan agregasi awal.

## 3. Tampilan Tabel Fakta





Tabel Fakta merupakan komponen penting dalam skema data warehouse, karena menjadi sentra keterhubungan dari setiap tabel dimensi yang sudah dibuat. Dalam hal ini, tabel fakta dibuat dengan nama tabel “Fakta\_Pengiriman\_Kendaraan” yang memiliki banyak metrik di dalamnya. Metrik-metrik tersebut diantaranya :

- Units\_delivered : Jmlah unit kendaraan yang dikirim
- productions\_delay\_days : Keterlambatan produksi dalam hari
- average\_price : Harga rata-rata per kendaraan
- service\_visits : Jumlah kunjungan servis kendaraan
- resale\_value\_percentage : Persentase nilai jual kembali
- Battery\_replacement\_rate : Tingkat penggantian baterai
- Total\_miles\_driven : Total mil yang ditempuh oleh kendaraan
- number\_of\_recalls : Jumlah recall kendaraan
- update\_frequency : Frekuensi Pembaruan perangkat lunak per tahun

#### 4. Tampilan Staging

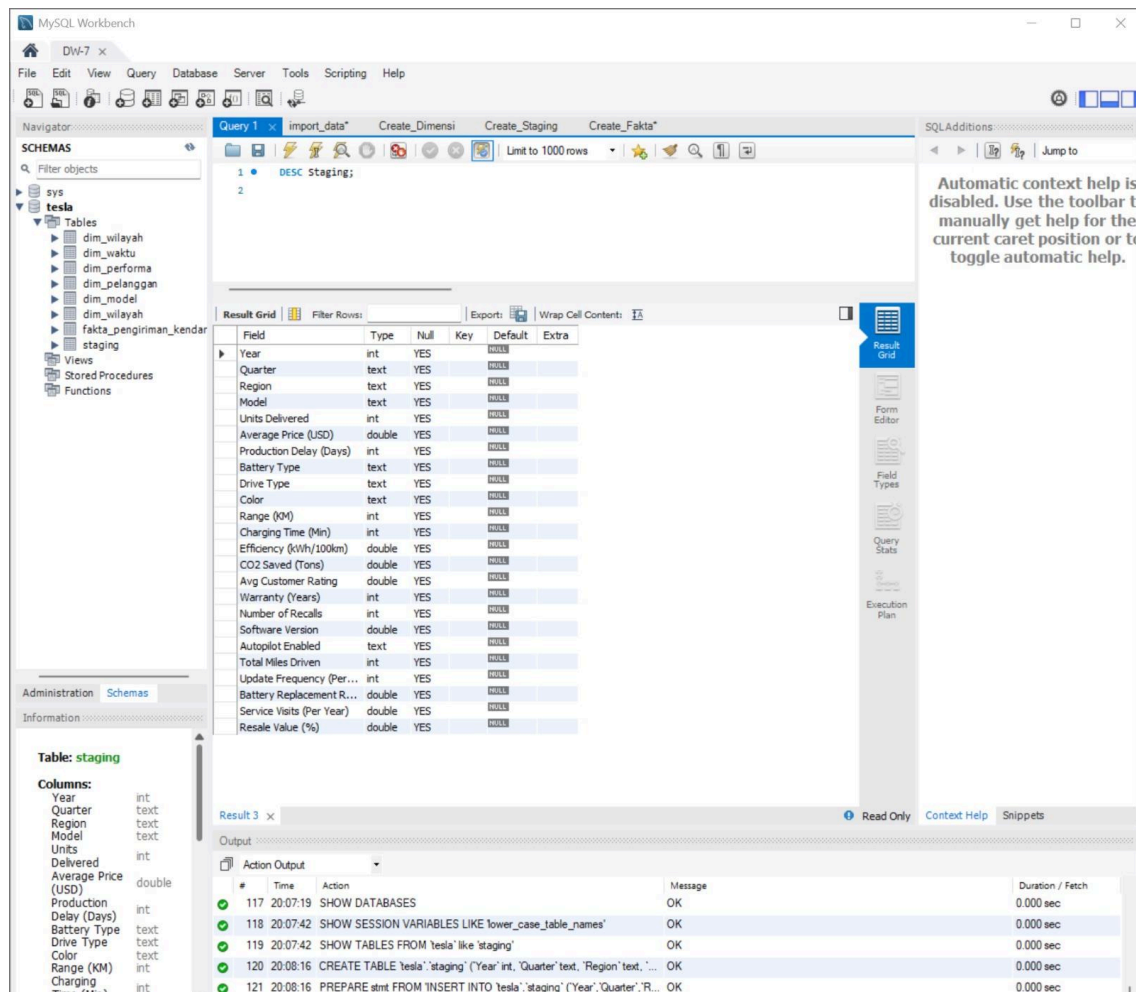
The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the 'tesla' database schema, including tables like 'dim\_wilayah', 'dim\_waktu', 'dim\_performa', 'dim\_pelanggan', 'dim\_model', 'fakta\_pengiriman\_kendaraan', 'staging', 'Views', 'Stored Procedures', and 'Functions'. The 'Information' pane shows the 'Schema: tesla'.

The central 'Result Grid' displays the structure of the 'staging' table. The table has the following columns:

Field	Type	Null	Key	Default	Extra
Year	int	YES		NULL	
Quarter	text	YES		NULL	
Region	text	YES		NULL	
Model	text	YES		NULL	
Units Delivered	int	YES		NULL	
Average Price (USD)	double	YES		NULL	
Production Delay (Days)	int	YES		NULL	
Battery Type	text	YES		NULL	
Drive Type	text	YES		NULL	
Color	text	YES		NULL	
Range (KM)	int	YES		NULL	
Charging Time (Min)	int	YES		NULL	
Efficiency (kWh/100km)	double	YES		NULL	
CO2 Saved (Tons)	double	YES		NULL	
Customer Rating	double	YES		NULL	
Warranty (Years)	int	YES		NULL	
Number of Recalls	int	YES		NULL	
Software Version	double	YES		NULL	
Autopilot Enabled	text	YES		NULL	
Total Miles Driven	int	YES		NULL	
Update Frequency (Per...	int	YES		NULL	
Battery Replacement R...	double	YES		NULL	
Service Visits (Per Year)	double	YES		NULL	
Resale Value (%)	double	YES		NULL	
raw_id	int	NO	PRI	NULL	auto...

The 'Output' pane at the bottom shows the execution of a query. The query is 'desc staging;'. The output shows the table structure as described above. The message indicates '25 row(s) returned'.

Berikut merupakan tampilan hasil dari proses staging. Hasilnya adalah data yang telah dibersihkan dan divalidasi; data dengan struktur yang terstandarisasi, telah ditransformasi, dan sebagainya.



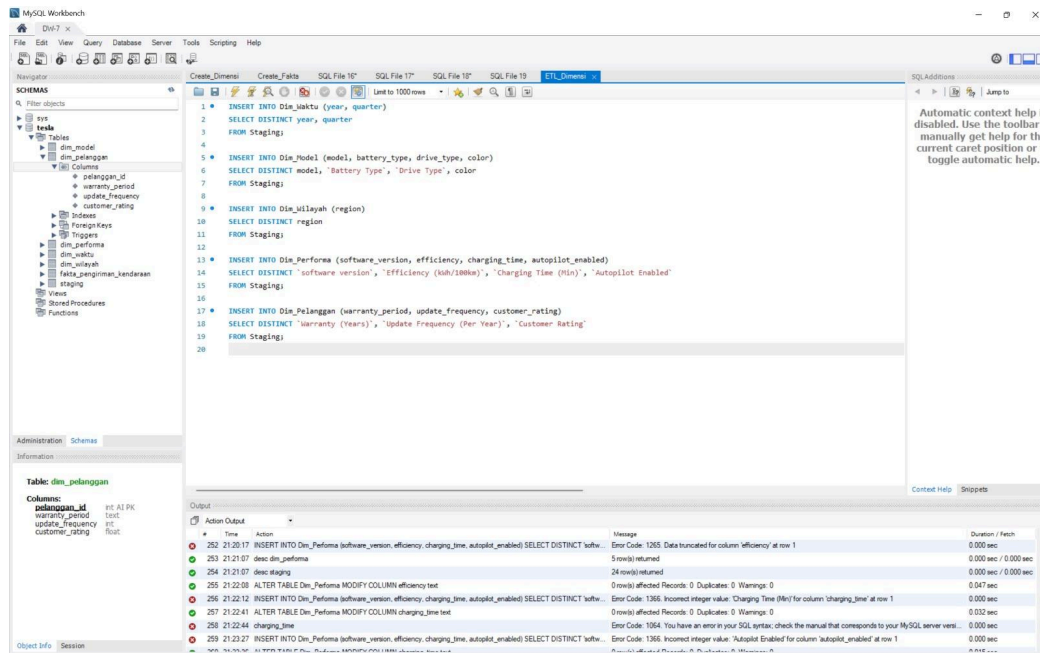
Deskripsi dari tahap staging ini terbagi menjadi beberapa bagian :

- Terdapat beberapa tabel dimensi, sesuai dengan yang telah dibuat sebelumnya (Dim\_Waktu, Dim\_Model, Dim\_Wilayah, Dim\_Performa, Dim\_Pelanggan) dan satu tabel fakta (fakta\_pengiriman kendaraan). Keterangan ini dapat dilihat dari tampilan di sebelah kiri-atas gambar. Tabel yang aktif saat gambar diatas diambil, adalah tabel “staging” .
- Berikutnya adalah isi dari tabel “staging“. Tabel ini memiliki terdiri dari beberapa kolom, yakni :
  - a. Field
  - a. Tipe Data
  - b. Keterangan

Disisi lain, beberapa baris yang dimuat adalah komponen-komponen yang sudah dimuat dalam tabel fakta dan tabel-tabel dimensi.

## 5. ETL Dimensi

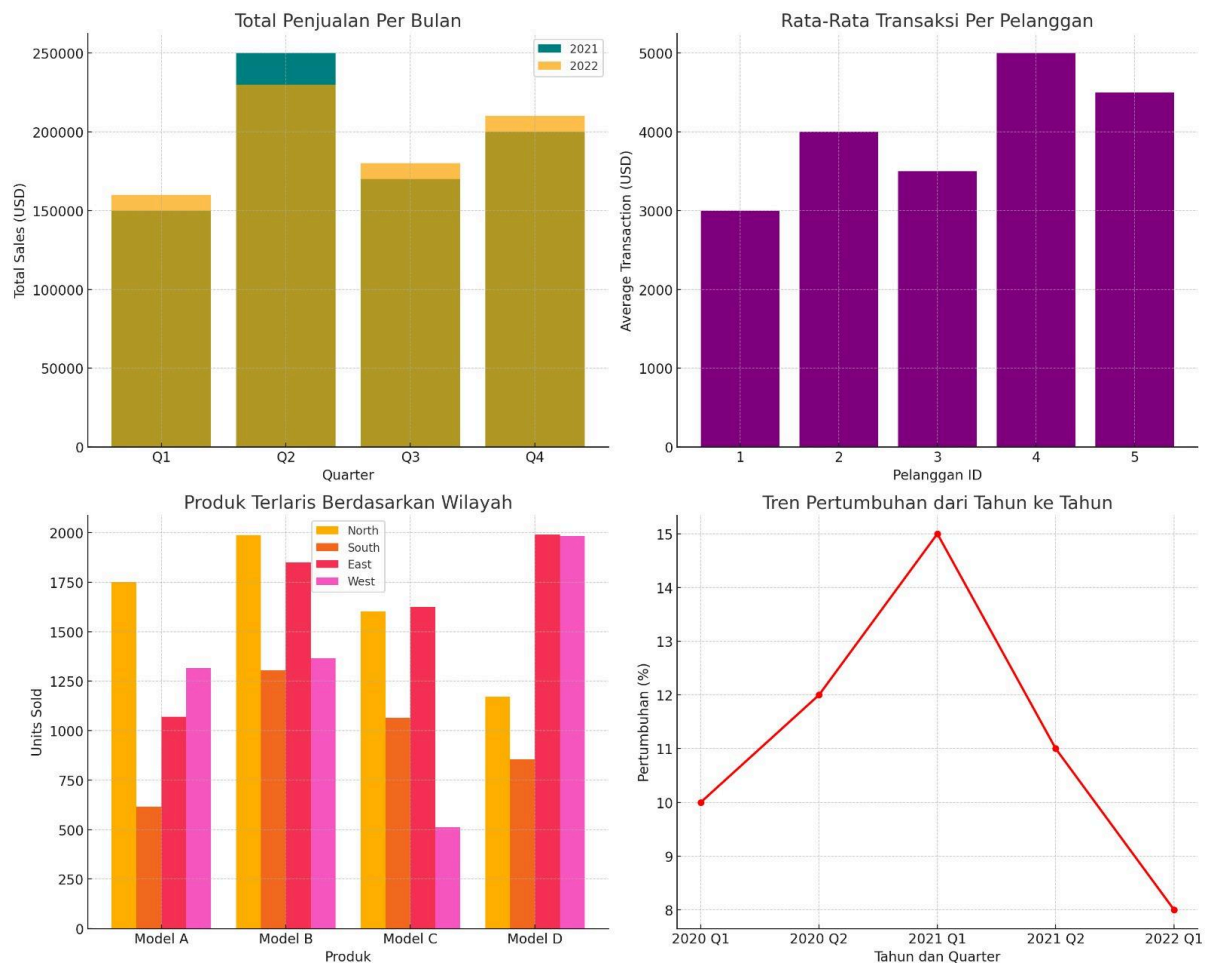




Analisis : menunjukkan aktivitas pengolahan data dalam MySQL Workbench, di mana sejumlah perintah SQL digunakan untuk mentransfer data dari tabel *Staging* ke beberapa tabel dimensi, seperti *dim\_waktu*, *dim\_model*, *dim\_wilayah*, *dim\_performa*, dan *dim\_pelanggan*. Setiap tabel dimensi dibuat dengan perintah *INSERT INTO* yang menggunakan data terpilih secara *distinct* dari tabel *Staging*. Pada panel bawah, terlihat beberapa pesan error, seperti "Data truncated for column 'efficiency'" dan masalah tipe data, yang menunjukkan bahwa ada ketidaksesuaian tipe data atau panjang data saat mencoba memasukkan data ke kolom tertentu. Hal ini menekankan pentingnya memastikan kecocokan tipe data antara sumber dan target selama proses ETL (Extract, Transform, Load).



## 4.2 Hasil



Hasil dari implementasi perancangan industri manufaktur ini, yaitu :

### 1. Total Penjualan Per Bulan

Dari grafik Total Penjualan Per Bulan, kita dapat melihat bahwa penjualan tertinggi terjadi pada Q2 2022, dengan total penjualan lebih dari 250,000 USD. Penjualan di Q1 2021 relatif lebih rendah dibandingkan dengan tahun berikutnya, tetapi masih menunjukkan angka yang solid, sementara Q3 dan Q4 2021 menunjukkan hasil yang cukup baik. Q2 2022 mengalami lonjakan penjualan yang signifikan, menunjukkan adanya peningkatan permintaan atau peluncuran produk yang lebih sukses pada kuartal tersebut.

### 2. Rata-Rata Transaksi Per Pelanggan

Grafik Rata-Rata Transaksi Per Pelanggan menunjukkan bahwa pelanggan dengan ID 4 memiliki transaksi rata-rata tertinggi, yaitu sekitar 5000 USD. Pelanggan lainnya, dengan ID 1 hingga 3, memiliki transaksi yang lebih rendah, tetapi tetap berada dalam kisaran 3000 hingga 4000 USD. Hal ini mungkin menunjukkan bahwa pelanggan dengan ID 4 mungkin lebih sering melakukan transaksi dalam jumlah besar atau membeli produk dengan harga lebih tinggi dibandingkan dengan yang lain. Pelanggan ID 5 juga menunjukkan transaksi yang relatif tinggi, namun tidak mencapai level pelanggan ID 4.

### 3. Produk Terlaris Berdasarkan Wilayah

Grafik Produk Terlaris Berdasarkan Wilayah memberikan wawasan yang jelas mengenai performa produk di wilayah yang berbeda. Model D terjual paling banyak di wilayah West, dengan total unit yang sangat tinggi dibandingkan produk lainnya. Model A menunjukkan penjualan yang cukup tinggi di wilayah North, sedangkan Model B dan Model C lebih terdistribusi merata di wilayah South dan East. Ini menunjukkan adanya preferensi wilayah terhadap produk tertentu dan pentingnya strategi pemasaran yang dapat menyesuaikan dengan selera pasar di masing-masing wilayah.

#### 4. Tren Pertumbuhan Tahun ke Tahun

Grafik Tren Pertumbuhan Tahun ke Tahun mengungkapkan bahwa penjualan mengalami pertumbuhan yang pesat di 2021 Q1, dengan tingkat pertumbuhan mencapai lebih dari 15%. Namun, pada tahun 2022 Q1, kita melihat penurunan dalam tingkat pertumbuhan penjualan, yang mungkin disebabkan oleh berbagai faktor seperti penurunan permintaan, kompetisi yang lebih ketat, atau masalah pasokan. Grafik ini memberikan gambaran bahwa meskipun ada lonjakan pada awal tahun 2021, ada fluktuasi yang perlu diwaspadai untuk menjaga momentum pertumbuhan di tahun-tahun berikutnya.

### 5. Evaluasi

Berdasarkan hasil implementasi awal sistem Data Warehouse, ditemukan beberapa permasalahan signifikan yang berdampak pada performa sistem:

- a. Ukuran Data yang Sangat Besar: Dataset yang digunakan memiliki volume data yang besar, sehingga mempengaruhi kecepatan pemrosesan, terutama saat melakukan query kompleks dan join antar tabel.
- b. Proses Join yang Lama: Join antar tabel fakta dan dimensi membutuhkan waktu yang lama karena tidak dioptimalkan dengan baik. Hal ini disebabkan oleh kurangnya indeks, struktur data yang belum dinormalisasi atau belum dinyatakan dalam bentuk bintang (star schema) yang efisien.
- c. Kurangnya Optimalisasi Query: Beberapa query tidak ditulis secara efisien, sehingga berdampak langsung pada performa eksekusi, terutama dalam analisis OLAP.
- d. Kurangnya Teknologi Pendukung Big Data: Sistem masih menggunakan RDBMS biasa, yang tidak dirancang untuk menangani skala data warehouse yang besar.

### 6. Rencana Pengembangan

Untuk mengatasi permasalahan yang ditemukan, berikut adalah rencana pengembangan sistem Data Warehouse:

1. Perancangan Ulang Skema Data Warehouse
  - Terapkan skema Star Schema atau Snowflake Schema untuk mengurangi kompleksitas join antar tabel.
  - Pisahkan tabel fakta dan tabel dimensi secara jelas.
2. Indexing dan Partisi
  - Terapkan index pada kolom-kolom yang sering digunakan dalam kondisi join dan filter.
  - Gunakan table partitioning untuk membagi data menjadi segmen-segmen berdasarkan waktu, wilayah, atau kategori tertentu.
3. ETL Optimization

- Optimalkan proses Extract, Transform, Load (ETL) dengan menggunakan batch processing dan teknik incremental load untuk mempercepat pemrosesan data baru.
4. Penggunaan Teknologi Big Data (Opsional)
    - Jika skala data terus membesar, pertimbangkan migrasi ke platform big data seperti Apache Hive, Apache Spark, atau menggunakan data lakehouse (misal Delta Lake dengan Databricks).
    - Gunakan distributed query engine seperti Presto atau Trino untuk mempercepat query analitik besar.
  5. Materialized View dan OLAP Cube
    - Buat materialized view untuk menyimpan hasil join atau agregasi yang sering digunakan, sehingga tidak perlu dihitung ulang setiap saat.
    - Gunakan OLAP cube untuk mendukung analisis multidimensi yang lebih cepat.
  6. Monitoring dan Tuning Berkala
    - Lakukan evaluasi performa secara berkala menggunakan tools monitoring (contoh: Query Profiler, EXPLAIN PLAN).
    - Lakukan tuning query dan perbaiki arsitektur sesuai kebutuhan.

## 7. Tim Proyek

Nama Kelompok	Status Keanggotaan	Tugas
Muhammad Rafif Rivaldi	Ketua	Koordinator proyek, penanggung jawab integrasi sistem, verifikasi ERD, desain logikal dan fisikal, serta penjadwalan tahapan misi.
Elia Meylani Simanjuntak	Anggota	Penanggung jawab desain konseptual, perancangan skema multidimensi (fakta dan dimensi), serta dokumentasi awal (misi 1 dan 2).
Sahid Maulana	Anggota	Bertanggung jawab pada implementasi database (SQL), pembuatan script ETL, transformasi data, dan partisi tabel.
Chalifia Wananda	Anggota	Bertugas pada dokumentasi akhir laporan, poster dan data mart, serta validasi visualisasi analitik.