

# **PENERAPAN DATA WAREHOUSE DALAM MENDUKUNG SEKTOR TRANSPORTASI DAN LOGISTIK**



## **Anggota:**

Lion Abdi Marga	121450047
Lia Alyani	121450138
Happy Syahrul Ramadhan	122450013
Eli Dwi Putra Berema	122450064
M. Deriansyah Okutra	122450101

**Program Studi Sains Data  
Fakultas Sains  
Institut Teknologi Sumatera  
Lampung Selatan  
2025**

## **BAB I**

### **Ringkasan Kebutuhan dari Misi**

Industri logistik memainkan peran vital dalam mendukung kelancaran rantai pasok global. Namun, perusahaan logistik modern menghadapi tantangan besar dalam mengelola data yang tersebar di berbagai sistem operasional seperti ERP, GPS, CRM, serta sistem pelaporan insiden. Fragmentasi data ini menyebabkan ketidakefisienan dalam analisis performa pengiriman, keterbatasan dalam perencanaan rute dan prediksi permintaan, serta peningkatan biaya operasional dan risiko keterlambatan. Permasalahan tersebut menunjukkan perlunya sistem yang mampu mengintegrasikan berbagai sumber data dan memberikan insight komprehensif terhadap proses logistik.

Sebagai solusi, proyek ini merancang dan mengimplementasikan sistem *Data Warehouse* khusus untuk industri logistik dengan pendekatan multidimensi berbasis *Star Schema*. Sistem ini dirancang dengan satu tabel fakta utama (fact\_shipment) yang terhubung dengan sejumlah tabel dimensi seperti dim\_vehicle, dim\_product, dim\_customer, dim\_route, dim\_incident, dim\_sortinghub, dan dim\_date. Tujuan utama dari sistem ini adalah menyediakan fondasi analitik yang terstruktur dan terintegrasi untuk mendukung pengambilan keputusan yang cepat, akurat, dan berbasis data.

Melalui integrasi data yang holistik, perusahaan dapat memperoleh visibilitas menyeluruh atas proses pengiriman, melakukan analisis prediktif dan historis guna mengantisipasi kebutuhan serta gangguan operasional, dan mengoptimalkan rute serta biaya berdasarkan evaluasi performa armada, insiden pengiriman, dan pola permintaan musiman. Dengan demikian, sistem *Data Warehouse* ini menjadi kebutuhan strategis yang krusial dalam meningkatkan efisiensi, akurasi, dan daya saing perusahaan logistik di era digital.

## **BAB II**

### **Tujuan dan Ruang Lingkup Sistem**

#### **2.1 Tujuan Sistem**

Tujuan utama dari sistem Data Warehouse yang dirancang dalam proyek ini adalah menyediakan platform analitik terpusat yang mampu mendukung kebutuhan bisnis dan operasional perusahaan logistik secara komprehensif. Sistem ini dirancang untuk:

1. Mengintegrasikan data operasional dari berbagai sumber seperti ERP (Enterprise Resource Planning), GPS & IoT, CRM (Customer Relationship Management), dan sistem pelaporan insiden, guna menghilangkan fragmentasi data yang selama ini menjadi hambatan utama dalam proses analitik.
2. Mendukung pengambilan keputusan strategis berbasis data, baik oleh manajemen eksekutif maupun tim operasional, melalui visualisasi performa pengiriman, analisis biaya, serta evaluasi efisiensi rute dan armada.
3. Meningkatkan efisiensi operasional, dengan memungkinkan analisis multidimensi terhadap pengiriman barang, keterlambatan, insiden, dan biaya logistik.
4. Menyediakan kemampuan prediktif, untuk memperkirakan permintaan berdasarkan tren historis dan pola musiman, serta mengidentifikasi potensi risiko dalam rantai distribusi.
5. Memfasilitasi skalabilitas sistem analitik, agar perusahaan dapat terus mengembangkan sumber data dan indikator kinerja tanpa mengganggu arsitektur utama sistem.

#### **2.2 Ruang Lingkup Sistem**

Sistem Data Warehouse yang dikembangkan mencakup:

1. Pengumpulan dan integrasi data logistik dari sistem internal perusahaan, seperti data pengiriman, kendaraan, pelanggan, produk, hub sortir, dan insiden selama pengiriman.
2. Perancangan skema multidimensi berbasis *Star Schema*, dengan satu tabel fakta utama *fact\_shipment* dan tujuh tabel dimensi (*dim\_date*, *dim\_route*, *dim\_vehicle*, *dim\_product*, *dim\_customer*, *dim\_incident*, dan *dim\_sortinghub*) yang mencakup semua entitas penting dalam proses pengiriman.
3. Implementasi komponen teknis seperti indeksasi (menggunakan B-Tree), partisi data, dan optimasi penyimpanan dengan *columnstore indexing* pada tabel fakta serta *rowstore indexing* pada tabel dimensi.

4. Pengembangan proses ETL (Extract, Transform, Load) menggunakan SQL Server Integration Services (SSIS) untuk menjamin kualitas dan konsistensi data dari sumber mentah hingga siap analisis.
5. Penerapan arsitektur Medallion (Bronze–Silver–Gold Layer) untuk memastikan alur data yang terstruktur dan mendukung monitoring historis, pembersihan data, hingga penyajian business-ready data.

Ruang lingkup proyek dibatasi pada tahap integrasi data internal dan desain sistem untuk mendukung analisis logistik. Sistem tidak mencakup pengembangan aplikasi front-end atau dashboard BI, namun telah disiapkan agar dapat terintegrasi dengan berbagai platform visualisasi data seperti Power BI atau Tableau.

## **BAB III**

### **Metodologi**

#### **3.1 Arsitektur**

Dalam pengembangan sistem Data Warehouse untuk industri logistik ini, digunakan pendekatan *Medallion Architecture*, yaitu sebuah model bertahap yang membagi proses pemrosesan data ke dalam tiga lapisan: Bronze, Silver, dan Gold. Pendekatan ini memberikan struktur yang sistematis dalam pengelolaan dan transformasi data, dari sumber mentah hingga siap digunakan untuk analisis bisnis dan visualisasi.

##### **1. Bronze Layer**

Lapisan ini menyimpan data mentah secara apa adanya dari berbagai sumber seperti ERP, GPS, CRM, dan laporan insiden. Data pada layer ini belum dibersihkan dan digunakan sebagai referensi historis yang lengkap dan utuh.

##### **2. Silver Layer**

Data dari Bronze Layer diproses di Silver Layer untuk dibersihkan, distandarisasi, dan dinormalisasi. Tahapan ini mencakup penghapusan duplikasi, koreksi format, dan transformasi nilai menjadi bentuk konsisten yang dapat dianalisis.

##### **3. Gold Layer**

Gold Layer menyajikan data yang sudah terstruktur dalam bentuk Star Schema, terdiri dari satu tabel fakta (*fact\_shipment*) dan tabel-tabel dimensi (*dim\_vehicle*, *dim\_product*, *dim\_customer*, *dim\_route*, *dim\_incident*, *dim\_sortinghub*, dan *dim\_date*). Data pada lapisan ini siap digunakan oleh stakeholder untuk analisis, pelaporan, dan visualisasi melalui alat *Business Intelligence*.

## **BAB IV**

### **Analisis Kebutuhan**

Analisis kebutuhan dilakukan untuk memastikan sistem data warehouse yang dibangun mampu memenuhi kebutuhan strategis perusahaan logistik serta mampu mendukung performa teknis dan analitik yang optimal.

#### **4.1 Kebutuhan Teknis**

Sistem yang dibangun harus memenuhi persyaratan teknis berikut:

- Model skema multidimensi dengan struktur *star schema* untuk efisiensi query OLAP.
- Integrasi database SQL Server sebagai platform utama untuk penyimpanan dan pengelolaan data.
- Proses ETL (Extract, Transform, Load) untuk migrasi data dari sumber operasional ke dalam struktur warehouse.
- Query analitik OLAP untuk mendukung analisis performa pengiriman dan biaya logistik.
- Penerapan indeks pada primary key, foreign key, dan kolom agregasi.
- Partisi data berdasarkan waktu (date\_id) untuk efisiensi manajemen data historis.
- Kompresi data untuk menghemat ruang penyimpanan dan mempercepat proses baca.

#### **4.2 Kebutuhan Data**

Jenis-jenis data yang diperlukan:

- Data pengiriman: waktu, status, jarak, dan biaya pengiriman.
- Data kendaraan: tipe, kapasitas, usia, bahan bakar, dan status pemeliharaan.
- Data pelanggan: identitas, lokasi geografis, dan karakteristik demografis.
- Data produk: nama, kategori, berat, dimensi, dan sensitivitas.
- Data waktu: untuk analisis periodik dan tren temporal.
- Data insiden: jenis gangguan seperti keterlambatan atau kecelakaan.
- Data sorting hub: kapasitas, lokasi, dan jam operasional.

Setiap jenis data ini akan dimasukkan ke dalam tabel dimensi dan dikaitkan melalui foreign key ke tabel fakta.

#### **4.3 Kebutuhan Bisnis**

Data warehouse yang dikembangkan bertujuan untuk mendukung proses pengambilan keputusan secara cepat dan berbasis data. Kebutuhan bisnis yang dipenuhi antara lain:

- **Evaluasi performa pengiriman** berdasarkan rute, kendaraan, dan waktu untuk mengurangi keterlambatan.

- **Analisis biaya operasional**, khususnya efisiensi bahan bakar dan total pengeluaran per pengiriman.
- **Prediksi permintaan** berdasarkan tren historis untuk membantu perencanaan armada dan stok.
- **Segmentasi pelanggan** berdasarkan wilayah dan volume transaksi untuk penawaran yang lebih relevan.
- **Identifikasi rute atau produk bermasalah** yang sering menyebabkan insiden atau keterlambatan.
- **Pemantauan kapasitas hub sortir** untuk mencegah bottleneck dalam distribusi.

Sistem harus mampu menyajikan data secara agregatif dan visual (jika dikembangkan lebih lanjut), agar manajemen dapat menindaklanjuti temuan berdasarkan pola dan tren.

#### 4.4 Ringkasan Kebutuhan

Aspek	Kebutuhan
Skema	Star Schema (1 tabel fakta, 7 tabel dimensi)
Platform	SQL Server
Jenis Data	Pengiriman, Kendaraan, Produk, Pelanggan, Rute, Insiden, Waktu
Operasi Utama	ETL, Indexing, OLAP Queries
Fungsi	Evaluasi performa, efisiensi, segmentasi pelanggan, tren produk
Output	Query analitik, laporan agregasi, integrasi ke sistem BI

## **BAB V**

### **Desain Konseptual, Logikal, dan Fisikal**

#### **5.1 Desain Konseptual**

Desain konseptual bertujuan untuk menyediakan fondasi analitik yang terstruktur dan terintegrasi dengan merancang entitas-entitas dimensi utama yang merepresentasikan berbagai entitas bisnis. Pendekatan yang digunakan adalah skema bintang (*star schema*), di mana satu tabel fakta utama menghubungkan berbagai tabel dimensi.

Entitas Utama (Konsep Bisnis):

- Pengiriman (Shipment): Transaksi inti yang melacak pergerakan barang. Ini akan menjadi tabel fakta sentral.
- Waktu (Date): Dimensi waktu untuk menganalisis tren dan kinerja berdasarkan periode tertentu.
- Rute (Route): Dimensi yang merepresentasikan jalur pengiriman dari asal ke tujuan.
- Kendaraan (Vehicle): Dimensi yang berisi informasi tentang armada yang digunakan untuk pengiriman.
- Produk (Product): Dimensi yang mendeskripsikan barang yang dikirim.
- Pelanggan (Customer): Dimensi yang berisi data pelanggan penerima barang.
- Insiden (Incident): Dimensi untuk mencatat kejadian tak terduga selama pengiriman.
- Pusat Sortir (Sorting Hub): Dimensi yang mewakili fasilitas penyortiran dalam rantai distribusi.

Hubungan Antar Entitas:

- Setiap Pengiriman terjadi pada Waktu tertentu, menggunakan Kendaraan tertentu, melalui Rute tertentu, untuk mengirimkan Produk tertentu kepada Pelanggan tertentu.
- Sebuah Pengiriman dapat melewati sebuah Pusat Sortir.
- Sebuah Insiden dapat terjadi selama Pengiriman.
- Semua entitas dimensi (Waktu, Rute, Kendaraan, Produk, Pelanggan, Insiden, Pusat Sortir) akan terhubung ke entitas pusat (Pengiriman).

Tujuan dari skema ini adalah untuk mengoptimalkan pengambilan keputusan berbasis data terkait analisis kinerja pengiriman, pengelolaan armada, biaya operasional

#### **5.2 Desain Logikal**

Desain logikal menguraikan struktur tabel secara detail, termasuk nama kolom, tipe data, dan relasi antar tabel. Setiap tabel dimensi memiliki *primary key* yang menjadi identifikasi unik dari setiap entitas, dan dirujuk sebagai *foreign key* di tabel fakta `fact_shipment`.

Contohnya:



- vehicle\_id adalah primary key di dim\_vehicle dan menjadi foreign key di fact\_shipment.
- product\_id dari dim\_product juga menjadi foreign key di tabel fakta.

Relasi ini menciptakan keterkaitan yang erat antara fakta dan konteks dimensi, memungkinkan analisis menyeluruh lintas entitas.

Untuk memastikan performa sistem saat melakukan query berskala besar, dilakukan penambahan *index* pada kolom yang sering digunakan dalam operasi JOIN, WHERE, dan GROUP BY. Beberapa kolom yang diindeks antara lain:

- date\_id, route\_id, vehicle\_id, product\_id, customer\_id di fact\_shipment
- region pada dim\_customer
- product\_category pada dim\_product
- incident\_type pada dim\_incident

Implementasi fisik dilakukan dalam lingkungan SQL Server menggunakan perintah CREATE TABLE untuk mendefinisikan struktur tabel, tipe data, dan constraint. Tipe data ditentukan berdasarkan karakteristik kolom:

- INT untuk ID dan durasi
- VARCHAR untuk nama dan lokasi
- FLOAT untuk nilai numerik seperti biaya dan jarak
- BIT untuk data biner seperti sensitivitas produk

Contoh skrip fisik:

```
CREATE TABLE dim_route (
  route_id INT PRIMARY KEY,
  location_origin VARCHAR(100),
  location_destination VARCHAR(100),
  planned_distance FLOAT
);
```

Untuk optimasi penyimpanan dan performa kueri, diterapkan:

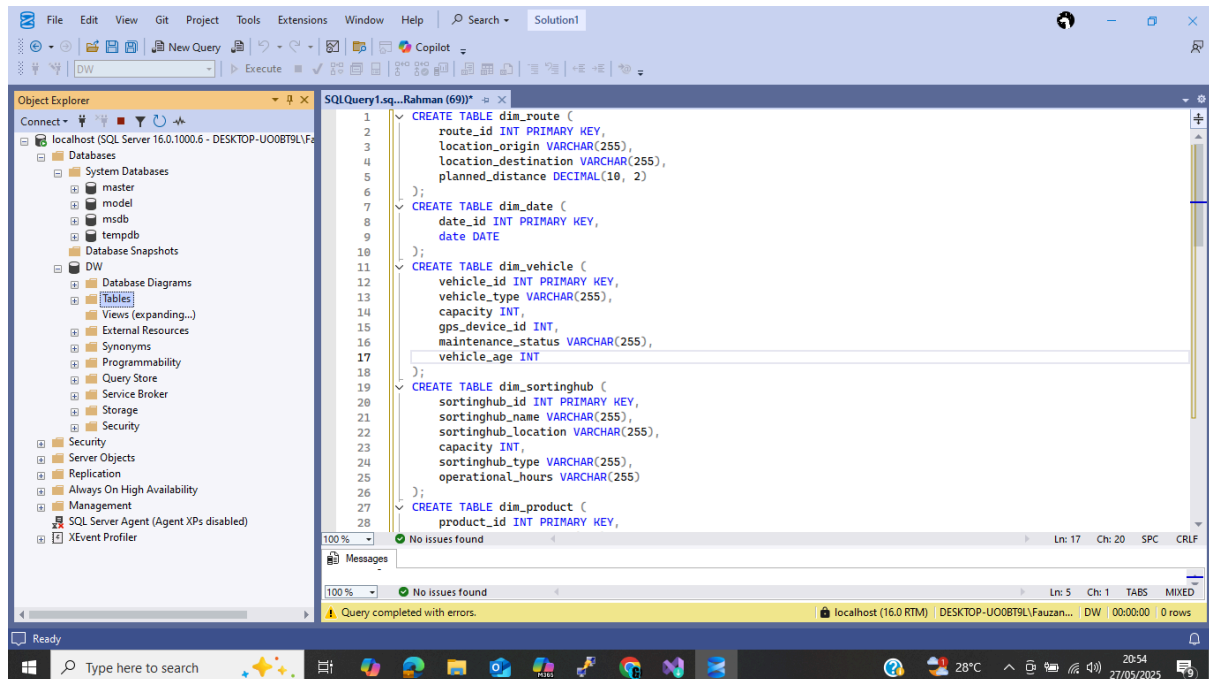
- Clustered Columnstore Index pada tabel fakta fact\_shipment, untuk mendukung operasi agregasi dan kompresi data dalam jumlah besar.
- Rowstore Clustered Index pada seluruh tabel dimensi, untuk mempercepat pencarian data kontekstual.
- Partisi horizontal pada tabel fact\_shipment berdasarkan date\_id, mendukung strategi manajemen data historis menggunakan *sliding window*.
- Kompresi data menggunakan metode PAGE atau COLUMNSTORE, mengurangi beban penyimpanan dan meningkatkan efisiensi pemrosesan data.

Pemuatan data dilakukan melalui kombinasi skrip SQL INSERT dan proses ETL dari file sumber CSV. Data yang dimuat telah melalui proses transformasi dan validasi sebelumnya agar sesuai dengan skema data warehouse.

## BAB VI

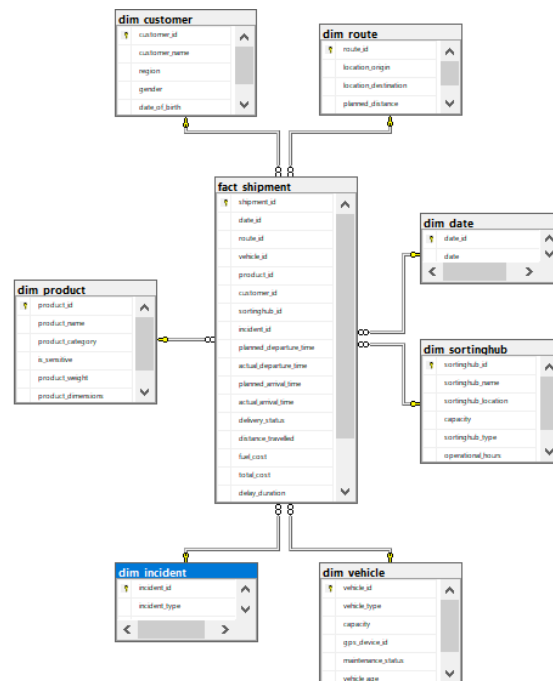
### Proses dan Hasil Implementasi

#### Implementasi skema menggunakan SQL Server SSMS



Implementasi skema data warehouse dilakukan menggunakan SQL Server Management Studio (SSMS), sebagaimana ditunjukkan pada gambar. Dalam proses ini, pengguna terhubung ke server lokal dan bekerja di dalam sebuah database bernama **DW**. Melalui jendela editor SQL, sejumlah perintah CREATE TABLE dijalankan untuk membangun tabel-tabel dimensi yang diperlukan dalam skema bintang (*star schema*). Tabel-tabel tersebut meliputi dim\_route, yang menyimpan informasi rute pengiriman seperti lokasi asal, tujuan, dan jarak rencana; dim\_date, yang menyimpan data waktu pengiriman; dim\_vehicle, yang berisi informasi kendaraan logistik termasuk tipe, kapasitas, dan status pemeliharaan; serta dim\_sortinghub, yang mendeskripsikan pusat sortir berdasarkan lokasi, kapasitas, jenis hub, dan jam operasional. Selain itu, proses pembuatan dim\_product juga sedang berlangsung pada bagian akhir skrip.

Setiap tabel dimensi didefinisikan dengan struktur kolom yang sesuai dan PRIMARY KEY sebagai identifikasi unik, yang nantinya akan dirujuk oleh tabel fakta. Seluruh skrip dijalankan tanpa kesalahan, sebagaimana terlihat dari status eksekusi pada bagian bawah SSMS yang menampilkan pesan “Query completed with no errors.” Hal ini menunjukkan bahwa struktur logikal dari data warehouse telah berhasil diimplementasikan secara fisik dalam sistem SQL Server. Dokumentasi visual ini sekaligus menjadi bukti bahwa proses awal pembuatan basis data telah dilakukan dengan benar dan siap dilanjutkan ke tahap pengisian data (ETL) dan penyusunan query analitik.



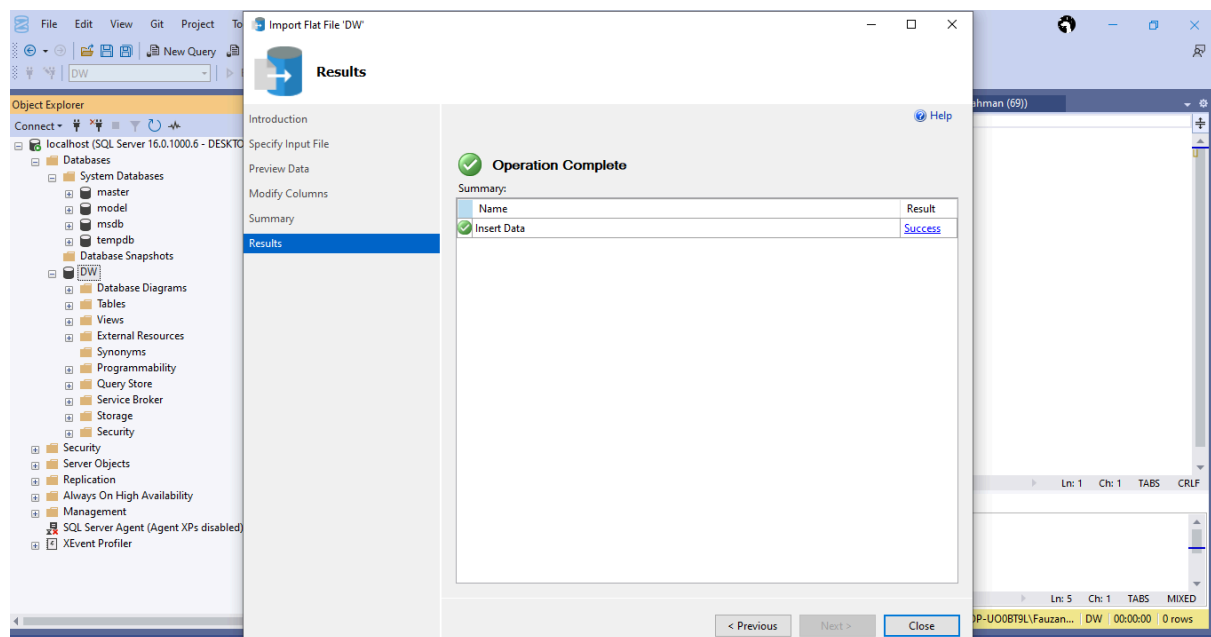
Gambar tersebut menampilkan struktur skema *star schema* yang telah berhasil diimplementasikan dalam SQL Server, dengan tabel fakta `fact_shipment` sebagai pusat dari model multidimensi. Tabel ini terhubung langsung ke tujuh tabel dimensi melalui relasi one-to-many, membentuk struktur yang memfasilitasi analisis logistik secara fleksibel dan efisien.

Dalam skema ini, `fact_shipment` menyimpan data transaksi pengiriman seperti waktu tempuh, biaya bahan bakar, total biaya, jarak yang ditempuh, serta waktu keberangkatan dan kedatangan, baik yang direncanakan maupun aktual. Tabel ini memiliki foreign key yang mengacu ke:

- **dim\_customer**: menyediakan informasi pelanggan, seperti nama, wilayah, jenis kelamin, dan tanggal lahir.
- **dim\_route**: menyimpan data lokasi asal dan tujuan pengiriman serta jarak yang direncanakan.
- **dim\_vehicle**: menyimpan detail kendaraan yang digunakan dalam proses pengiriman.
- **dim\_product**: berisi atribut produk yang dikirim, termasuk kategori dan dimensi produk.
- **dim\_incident**: merekam jenis insiden yang terjadi selama proses pengiriman.
- **dim\_sortinghub**: menyimpan data pusat sortir seperti lokasi, kapasitas, dan jenis hub.
- **dim\_date**: menyimpan informasi waktu yang digunakan untuk analisis musiman dan tren historis.

Relasi antar tabel diperlihatkan dengan jelas melalui garis penghubung antara foreign key di fact\_shipment dan primary key di masing-masing tabel dimensi. Visualisasi ini memperjelas bahwa model telah sesuai dengan prinsip *star schema*, di mana seluruh dimensi langsung berelasi ke satu pusat fakta, tanpa keterkaitan antar dimensi. Model ini memudahkan pelaksanaan query analitik seperti agregasi berdasarkan waktu, evaluasi biaya berdasarkan rute atau kendaraan, dan identifikasi tren pengiriman berdasarkan jenis produk atau insiden tertentu. Struktur ini tidak hanya mencerminkan pemahaman yang baik terhadap desain data warehouse, tetapi juga memastikan bahwa sistem siap untuk mendukung analisis operasional dan strategis dalam domain logistik.

## ETL Data Warehouse



Gambar ini menunjukkan salah satu tahap dari proses ETL (Extract, Transform, Load) dalam pembangunan data warehouse menggunakan SQL Server Management Studio (SSMS). Secara spesifik, gambar menampilkan hasil akhir dari proses **import file flat (teks) ke dalam database DW** yang berada di server lokal. Proses ini telah selesai dilakukan dengan status **"Operation Complete"**, dan langkah **Insert Data** tercatat berhasil (dengan status **Success**), menandakan bahwa data dari file sumber berhasil dimuat ke dalam tabel di database tujuan.

Di bagian kiri tampak struktur pohon objek dari SQL Server, memperlihatkan database bernama **DW** yang berisi beberapa elemen penting seperti Database Diagrams, Tables, Views, dan lainnya. Proses ETL ini kemungkinan dilakukan untuk memasukkan data dari sumber eksternal (misalnya file CSV atau TXT) ke dalam data warehouse agar dapat dianalisis lebih lanjut menggunakan skema bintang (star schema) seperti yang ditampilkan pada gambar sebelumnya.

Tahap ini merupakan bagian penting dalam pipeline ETL, yaitu **Load**, yang berfungsi untuk memindahkan data hasil ekstraksi dan transformasi ke dalam storage sistem data warehouse untuk digunakan dalam pelaporan dan analisis bisnis.

SQL Server Enterprise Edition (64-bit) - Solution1

Object Explorer: localhost (SQL Server 16.0.1000.6 - DESKTOP-...)

Query: SQLQuery3.sql...Rahman (521)\*

Results: 100% | select \* from fact\_shipment\_dummy

	shipment_id	date_id	route_id	vehicle_id	product_id	customer_id	sortinghub_id	incident_id	planned_departure_time	actual_departure_time	planned_arrival_time
1	1.00	20250418.00	4	V04	5.00	C004	SH03	I02	2025-04-18 06:00:00.0000000	2025-04-18 06:17:00.0000000	2025-04-18 10:00:00.0000000
2	2.00	20250417.00	2	V05	3.00	C003	SH01	I01	2025-04-17 06:00:00.0000000	2025-04-17 06:02:00.0000000	2025-04-17 12:00:00.0000000
3	3.00	20250426.00	2	V03	4.00	C002	SH04	I05	2025-04-26 06:00:00.0000000	2025-04-26 06:07:00.0000000	2025-04-26 12:00:00.0000000
4	4.00	20250421.00	2	V01	2.00	C001	SH01	I03	2025-04-21 06:00:00.0000000	2025-04-21 06:23:00.0000000	2025-04-21 11:00:00.0000000
5	5.00	20250410.00	1	V02	3.00	C003	SH01	I03	2025-04-10 06:00:00.0000000	2025-04-10 06:26:00.0000000	2025-04-10 12:00:00.0000000
6	6.00	20250424.00	1	V03	4.00	C004	SH02	I01	2025-04-24 06:00:00.0000000	2025-04-24 06:17:00.0000000	2025-04-24 09:00:00.0000000
7	7.00	20250406.00	1	V03	3.00	C001	SH01	I02	2025-04-06 06:00:00.0000000	2025-04-06 06:30:00.0000000	2025-04-06 09:00:00.0000000
8	8.00	20250420.00	4	V05	3.00	C001	SH02	I02	2025-04-20 06:00:00.0000000	2025-04-20 06:12:00.0000000	2025-04-20 10:00:00.0000000
9	9.00	20250425.00	3	V01	3.00	C001	SH03	I04	2025-04-25 06:00:00.0000000	2025-04-25 06:30:00.0000000	2025-04-25 09:00:00.0000000
10	10.00	20250412.00	5	V01	1.00	C002	SH02	I02	2025-04-12 06:00:00.0000000	2025-04-12 06:22:00.0000000	2025-04-12 11:00:00.0000000
11	11.00	20250409.00	4	V01	5.00	C001	SH04	I05	2025-04-09 06:00:00.0000000	2025-04-09 06:17:00.0000000	2025-04-09 09:00:00.0000000
12	12.00	20250408.00	1	V05	5.00	C001	SH02	I04	2025-04-08 06:00:00.0000000	2025-04-08 06:13:00.0000000	2025-04-08 10:00:00.0000000
13	13.00	20250411.00	4	V04	2.00	C004	SH04	I04	2025-04-11 06:00:00.0000000	2025-04-11 06:02:00.0000000	2025-04-11 09:00:00.0000000
14	14.00	20250430.00	4	V02	3.00	C001	SH03	I04	2025-04-30 06:00:00.0000000	2025-04-30 06:29:00.0000000	2025-04-30 12:00:00.0000000
15	15.00	20250402.00	2	V03	2.00	C003	SH01	I04	2025-04-02 06:00:00.0000000	2025-04-02 06:01:00.0000000	2025-04-02 10:00:00.0000000
16	16.00	20250410.00	2	V02	2.00	C001	SH02	I01	2025-04-10 06:00:00.0000000	2025-04-10 06:20:00.0000000	2025-04-10 12:00:00.0000000
17	17.00	20250427.00	4	V04	1.00	C001	SH01	I03	2025-04-27 06:00:00.0000000	2025-04-27 06:01:00.0000000	2025-04-27 11:00:00.0000000
18	18.00	20250403.00	3	V04	2.00	C003	SH02	I05	2025-04-03 06:00:00.0000000	2025-04-03 06:26:00.0000000	2025-04-03 12:00:00.0000000
19	19.00	20250417.00	5	V02	3.00	C003	SH01	I02	2025-04-17 06:00:00.0000000	2025-04-17 06:28:00.0000000	2025-04-17 09:00:00.0000000
20	20.00	20250405.00	4	V01	4.00	C001	SH01	I01	2025-04-05 06:00:00.0000000	2025-04-05 06:23:00.0000000	2025-04-05 12:00:00.0000000
21	21.00	20250405.00	2	V04	1.00	C005	SH03	I05	2025-04-05 06:00:00.0000000	2025-04-05 06:30:00.0000000	2025-04-05 11:00:00.0000000
22	22.00	20250418.00	3	V03	4.00	C005	SH02	I04	2025-04-18 06:00:00.0000000	2025-04-18 06:12:00.0000000	2025-04-18 10:00:00.0000000
23	23.00	20250415.00	4	V05	4.00	C002	SH03	I04	2025-04-15 06:00:00.0000000	2025-04-15 06:06:00.0000000	2025-04-15 11:00:00.0000000
24	24.00	20250420.00	3	V05	1.00	C004	SH01	I02	2025-04-20 06:00:00.0000000	2025-04-20 06:26:00.0000000	2025-04-20 12:00:00.0000000

Query executed successfully. | localhost (16.0 RTM) | DESKTOP-U008T9L\Fauzan... | DW | 00:00:00 | 1,000 rows

Export Data-tier Application 'DW'

Results

Introduction | Export Settings | Summary | Results

Operation Complete

Summary:

Name	Result
Exporting database	Success
Extracting schema	Success
Extracting schema from database	Success
Resolving references in schema model	Success
Validating schema model	Success
Validating schema model for data package	Success
Validating schema	Success
Processing tables for data extraction	Success
Exporting data from database	Success
Exporting data	Success
Processing Export.	Success
Processing Table [dbo].[dim_route].	Success
Processing Table [dbo].[dim_date].	Success
Processing Table [dbo].[dim_vehicle].	Success
Processing Table [dbo].[dim_sortinghub].	Success
Processing Table [dbo].[dim_product].	Success
Processing Table [dbo].[dim_customer].	Success
Processing Table [dbo].[dim_incident].	Success
Processing Table [dbo].[fact_shipment].	Success

## Agregasi Data

```
1  
2  SELECT customer_id, AVG(total_cost) AS avg_transaction  
3  FROM fact_shipment_dummy  
4  GROUP BY customer_id;  
5
```

100 % 4 0

Results Messages

	customer_id	avg_transaction
1	C003	633250
2	C001	621909
3	C002	613472
4	C004	600814
5	C005	649140

```
1  
2  SELECT customer_id, SUM(total_cost) AS SUM_transaction  
3  FROM fact_shipment_dummy  
4  GROUP BY customer_id;  
5
```

100 % 4 0

Results Messages

	customer_id	SUM_transaction
1	C003	121584114
2	C001	129357092
3	C002	138031362
4	C004	114755602
5	C005	119441764

```

1  SELECT customer_id,
2         MIN(total_cost) AS min_transaction,
3         MAX(total_cost) AS max_transaction
4  FROM fact_shipment_dummy
5  GROUP BY customer_id;
6

```

100 %

✖ 5

⚠ 0



Results



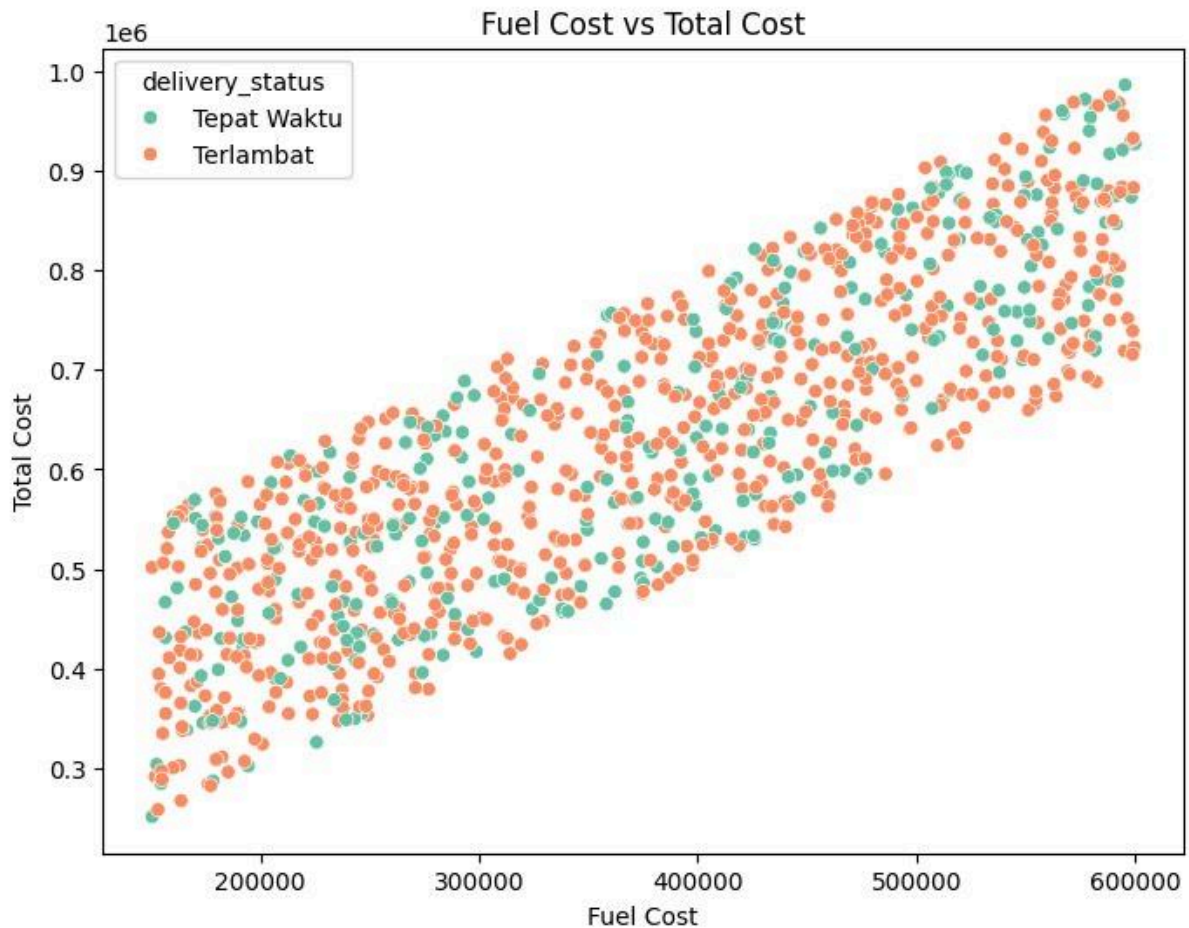
Messages

	customer_id	min_transaction	max_transaction
1	C003	258035	956103
2	C001	299935	975277
3	C002	283538	971988
4	C004	250968	986416
5	C005	284212	968905



## BAB VII

### Hasil Implementasi



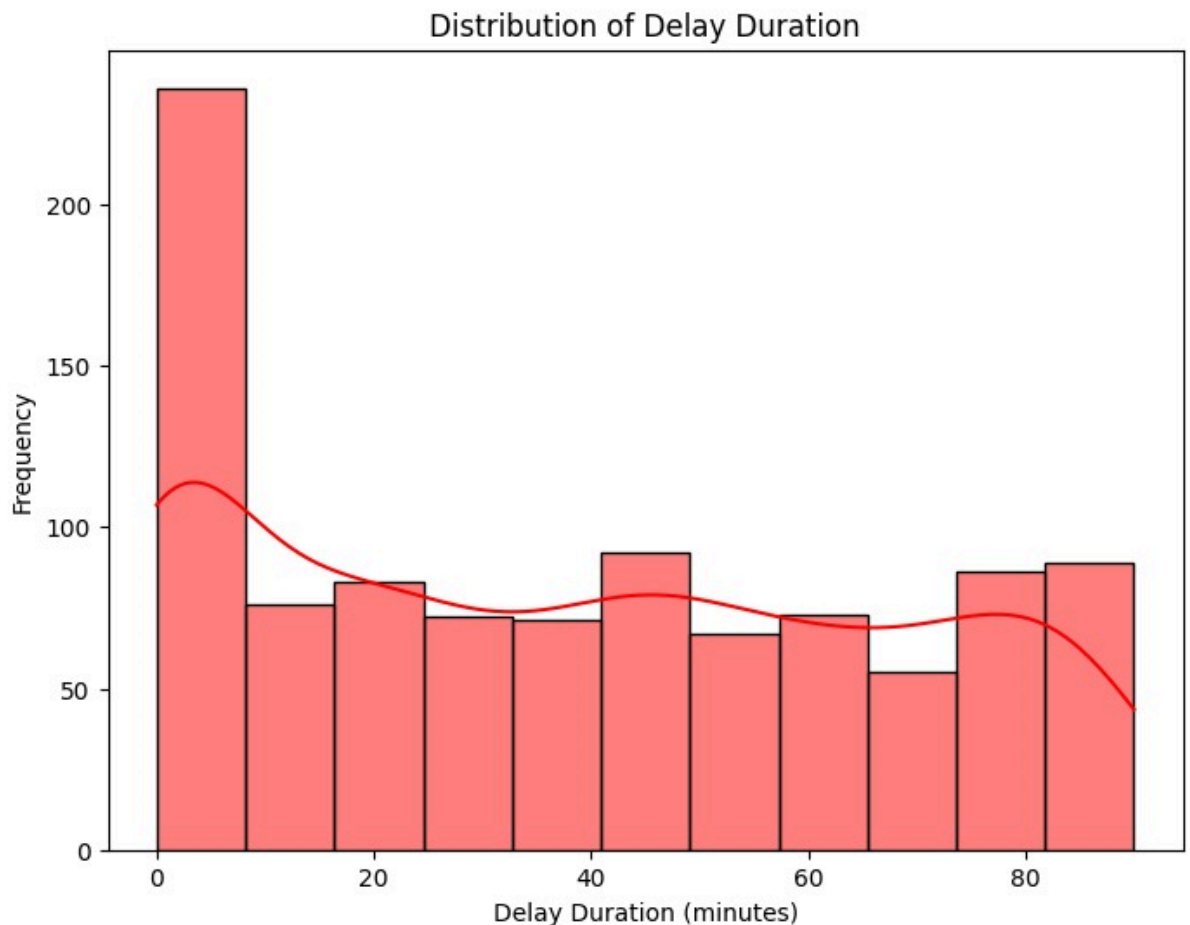
Grafik "Fuel Cost vs Total Cost" ini menunjukkan hubungan antara biaya bahan bakar dan total biaya pengiriman, dengan membedakan status pengiriman ("Tepat Waktu" dan "Terlambat").

Secara singkat, interpretasi dari grafik tersebut adalah:

- Ada korelasi positif antara biaya bahan bakar (Fuel Cost) dan total biaya (Total Cost). Artinya, secara umum, ketika biaya bahan bakar meningkat, total biaya pengiriman juga cenderung meningkat.
- Titik-titik data untuk pengiriman yang "Tepat Waktu" (warna hijau kebiruan) dan "Terlambat" (warna orange) tersebar di seluruh rentang biaya. Tidak ada pemisahan yang jelas yang menunjukkan bahwa keterlambatan secara signifikan mengubah hubungan dasar antara biaya bahan bakar dan total biaya, meskipun kedua jenis status

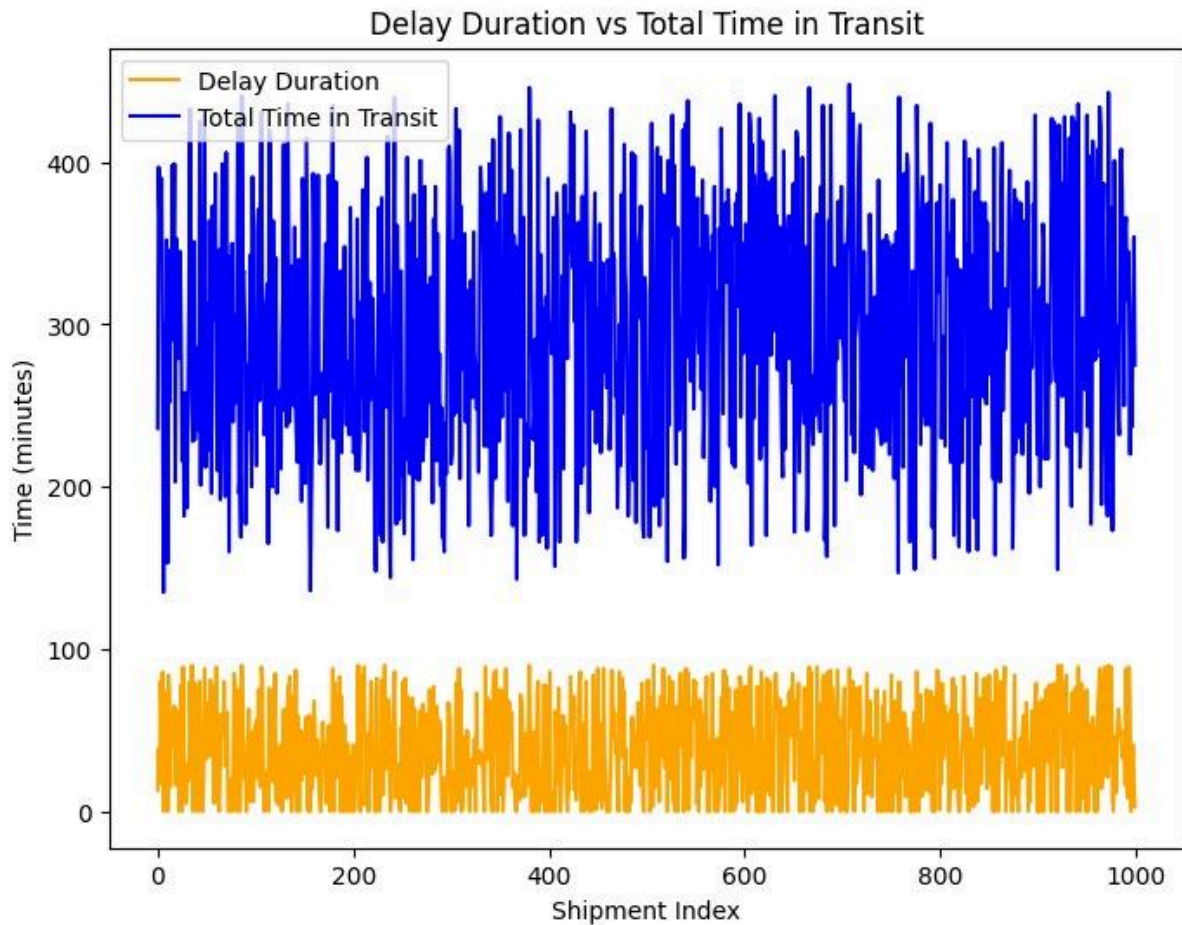
pengiriman menunjukkan tren peningkatan total biaya seiring dengan peningkatan biaya bahan bakar.

- Penyebaran titik menunjukkan bahwa faktor lain selain biaya bahan bakar juga mempengaruhi total biaya, karena pada tingkat biaya bahan bakar tertentu, masih terdapat variasi dalam total biaya baik untuk pengiriman yang tepat waktu maupun yang terlambat.



Garis merah pada grafik "Distribution of Delay Duration" tersebut kemungkinan besar adalah Kernel Density Estimate (KDE) atau Kurva Kepadatan (Density Curve).

Garis ini merupakan estimasi yang dihaluskan dari fungsi kepadatan probabilitas variabel acak (dalam hal ini, durasi keterlambatan). Tujuannya adalah untuk memberikan visualisasi yang lebih halus mengenai bentuk distribusi data dibandingkan dengan histogram saja.



**Skala Waktu yang Berbeda:** Terlihat jelas bahwa "Total Time in Transit" (waktu total perjalanan) secara konsisten jauh lebih besar nilainya dibandingkan dengan "Delay Duration" (durasi keterlambatan) untuk setiap pengiriman. Ini adalah hal yang wajar, karena durasi keterlambatan merupakan sebagian kecil dari keseluruhan waktu tempuh.

**Variabilitas pada Kedua Metrik:** Kedua garis menunjukkan fluktuasi yang signifikan di sepanjang indeks pengiriman. Ini mengindikasikan bahwa baik total waktu transit maupun durasi keterlambatan sangat bervariasi dari satu pengiriman ke pengiriman lainnya.

**Kontribusi Keterlambatan:** Meskipun durasi keterlambatan lebih kecil, puncaknya seringkali tampak berkorelasi dengan beberapa peningkatan pada total waktu transit, meskipun tidak selalu proporsional. Ini menunjukkan bahwa keterlambatan berkontribusi pada total waktu transit, namun total waktu transit juga dipengaruhi oleh faktor-faktor lain di luar keterlambatan yang tercatat.

**Pola Umum:** Tidak ada tren naik atau turun yang jelas pada kedua metrik seiring dengan bertambahnya indeks pengiriman. Keduanya tampak berfluktuasi di sekitar rata-rata masing-masing dengan varians yang cukup tinggi.

## **BAB VIII**

### **Evaluasi**

Evaluasi terhadap implementasi Data Warehouse dilakukan untuk mengukur kesesuaian sistem dengan tujuan bisnis yang telah dirumuskan serta penerapan konsep perancangan yang diusulkan. Secara umum, sistem berhasil dibangun dengan baik menggunakan SQL Server Management Studio (SSMS), di mana seluruh tabel dimensi dan tabel fakta berhasil dibuat tanpa kesalahan sintaks, menandakan struktur logikal telah diterjemahkan dengan benar ke dalam bentuk fisik. Proses ETL yang dilakukan mampu memindahkan data ke dalam sistem secara terstruktur, yang terbukti dari hasil query analitik dan visualisasi data yang dapat dijalankan dengan baik.

Dari sisi pencapaian tujuan bisnis, sistem telah mampu mengintegrasikan data dari berbagai sumber seperti ERP, CRM, GPS, dan pelaporan insiden melalui representasi tabel-tabel dimensi seperti `dim_vehicle`, `dim_customer`, `dim_product`, `dim_route`, dan `dim_incident`. Selain itu, kebutuhan untuk melakukan optimasi biaya dan rute telah didukung melalui data transaksi pada `fact_shipment` yang mencakup metrik penting seperti `total_cost`, `fuel_cost`, `delay_duration`, dan `distance_travelled`. Kemampuan analitik prediktif juga didukung oleh data historis yang lengkap dan terstruktur, memungkinkan perusahaan melakukan evaluasi dan peramalan berbasis waktu melalui `dim_date`.

Secara konseptual, sistem telah mengikuti pendekatan *Star Schema*, yang sesuai untuk analisis multidimensi dalam konteks logistik. Visualisasi data seperti hubungan antara `fuel_cost` dan `total_cost`, distribusi `delay_duration`, serta perbandingan antara `delay_duration` dan `total_time_in_transit` memberikan insight yang berguna dalam pengambilan keputusan operasional dan strategis. Dengan demikian, sistem Data Warehouse yang dibangun tidak hanya sesuai dengan perencanaan konseptual, tetapi juga telah mendukung pencapaian tujuan bisnis utama yang ditetapkan dalam proyek ini.