

LAPORAN *DATA WAREHOUSE*
**“Perancangan Data Warehouse untuk Analisis Preferensi
Konsumen Kendaraan di Industri Otomotif”**



Disusun Oleh :

Arafi Ramadhan Maulana	122450002
Rayan Koemi Karuby	122450038
Hermawan Manurung	122450069
Chevando Daffa Pramanda	122450095
Mirzan Yusuf Rabbani	122450118
Daffa Ahmad Naufal	122450137

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

2025

Laporan Akhir Proyek Data Warehouse PT Mobilita Nusantara

1. Ringkasan Proyek dan Latar Belakang

Proyek ini bertujuan utama untuk merancang dan mengimplementasikan sebuah solusi data warehouse yang komprehensif beserta kapabilitas analitik untuk PT Mobilita Nusantara. Perusahaan ini bergerak dalam sektor bisnis penjualan mobil bekas, di mana data memegang peranan penting dalam operasional dan pengambilan keputusan strategis.

Kehadiran data warehouse ini menjadi sangat signifikan bagi PT Mobilita Nusantara karena akan mengintegrasikan berbagai data operasional yang sebelumnya mungkin tersebar. Dengan data yang terpusat dan terstruktur, perusahaan dapat memperoleh wawasan mendalam mengenai aspek krusial seperti dinamika penjualan, manajemen stok unit mobil, serta profil dan preferensi pelanggan. Ini akan memberdayakan PT Mobilita Nusantara untuk beralih ke pengambilan keputusan yang berbasis data (data-driven decision-making), memungkinkan optimalisasi strategi penjualan, peningkatan efisiensi operasional, dan pada akhirnya, pertumbuhan profitabilitas di pasar mobil bekas yang kompetitif.

2. Tujuan dan Ruang Lingkup Sistem

Proyek ini bertujuan utama untuk mengembangkan data warehouse modern menggunakan SQL Server guna mengkonsolidasikan data penjualan PT Mobilita Nusantara. Implementasi ini akan memungkinkan pelaporan analitik yang mendalam untuk memahami preferensi konsumen, tren harga mobil bekas, dan performa dealer. Lebih lanjut, akan dikembangkan analitik berbasis SQL untuk menghasilkan wawasan terperinci yang mendukung pengambilan keputusan bisnis.

Ruang lingkup proyek ini mencakup implementasi skema data warehouse dengan tiga lapisan (Bronze, Silver, Gold) di platform SQL Server. Akan

ada proses ETL (Extract, Transform, Load) untuk memindahkan dan mentransformasi data dari sumber data berformat CSV ke dalam data warehouse. Selain itu, akan dilakukan pembuatan kueri-kueri analitik yang dirancang khusus untuk menjawab berbagai kebutuhan dan pertanyaan bisnis PT Mobilita Nusantara.

3. Metodologi

- **Bronze:** Data mentah dari file CSV, misalnya car_data.csv, disimpan di lapisan bronze Raw_Car_Sales_Transactions tanpa pembersihan awal. Prinsip data lineage dan auditabilitas didukung oleh prosedur ini.
- **Silver:** Data lapisan emas perunggu dibersihkan, distandarisasi, dan divalidasi dalam format terstruktur. Proses ETL ini menghasilkan tabel berwarna emas.Conformed_Pelanggan Salah satu jenis transformasi adalah mengubah format tanggal, standarisasi teks, menangani nilai NULL, dan deduplikasi data.
- **Gold:** Gold Layer adalah Analytical Data Mart yang dibangun dengan Star Schema. Terdapat di dalamnya: Tabel Fakta Penjualan, Tabel dimensi untuk Waktu, Pelanggan, Kendaraan, dan Penjual. Kebutuhan analitik dan pelaporan seperti dashboard penjualan dan analisis preferensi pelanggan dipenuhi oleh lapisan ini.

Tools yang Digunakan:

- SQL Server 2022 Developer Edition (atau versi yang Anda gunakan).

Digunakan untuk: SQL Server 2022 Developer Edition adalah platform utama untuk pengelolaan basis data relasional.Menyimpan semua informasi di tiga lapisan emas, perak, dan emas, menggunakan prosedur penyimpanan untuk menjalankan proses ETL (Extract, Transform, Load). Membangun Analytical Data Mart menggunakan struktur schema bintang.SQL Server sangat cepat, mendukung operasi skala besar, dan memiliki banyak fitur manajemen data.

- SQL Server Management Studio (SSMS).

SSMS, atau SQL Server Management Studio, adalah antarmuka

grafis (GUI) yang digunakan untuk berinteraksi langsung dengan SQL Server. Manfaat utamanya: menciptakan dan menjalankan query SQL. Mengembangkan dan mengawasi struktur tabel, skema, dan hubungan antar entitas. SSMS memudahkan debugging, visualisasi data, dan pengelolaan database dengan mengimplementasikan prosedur penyimpanan ETL dan memverifikasi hasil transformasi data.

- Microsoft Excel (atau spreadsheet software lain) untuk persiapan data awal. Pada awalnya, Microsoft Excel digunakan untuk: Menyiapkan dan menyimpan data penjualan mobil bekas yang telah dijual sebelumnya dalam bentuk spreadsheet. Mengubah data ke format Comma-Separated Values (CSV) untuk dimasukkan ke Bronze Layer. Meskipun tidak digunakan dalam proses analitik langsung, Excel sangat penting untuk menjamin kualitas dan struktur awal data yang akan diproses.
- Git & GitHub untuk manajemen versi kode.
Untuk kontrol versi dan kolaborasi tim, Git dan GitHub digunakan. Selama pengembangan proyek, Git membantu mencatat semua perubahan skrip SQL, instruksi, dan file konfigurasi. Repositori proyek GitHub termasuk folder emas/, perak/, emas/, dokumen/, dan script/. README.md GitHub juga berfungsi sebagai dokumentasi singkat tentang prosedur pelaksanaan dan arsitektur proyek.

Tahapan Proyek (Misi 1-4):

- Misi 1 - Analisis Kebutuhan
Menemukan kebutuhan bisnis untuk tiap departemen (pemasaran, penjualan, dealer, keuangan), dan kemudian membuat dimensi dan fakta berdasarkan data sebelumnya.
- Misi 2 - Desain Konseptual
Membangun skema multidimensi untuk mendukung analisis seperti tren penjualan, model favorit konsumen, dan performa dealer serta memastikan kesesuaian dengan sumber data.

- Misi 3 - Desain Logikal dan Fisik

Menciptakan skema konseptual dalam bentuk tabel relasional sambil menerapkan indexing, surrogate key, dan strategi penyimpanan optimal di SQL Server.

- Misi 4 - Implementasi dan Analitik

Eksekusi proses ETL berlapis menggunakan prosedur yang disimpan di masing-masing lapisan. Insight dari Gold Layer diperoleh melalui query OLAP. Laporan bisnis dan visualisasi menjawab pertanyaan analitis perusahaan.

4. Analisis Kebutuhan

PT Mobilita Nusantara, perusahaan nasional yang menjual mobil bekas melalui showroom dan platform digital AutoPasar, menghadapi tantangan dalam mengelola data penjualan yang tersebar dan tidak terintegrasi. Ini membuat pemantauan kinerja secara real-time menjadi sulit. Selain itu, perusahaan tidak memiliki metode untuk mengamati tren harga berdasarkan model dan tahun produksi, sehingga sulit untuk menentukan harga jual kembali yang ideal. Laporan preferensi pelanggan yang didasarkan pada fitur kendaraan juga tidak tersedia. Akibatnya, strategi stok dan pemasaran yang tidak tepat sasaran sering terjadi. Ketidaktahuan tentang preferensi konsumen lokal ini dianggap sebagai masalah yang sangat penting karena menyulitkan pembuatan strategi yang relevan di tiap wilayah.

Data warehouse ini dirancang dengan strategi Bottom-Up, membangun data mart spesifik seperti Data Mart Penjualan dan Preferensi Konsumen terlebih dahulu, yang kemudian akan dikonsolidasikan ke dalam Data Warehouse terpusat. Metode ini digunakan untuk memenuhi kebutuhan analisis khusus dengan cepat dan memungkinkan pengembangan bertahap.

5. Desain Konseptual, Logikal, dan Fisikal

Desain Konseptual

Desain konseptual Data Warehouse PT Mobilita Nusantara mengadopsi model Skema Bintang (Star Schema) yang telah terbukti efisien untuk analisis data dan pelaporan. Skema ini dirancang untuk memusatkan metrik penjualan sambil memberikan konteks yang kaya melalui tabel-tabel dimensi. Inti dari Skema Bintang ini adalah:

- Tabel Fakta Sentral: GOLD.FACT_CAR_SALES

Tabel ini menjadi jantung dari Data Warehouse, menyimpan metrik kuantitatif utama dari setiap transaksi penjualan mobil.

Atribut kunci:

- sales_id: Kunci unik untuk setiap penjualan.
- sales_price: Harga penjualan, metrik utama yang akan diagregasi.
- car_id_original: ID mobil asli dari sumber data untuk referensi.

Tabel fakta ini juga mengandung kunci asing (Foreign Keys) yang terhubung langsung ke setiap tabel dimensi, memungkinkan navigasi dan filter data yang cepat.

- Tabel Dimensi Kontekstual

Tabel dimensi menyediakan konteks deskriptif untuk data penjualan dalam tabel fakta. Mereka memungkinkan analisis dilakukan dari berbagai sudut pandang bisnis.

- GOLD.DIM_DATE:

- Menyediakan informasi temporal (waktu) terkait transaksi penjualan.
- Atribut: date, day, month, quarter, weekday. Ini memungkinkan analisis tren penjualan berdasarkan waktu (harian, bulanan, kuartalan, tahunan).

- GOLD.DIM_CUSTOMER:

- Menyimpan detail tentang pelanggan yang melakukan pembelian.

- Atribut: customer_key (Primary Key), annual_income, customer_name, customer_phone, gender. Memungkinkan segmentasi pelanggan dan analisis preferensi berdasarkan demografi.
- GOLD.DIM_DEALER:
 - Berisi informasi tentang dealer yang melakukan penjualan.
 - Atribut: dealer_key (Primary Key), dealer_name, dealer_number, dealer_region. Ini mendukung analisis performa dealer berdasarkan wilayah atau nama.
- GOLD.DIM_CAR:
 - Menyimpan detail mengenai kendaraan yang dijual.
 - Atribut: car_key (Primary Key), car_id, body_style, color, engine_type, model, transmission_type. Ini memungkinkan analisis popularitas mobil berdasarkan merek, model, atau fitur.

Dengan desain konseptual ini, PT Mobilita Nusantara akan memiliki fondasi yang kuat untuk pengambilan keputusan berbasis data yang akurat dan cepat dalam bisnis penjualan mobil bekas.

B. Desain Logikal dan Fisikal

Desain Data Warehouse PT Mobilita Nusantara dibangun di atas fondasi Skema Bintang yang kokoh pada lapisan logikal, menempatkan tabel fakta gold.fact_car_sales sebagai pusat data kuantitatif yang terhubung erat dengan dimensi-dimensi kontekstual seperti gold.dim_date, gold.dim_customer, gold.dim_dealer, dan gold.dim_car, semua dilengkapi dengan kunci surrogate dan tipe data yang presisi untuk menjamin integritas referensial serta efisiensi kueri multidimensional. Melengkapi aspek logikal ini, desain fisik difokuskan pada optimasi performa dan penyimpanan data bervolume tinggi, di mana implementasi indeks memainkan peran krusial. Diterapkan Clustered Index, Nonclustered Index pada kolom dimensi yang

sering difilter, serta Columnstore Index pada tabel fakta untuk mempercepat agregasi dan kompresi data.

Dari sisi strategi penyimpanan, kami mengadopsi arsitektur Hybrid Storage yang memadukan SSD untuk data aktif dan HDD untuk arsip, didukung oleh konfigurasi RAID (RAID 10 untuk data utama dan log, RAID 5 untuk backup) serta segregasi disk melalui filegroup terpisah guna mengelola beban kerja I/O secara efektif.

Lebih lanjut, tabel FACT_SALES dipartisi berdasarkan rentang tanggal menggunakan Partition Function dan Scheme, memungkinkan Partition Elimination dan Partition Switching yang secara signifikan mempercepat proses ETL dan kueri analitik historis, yang juga dapat disimulasikan melalui Partitioned View jika diperlukan, sehingga keseluruhan desain ini secara sinergis mendukung analisis data yang cepat, akurat, dan terukur bagi PT Mobilita Nusantara.

6. Proses Implementasi

- Lapisan Bronze

```
USE [pt-mobilita-nusantara];
GO

IF NOT EXISTS (SELECT * FROM sys.schemas WHERE
name = 'bronze')
BEGIN
    EXEC('CREATE SCHEMA bronze');
END
GO

IF OBJECT_ID('bronze.raw_car_sales_transactions', 'U') IS
NOT NULL
    DROP TABLE bronze.raw_car_sales_transactions;
```



```
GO
```

```
CREATE TABLE bronze.raw_car_sales_transactions (  
  Car_id VARCHAR(255),  
  [Date] DATE,  
  [Customer Name] VARCHAR(255),  
  Gender VARCHAR(50),  
  [Annual Income] VARCHAR(255),  
  Dealer_Name VARCHAR(255),  
  Company VARCHAR(255),  
  Model VARCHAR(255),  
  Engine VARCHAR(255),  
  Transmission VARCHAR(255),  
  Color VARCHAR(255),  
  [Price ($)] VARCHAR(255),  
  [Dealer_No] VARCHAR(255),  
  [Body Style] VARCHAR(255),  
  Phone VARCHAR(50),  
  Dealer_Region VARCHAR(255)  
);
```

```
GO
```

```
PRINT 'Tabel bronze.raw_car_sales_transactions berhasil  
dibuat/diperbarui.';
```

Tujuan dari skrip DDL ini adalah untuk membuat tabel dalam skema Bronze dalam data warehouse. Pertama, script memastikan bahwa skema Bronze ada; jika tidak ada, itu dibuat. Setelah itu, jika ada tabel bronze.raw_car_sales_transactions, script menghapusnya. Setelah itu, tabel baru dibuat dengan berbagai kolom untuk menyimpan informasi tentang transaksi penjualan mobil. Kolom-kolom ini termasuk Car_id, Tanggal, Nama

Pelanggan, dll. Pesanan konfirmasi akan dicetak setelah pembuatan tabel selesai. Data mentah transaksi mobil di lapisan Bronze disimpan dalam tabel ini.

- Lapisan Silver

```
USE [pt-mobilita-nusantara];
GO

IF NOT EXISTS (SELECT * FROM sys.schemas WHERE
name = 'silver')
BEGIN
    EXEC('CREATE SCHEMA silver');
END
GO

IF OBJECT_ID('silver.transformed_car_sales_transactions', 'U')
IS NOT NULL
    DROP TABLE silver.transformed_car_sales_transactions;
GO

CREATE TABLE silver.transformed_car_sales_transactions (
    Car_ID VARCHAR(255) PRIMARY KEY,
    Sales_Date DATE,
    Customer_Name VARCHAR(255),
    Gender VARCHAR(50),
    Annual_Income DECIMAL(18, 2),
    Dealer_Name VARCHAR(255),
    Car_Make VARCHAR(255),
    Car_Model VARCHAR(255),
```

```

Engine_Type VARCHAR(255),
Transmission_Type VARCHAR(255),
Car_Color VARCHAR(255),
Sales_Price DECIMAL(18, 2),
Dealer_Number VARCHAR(255),
Body_Style VARCHAR(255),
Customer_Phone VARCHAR(50),
Dealer_Region VARCHAR(255),
Sales_Year INT,
Sales_Month INT,
Sales_Day INT,
Last_Updated DATETIME DEFAULT GETDATE()
);
GO

SELECT 'Tabel silver.transformed_car_sales_transactions
berhasil dibuat dengan struktur yang benar.' AS Status;

```

Tujuan dari skrip ini adalah untuk membuat tabel di skema Silver. Pertama, skrip memeriksa apakah ada skema Silver, dan kemudian menghapus tabel transformed_car_sales_transactions jika ada. Tabel baru terdiri dari kolom yang menyimpan data transaksi penjualan mobil yang telah diproses, ini termasuk kolom seperti Car_ID, Sales_Date, dan Customer_Name, serta kolom tambahan untuk informasi terstruktur seperti Sales_Year, Sales_Month, dan Last_Updated. Kolom Car_ID berfungsi sebagai Primary Key, sementara beberapa kolom menggunakan tipe data DECIMAL untuk tingkat keakuratan yang lebih tinggi. Setelah itu, Skrip mencetak pesan untuk mengkonfirmasi bahwa tabel telah dibuat dengan sukses. Data transaksi yang telah diproses di lapisan emas disimpan dalam tabel ini.

- Lapisan Gold

```
USE [pt-mobilita-nusantara];
GO

IF NOT EXISTS (SELECT * FROM sys.schemas WHERE
name = 'gold')
BEGIN
    EXEC('CREATE SCHEMA gold');
END
GO

IF OBJECT_ID('gold.fact_car_sales', 'U') IS NOT NULL DROP
TABLE gold.fact_car_sales;
IF OBJECT_ID('gold.dim_date', 'U') IS NOT NULL DROP
TABLE gold.dim_date;
IF OBJECT_ID('gold.dim_car', 'U') IS NOT NULL DROP
TABLE gold.dim_car;
IF OBJECT_ID('gold.dim_dealer', 'U') IS NOT NULL DROP
TABLE gold.dim_dealer;
IF OBJECT_ID('gold.dim_customer', 'U') IS NOT NULL DROP
TABLE gold.dim_customer;
GO

CREATE TABLE gold.dim_date (
    date_key INT PRIMARY KEY,
    [date] DATE NOT NULL,
    year INT NOT NULL,
    month INT NOT NULL,
    day INT NOT NULL,
    weekday VARCHAR(20) NOT NULL,
    month_name VARCHAR(20) NOT NULL,
```

```

    quarter INT NOT NULL
);
GO
PRINT 'Tabel gold.dim_date berhasil dibuat/diperbarui.';

CREATE TABLE gold.dim_car (
    car_key INT PRIMARY KEY IDENTITY(1,1),
    car_id VARCHAR(255) NOT NULL UNIQUE,
    make VARCHAR(255),
    model VARCHAR(255),
    engine_type VARCHAR(255),
    transmission_type VARCHAR(255),
    color VARCHAR(255),
    body_style VARCHAR(255)
);
GO
PRINT 'Tabel gold.dim_car berhasil dibuat/diperbarui.';

CREATE TABLE gold.dim_dealer (
    dealer_key INT PRIMARY KEY IDENTITY(1,1),
    dealer_number VARCHAR(255) UNIQUE,
    dealer_name VARCHAR(255),
    dealer_region VARCHAR(255)
);
GO
PRINT 'Tabel gold.dim_dealer berhasil dibuat/diperbarui.';

CREATE TABLE gold.dim_customer (
    customer_key INT PRIMARY KEY IDENTITY(1,1),
    customer_name VARCHAR(255),
    gender VARCHAR(50),

```

```

    annual_income DECIMAL(18,2),
    customer_phone VARCHAR(50)
);
GO
PRINT 'Tabel gold.dim_customer berhasil dibuat/diperbarui.';

CREATE TABLE gold.fact_car_sales (
    sales_id INT PRIMARY KEY IDENTITY(1,1),
    car_id_original VARCHAR(255),
    date_key INT NOT NULL,
    car_key INT NOT NULL,
    dealer_key INT NOT NULL,
    customer_key INT NOT NULL,
    sales_price DECIMAL(18, 2),
    FOREIGN KEY (date_key) REFERENCES
gold.dim_date(date_key),
    FOREIGN KEY (car_key) REFERENCES
gold.dim_car(car_key),
    FOREIGN KEY (dealer_key) REFERENCES
gold.dim_dealer(dealer_key),
    FOREIGN KEY (customer_key) REFERENCES
gold.dim_customer(customer_key)
);
GO
PRINT 'Tabel gold.fact_car_sales berhasil dibuat/diperbarui.';
PRINT 'Struktur tabel Gold Layer berhasil dibuat/diperbarui.';

```

Skrip ini digunakan untuk membuat struktur tabel di skema Gold dalam data warehouse, yang bertujuan untuk menyimpan data yang telah diproses dan siap dianalisis lebih lanjut. Skrip dimulai dengan memeriksa apakah skema gold sudah ada, dan jika belum, skema tersebut akan dibuat. Setelah itu, skrip memeriksa dan menghapus

tabel-tabel yang sudah ada, seperti fact_car_sales, dim_date, dim_car, dim_dealer, dan dim_customer, untuk menghindari konflik saat membuat tabel baru. Selanjutnya, tabel-tabel dimensi dibuat, yaitu dim_date yang menyimpan informasi terkait tanggal, dim_car untuk data mobil, dim_dealer untuk informasi dealer, dan dim_customer untuk data pelanggan. Tabel utama atau tabel fakta, yaitu fact_car_sales, dibuat untuk menyimpan data penjualan mobil, yang menghubungkan informasi dari tabel-tabel dimensi dengan data penjualan, seperti harga jual. Tabel ini menggunakan kunci asing untuk mengaitkan data penjualan dengan tabel dimensi yang relevan. Setelah setiap tabel berhasil dibuat, skrip mencetak pesan konfirmasi untuk memastikan proses telah berhasil.

```
USE [pt-mobilita-nusantara];
GO

INSERT INTO bronze.raw_car_sales_transactions (
    Car_id,
    [Date],
    [Customer Name],
    Gender,
    [Annual Income],
    Dealer_Name,
    Company,
    Model,
    Engine,
    Transmission,
    Color,
    [Price ($)],
    [Dealer_No],
```

```

    [Body Style],
    Phone,
    Dealer_Region
)
SELECT
    [Car_id],
    [Date],
    [Customer Name],
    [Gender],
    [Annual Income],
    [Dealer_Name],
    [Company],
    [Model],
    [Engine],
    [Transmission],
    [Color],
    [Price ($)],
    [Dealer_No],
    [Body Style],
    [Phone],
    [Dealer_Region]
FROM
    ['Car Sales#xlsx - car_data$'];
GO

SELECT COUNT(*) AS TotalRowsLoadedInBronze FROM
bronze.raw_car_sales_transactions;
GO

SELECT TOP 10 * FROM bronze.raw_car_sales_transactions;
GO

```




Dengan menggunakan script SQL ini, data penjualan mobil dapat dimuat ke dalam tabel `bronze.raw_car_sales_transactions` di database `pt-mobilita-nusantara`. Langkah pertama adalah memilih database. Selanjutnya, kolom-kolom di sheet `car_data` dalam file Excel dicocokkan untuk memasukkan data ke dalam tabel. Perintah `SELECT COUNT(*)` digunakan untuk menghitung berapa banyak baris yang dimasukkan ke dalam tabel untuk memastikan bahwa data yang dimasukkan sesuai. Terakhir, perintah `SELECT TOP 10` menampilkan sepuluh baris pertama dari tabel untuk memastikan bahwa data yang dimasukkan sudah benar. Data mentah ditransfer ke dalam database untuk diproses lebih lanjut melalui prosedur ini.

- Silver Layer Load:

```
USE [pt-mobilita-nusantara];
GO

CREATE OR ALTER PROCEDURE
dbo.LoadSilverLayerCarSalesTransactions_V2
AS
BEGIN
    SET NOCOUNT ON;

    PRINT '--- Memulai Proses
LoadSilverLayerCarSalesTransactions_V2 ---';

    DECLARE @BronzeCount INT;
    SELECT @BronzeCount = COUNT(*) FROM
```

```

bronze.raw_car_sales_transactions;

    PRINT 'Jumlah baris di bronze.raw_car_sales_transactions: '
+ CAST(@BronzeCount AS VARCHAR(10));

    IF @BronzeCount = 0
    BEGIN
        PRINT 'PERINGATAN: Tidak ada data di
bronze.raw_car_sales_transactions. Proses pemuatan ke silver
dihentikan.';

        RETURN;
    END

    TRUNCATE TABLE
silver.transformed_car_sales_transactions;

    PRINT 'Tabel silver.transformed_car_sales_transactions telah
di-TRUNCATE.';

    PRINT 'Mencoba INSERT ke
silver.transformed_car_sales_transactions...';

    INSERT INTO silver.transformed_car_sales_transactions (
        Car_ID,
        Sales_Date,
        Customer_Name,
        Gender,
        Annual_Income,
        Dealer_Name,
        Car_Make,
        Car_Model,
        Engine_Type,
        Transmission_Type,

```

```

        Car_Color,
        Sales_Price,
        Dealer_Number,
        Body_Style,
        Customer_Phone,
        Dealer_Region,
        Sales_Year,
        Sales_Month,
        Sales_Day

    )

SELECT
    b.Car_id,
    b.[Date],
    b.[Customer Name],
    b.Gender,
    TRY_CONVERT(DECIMAL(18, 2),
REPLACE(REPLACE(b.[Annual Income], '$', ''), ',', '')),
    b.Dealer_Name,
    b.Company,
    b.Model,
    b.Engine,
    b.Transmission,
    b.Color,
    TRY_CONVERT(DECIMAL(18, 2),
REPLACE(REPLACE(b.[Price ($)], '$', ''), ',', '')),
    b.[Dealer_No],
    b.[Body Style],
    b.Phone,
    b.Dealer_Region,
    YEAR(b.[Date]),

```

```

MONTH(b.[Date]),
DAY(b.[Date])
FROM
    bronze.raw_car_sales_transactions AS b
WHERE

    b.Car_id IS NOT NULL AND b.Car_id != " AND b.[Date]
IS NOT NULL;

DECLARE @RowsLoaded INT;
SET @RowsLoaded = @@ROWCOUNT;
PRINT 'Jumlah baris yang berhasil dimasukkan ke
silver.transformed_car_sales_transactions: ' +
CAST(@RowsLoaded AS VARCHAR(10));

IF @RowsLoaded = 0 AND @BronzeCount > 0
BEGIN
    PRINT 'PERINGATAN PENTING: Tidak ada baris yang
dimuat ke silver, meskipun bronze memiliki data. Periksa
kondisi WHERE atau masalah konversi data yang mungkin
membuat semua baris gagal (meskipun TRY_CONVERT
seharusnya menghasilkan NULL).';
    PRINT 'Coba jalankan bagian SELECT dari INSERT
secara manual untuk melihat hasilnya.';
END
ELSE IF @RowsLoaded > 0
BEGIN
    PRINT 'BERHASIL: Data telah dimuat ke
silver.transformed_car_sales_transactions.';
END

```

```
PRINT '--- Proses LoadSilverLayerCarSalesTransactions_V2
Selesai ---';
END
GO

USE [pt-mobilita-nusantara];
GO
EXEC dbo.LoadSilverLayerCarSalesTransactions_V2;
```

Salah satu script SQL yang diberikan adalah prosedur tersimpan bernama "LoadSilverLayerCarSalesTransactions_V2." Tujuan prosedur ini adalah untuk memindahkan data penjualan mobil dari lapisan bronze ke lapisan emas dalam pipeline data. Untuk memastikan bahwa data tersedia, prosedur ini dimulai dengan memilih database "pt-mobilita-nusantara" dan menghitung berapa banyak baris yang ada di tabel "bronze.raw_car_sales_transactions". Jika tabel "bronze" kosong, prosedur ini akan menghentikan proses dan memberikan peringatan. Selanjutnya, prosedur ini melakukan "truncate" pada tabel "silver.transformed_car_sales_transactions" untuk menghapus semua data sebelumnya. Ini memastikan tabel bersih sebelum memasukkan data baru.

Selanjutnya, data dari tabel bronze dimasukkan ke dalam tabel silver. Proses konversi data termasuk mengonversi tipe data desimal dan menghilangkan simbol dolar dan koma dari kolom tahunan pendapatan dan harga (\$). Selain itu, proses ini mengekstrak tahun, bulan, dan hari penjualan untuk dimasukkan sebagai kolom terpisah dalam tabel "silver". Proses pemuatan hanya memasukkan data yang valid, yaitu nilai pada kolom "Car_id" dan "Date".

Dicatat dan ditampilkan jumlah baris yang berhasil dimuat setelah

data dimasukkan. Prosedur ini akan memberikan peringatan agar pengguna memeriksa kondisi "WHERE" atau kemungkinan masalah dalam konversi data jika tidak ada data yang dimuat, meskipun ada data di tabel "bronze". Jika data dimuat dengan sukses, prosedur ini akan mencetak pesan sukses. Pesan menunjukkan bahwa proses pemuatan telah selesai. Pengguna dapat memulai proses dengan mengetik perintah "EXEC dbo.LoadSilverLayerCarSalesTransactions_V2." Sebagai bagian dari alur kerja pengolahan data yang lebih besar, proses ini memungkinkan pemrosesan dan transfer data yang lebih terstruktur dari bronz ke perak.

- Gold Layer Load:

```
USE [pt-mobilita-nusantara];
GO

CREATE OR ALTER PROCEDURE
dbo.LoadGoldLayerData_V2
AS
BEGIN
    SET NOCOUNT ON;
    PRINT '--- Memulai Proses LoadGoldLayerData_V2 ---';

    DECLARE @SilverCount INT;
    SELECT @SilverCount = COUNT(*) FROM
silver.transformed_car_sales_transactions;
    PRINT 'Jumlah baris di
silver.transformed_car_sales_transactions: ' +
CAST(@SilverCount AS VARCHAR(10));
    IF @SilverCount = 0 BEGIN PRINT 'PERINGATAN: Tidak
ada data di silver. Proses dihentikan.'; RETURN; END
```

```

PRINT 'Memulai populasi gold.dim_date...';
INSERT INTO gold.dim_date (date_key, [date], year, month,
day, weekday, month_name, quarter)
SELECT DISTINCT
    CONVERT(INT, CONVERT(VARCHAR(8),
s.Sales_Date, 112)),
    s.Sales_Date,
    YEAR(s.Sales_Date),
    MONTH(s.Sales_Date),
    DAY(s.Sales_Date),
    DATENAME(dw, s.Sales_Date),
    DATENAME(month, s.Sales_Date),
    DATEPART(qq, s.Sales_Date)
FROM silver.transformed_car_sales_transactions s
WHERE NOT EXISTS (
    SELECT 1 FROM gold.dim_date dd
    WHERE dd.date_key = CONVERT(INT,
CONVERT(VARCHAR(8), s.Sales_Date, 112))
)
AND s.Sales_Date IS NOT NULL;
PRINT CAST(@@ROWCOUNT AS VARCHAR(10)) + '
baris baru ditambahkan ke gold.dim_date.';

PRINT 'Memulai populasi gold.dim_car...';
MERGE gold.dim_car AS T
USING (
    SELECT DISTINCT Car_ID, Car_Make, Car_Model,
Engine_Type, Transmission_Type, Car_Color, Body_Style
    FROM silver.transformed_car_sales_transactions
    WHERE Car_ID IS NOT NULL
) AS S

```

```

ON T.car_id = S.Car_ID
WHEN MATCHED THEN
    UPDATE SET T.make=S.Car_Make,
T.model=S.Car_Model, T.engine_type=S.Engine_Type,
    T.transmission_type=S.Transmission_Type,
T.color=S.Car_Color, T.body_style=S.Body_Style
WHEN NOT MATCHED BY TARGET THEN
    INSERT
(car_id,make,model,engine_type,transmission_type,color,body_
style)
    VALUES
(S.Car_ID,S.Car_Make,S.Car_Model,S.Engine_Type,S.Transmi
sion_Type,S.Car_Color,S.Body_Style);
    PRINT CAST(@@ROWCOUNT AS VARCHAR(10)) + '
baris terpengaruh (INSERT/UPDATE) di gold.dim_car.';

PRINT 'Memulai populasi gold.dim_dealer...';
MERGE gold.dim_dealer AS T
USING (
    SELECT Dealer_Number,Dealer_Name,Dealer_Region
    FROM (
        SELECT Dealer_Number,Dealer_Name,Dealer_Region,
        ROW_NUMBER()OVER(PARTITION BY
Dealer_Number ORDER BY Dealer_Name,Dealer_Region) as
rn
        FROM silver.transformed_car_sales_transactions
        WHERE Dealer_Number IS NOT NULL AND
Dealer_Number != ''
    ) AS RD
    WHERE rn=1
) AS S

```



```

ON T.dealer_number=S.Dealer_Number
WHEN MATCHED THEN
    UPDATE SET T.dealer_name=S.Dealer_Name,
T.dealer_region=S.Dealer_Region
WHEN NOT MATCHED BY TARGET THEN
    INSERT (dealer_number,dealer_name,dealer_region)

VALUES(S.Dealer_Number,S.Dealer_Name,S.Dealer_Region);
PRINT CAST(@@ROWCOUNT AS VARCHAR(10)) + '
baris terpengaruh (INSERT/UPDATE) di gold.dim_dealer.';

PRINT 'Memulai populasi gold.dim_customer...';
MERGE gold.dim_customer AS T
USING (
    SELECT DISTINCT
Customer_Name,Gender,Annual_Income,Customer_Phone
    FROM silver.transformed_car_sales_transactions
    WHERE (Customer_Name IS NOT NULL AND
Customer_Name != "")
        OR (Customer_Phone IS NOT NULL AND
Customer_Phone != "")
    ) AS S
ON T.customer_name=S.Customer_Name AND
ISNULL(T.customer_phone,'N/A')=ISNULL(S.Customer_Phone,'N/A')
WHEN MATCHED THEN
    UPDATE SET T.gender=S.Gender,
T.annual_income=S.Annual_Income
WHEN NOT MATCHED BY TARGET THEN
    INSERT
(customer_name,gender,annual_income,customer_phone)

```

```

VALUES(S.Customer_Name,S.Gender,S.Annual_Income,S.Cus
tomer_Phone);

PRINT CAST(@@ROWCOUNT AS VARCHAR(10)) + '
baris terpengaruh (INSERT/UPDATE) di gold.dim_customer.';

PRINT 'Memulai populasi gold.fact_car_sales...';
TRUNCATE TABLE gold.fact_car_sales;
PRINT 'Tabel gold.fact_car_sales telah di-TRUNCATE.';
INSERT INTO gold.fact_car_sales
(car_id_original,date_key,car_key,dealer_key,customer_key,sale
s_price)
SELECT s.Car_ID, dd.date_key, dc.car_key, ddl.dealer_key,
dcu.customer_key, s.Sales_Price
FROM silver.transformed_car_sales_transactions s
INNER JOIN gold.dim_date dd ON dd.date_key =
CONVERT(INT, CONVERT(VARCHAR(8), s.Sales_Date,
112))
INNER JOIN gold.dim_car dc ON dc.car_id = s.Car_ID
INNER JOIN gold.dim_dealer ddl ON ddl.dealer_number =
s.Dealer_Number
INNER JOIN gold.dim_customer dcu ON dcu.customer_name
= s.Customer_Name
AND ISNULL(dcu.customer_phone,
'N/A') = ISNULL(s.Customer_Phone, 'N/A')
WHERE s.Sales_Price IS NOT NULL;
PRINT CAST(@@ROWCOUNT AS VARCHAR(10)) + '
baris berhasil dimasukkan ke gold.fact_car_sales.';

PRINT '--- Proses LoadGoldLayerData_V2 Selesai ---';
END

```

```
GO
```

```
PRINT 'Stored procedure dbo.LoadGoldLayerData_V2 berhasil  
dibuat/diperbarui.';
```

Dalam script SQL yang diberikan, prosedur tersimpan "LoadGoldLayerData_V2" digunakan untuk memindahkan dan mengolah data dari lapisan emas ke lapisan perak dalam alur data. Metode ini dimulai dengan melihat berapa banyak data yang ada di tabel "silver.transformed_car_sales_transactions". Jika tidak ada data di dalamnya, proses akan dihentikan dengan peringatan. Selanjutnya, proses ini dimulai dengan memasukkan populasi data ke dalam tabel "gold.dim_date", yang mengandung informasi seperti "date_key", tahun, bulan, hari, nama bulan, dan kuartal yang diambil dari tanggal penjualan. Prosedur ini memastikan bahwa hanya data yang belum ada di dalam tabel dimasukkan.

Setelah itu, perintah "MERGE" digunakan untuk memasukkan data tentang mobil ke dalam tabel "gold.dim_car". Proses ini memeriksa apakah data mobil sudah ada dan memasukkan data baru jika ada. Proses yang sama dilakukan untuk tabel "gold.dim_dealer", di mana data dealer yang ada di tabel "silver" dimasukkan atau diperbarui ke dalam tabel "gold". Dengan menggunakan "ROW_NUMBER()&

Selanjutnya, data pelanggan dimasukkan atau diperbarui ke dalam tabel "gold.dim_customer" untuk memastikan bahwa semua nama dan nomor pelanggan dimasukkan dengan benar. Setelah itu, perintah "TRUNCATE" digunakan untuk mengosongkan tabel "gold.fact_car_sales" dan kemudian mengisi tabel dengan data dari "silver": "car_id_original", "date_key", "car_key&

Pesan yang mencetak jumlah baris yang telah dimasukkan atau

diperbarui di masing-masing tabel menunjukkan setiap langkah prosedur ini, memberikan informasi yang berguna tentang proses pemuatan data. Prosedur ini diakhiri dengan pesan yang menunjukkan bahwa proses pemuatan data ke dalam layer gold telah selesai, dan pesan akan muncul untuk mengonfirmasi bahwa definisi atau perubahan prosedur berhasil.

7. Hasil Implementasi

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'pt-mobilita-nusantara'. The central pane shows a query window with the following SQL code:

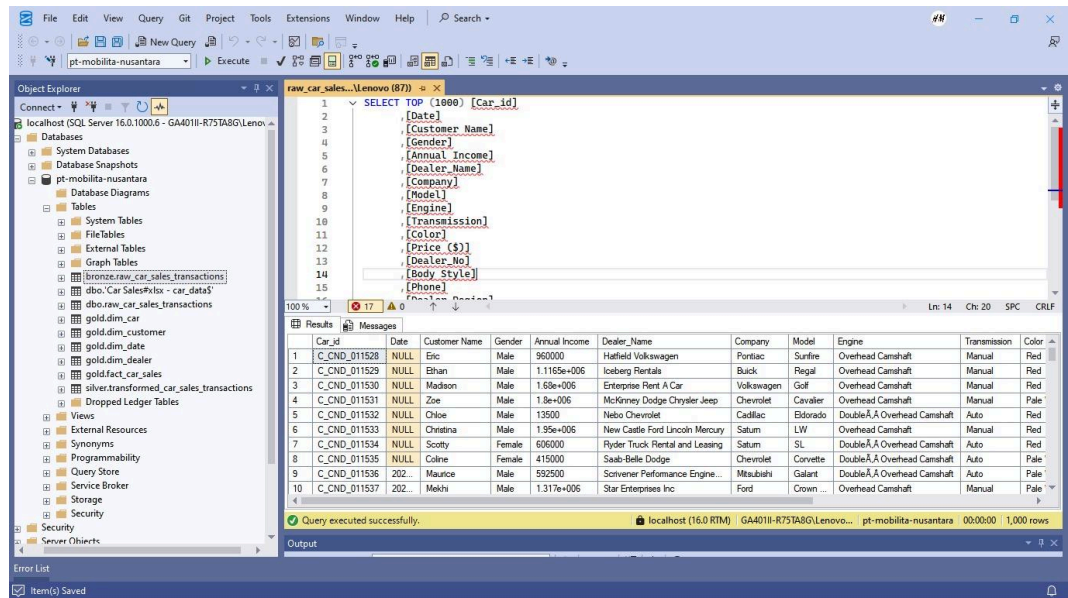
```
SELECT TOP (1000) [Car_id]
, [Date]
, [Customer Name]
, [Gender]
, [Annual Income]
, [Dealer_Name]
, [Company]
, [Model]
, [Engine]
, [Transmission]
, [Color]
, [Price ($)]
, [Dealer_No]
, [Body Style]
, [Phone]
FROM [raw_car_sales_transactions]
```

The Results pane at the bottom displays the following data:

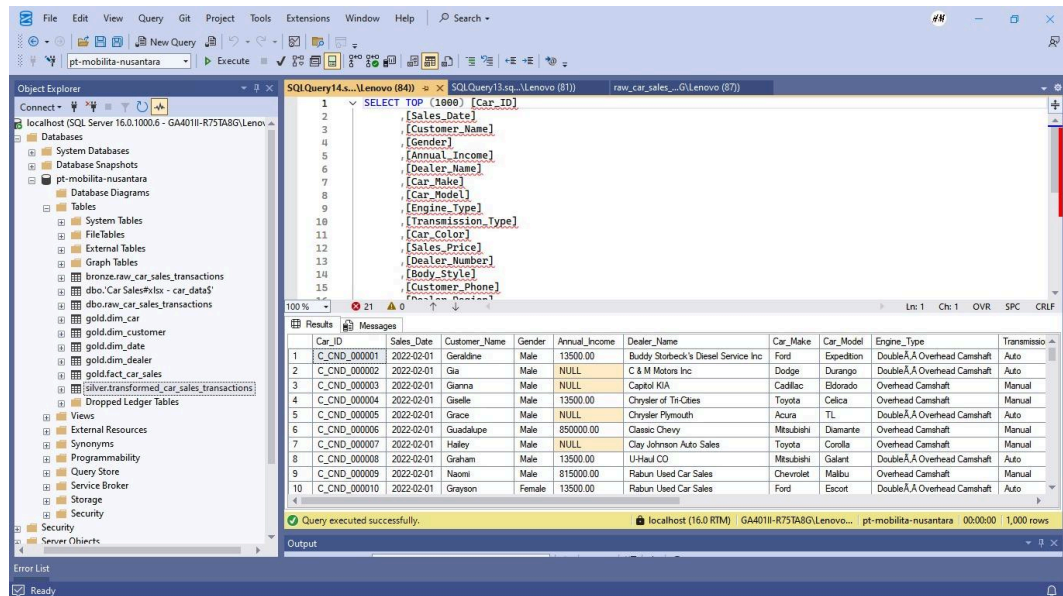
Car_id	Date	Customer Name	Gender	Annual Income	Dealer_Name	Company	Model	Engine	Transmission	Color
C_CND_011528	NULL	Eric	Male	960000	Hatfield Volkswagen	Pontiac	Surfline	Overhead Camshaft	Manual	Red
C_CND_011529	NULL	Ethan	Male	1.115e+006	Isenberg Rentals	Buick	Regal	Overhead Camshaft	Manual	Red
C_CND_011530	NULL	Madsen	Male	1.63e+006	Enterprise Rent A Car	Volkswagen	Golf	Overhead Camshaft	Manual	Red
C_CND_011531	NULL	Zoe	Male	1.8e+006	McKinney Dodge Chrysler Jeep	Chevrolet	Cavalier	Overhead Camshaft	Manual	Pale
C_CND_011532	NULL	Chloe	Male	13500	Nebo Chevrolet	Cadillac	Elorado	Double A Overhead Camshaft	Auto	Red
C_CND_011533	NULL	Christina	Male	1.95e+006	New Castle Ford Lincoln Mercury	Saturn	LW	Overhead Camshaft	Manual	Red
C_CND_011534	NULL	Scotty	Female	606000	Ryder Truck Rental and Leasing	Saturn	SL	Double A Overhead Camshaft	Auto	Red
C_CND_011535	NULL	Coline	Female	415000	Saab-Belle Dodge	Chevrolet	Corvette	Double A Overhead Camshaft	Auto	Pale
C_CND_011536	202...	Maurice	Male	532500	Scovener Performance Engine...	Mitsubishi	Galant	Double A Overhead Camshaft	Auto	Pale
C_CND_011537	202...	Mekhi	Male	1.317e+006	Star Enterprises Inc	Ford	Crown	Overhead Camshaft	Manual	Pale

The status bar at the bottom indicates 'Query executed successfully' and shows the execution time as 00:00:00 for 1,000 rows.

Pesan "Query executed successfully" yang muncul di bagian bawah layar menandakan bahwa query untuk menampilkan data dari tabel berhasil dieksekusi tanpa kesalahan. Selain itu, data transaksi penjualan mobil, seperti informasi mengenai pelanggan, dealer, model mobil, transmisi, dan harga, ditampilkan dengan benar dalam hasil query. Struktur database yang terlihat di panel kiri menunjukkan bahwa data ini telah tersimpan dengan baik di dalam tabel raw_car_sales_transactions. Tidak ada pesan kesalahan yang muncul, yang berarti proses pengambilan data berjalan lancar. Dengan SQL yang terstruktur dengan baik, data yang ditampilkan siap digunakan untuk analisis lebih lanjut dalam tugas data warehouse Anda. Secara keseluruhan, dapat disimpulkan bahwa dataset ini telah berhasil dimasukkan ke dalam SQL Server dan siap untuk proses selanjutnya.



Di layer bronze, data umumnya berupa data mentah yang belum mengalami transformasi atau pembersihan yang signifikan. Dalam gambar ini, kita dapat melihat transaksi penjualan mobil yang mencakup informasi seperti ID mobil, nama pelanggan, jenis kelamin, penghasilan tahunan, nama dealer, model mobil, transmisi, dan warna. Data ini berasal langsung dari sumbernya dan belum diproses lebih lanjut. Layer bronze berfungsi sebagai penyimpanan data mentah yang akan diproses lebih lanjut di layer silver (transformasi) dan gold (analisis dan laporan). Data ini memungkinkan Anda untuk menyimpan dan mengakses informasi yang lengkap sebelum melakukan pemrosesan lebih lanjut.



Di SQL Server, layer silver biasanya mencakup data yang sudah mengalami transformasi, yaitu data yang telah dibersihkan dan diubah dari format mentah ke format yang lebih terstruktur dan siap untuk dianalisis. Dalam gambar ini, tabel `transformed_car_sales_transactions` menunjukkan transaksi penjualan mobil yang telah diproses dan disusun dengan lebih terorganisir.

Data yang ditampilkan mencakup informasi penting seperti `Sales_Date`, `Customer_Name`, `Gender`, `Annual_Income`, `Dealer_Name`, `Car_Make`, `Car_Model`, `Engine_Type`, `Transmission`, dan harga jual mobil (`Sales_Price`). Berbeda dengan layer bronze, yang berisi data mentah tanpa banyak perubahan, layer silver biasanya berisi data yang telah dibersihkan dan diproses untuk memastikan kualitasnya, seperti menghapus data yang tidak lengkap atau mengoreksi kesalahan.

Query executed successfully. 1,000 rows.

customer_key	customer_name	gender	annual_income	customer_phone
1	Aanya	Male	13500.00	6.48276e+006
2	Aayan	Male	13500.00	7.38128e+006
3	Abdel	Female	NULL	6.59128e+006
4	Aden	Female	NULL	8.19837e+006
5	Aden	Female	700000.00	8.92016e+006
6	Ahanon	Male	NULL	6.25021e+006
7	Asha	Male	13500.00	6.87347e+006
8	Aissatou	Male	13500.00	7.48762e+006
9	Albert	Female	NULL	6.63204e+006
10	Alexander	Male	NULL	7.63461e+006

Di layer gold, data sudah mengalami pembersihan, transformasi, dan penggabungan untuk siap digunakan dalam analisis bisnis atau laporan. Dalam hal ini, tabel `dim_customer` menyimpan data terkait pelanggan, termasuk `customer_key`, `customer_name`, `gender`, `annual_income`, dan `customer_phone`. Layer gold adalah tempat di mana data sudah dalam bentuk final yang siap untuk digunakan oleh pengambil keputusan. Dengan data yang sudah terstruktur dan bersih, Anda dapat melakukan analisis yang lebih mendalam dan membuat laporan yang lebih tepat sasaran.

Query executed successfully. 1,000 rows.

car_key	car_id	make	model	engine_type	transmission_type	color	body_style
1	C_CND_000001	Ford	Expedition	Double A Overhead Camshaft	Auto	Black	SUV
2	C_CND_000002	Dodge	Durango	Double A Overhead Camshaft	Auto	Black	SUV
3	C_CND_000003	Cadillac	Eldorado	Overhead Camshaft	Manual	Red	Passenger
4	C_CND_000004	Toyota	Celica	Overhead Camshaft	Manual	Pale White	SUV
5	C_CND_000005	Acura	TL	Double A Overhead Camshaft	Auto	Red	Hatchback
6	C_CND_000006	Mitsubishi	Diamante	Overhead Camshaft	Manual	Pale White	Hatchback
7	C_CND_000007	Toyota	Corolla	Overhead Camshaft	Manual	Pale White	Passenger
8	C_CND_000008	Mitsubishi	Galant	Double A Overhead Camshaft	Auto	Pale White	Passenger
9	C_CND_000009	Chevrolet	Malibu	Overhead Camshaft	Manual	Pale White	Hardtop
10	C_CND_000010	Ford	Escort	Double A Overhead Camshaft	Auto	Pale White	Passenger
11	C_CND_000011	Acura	RL	Overhead Camshaft	Manual	Pale White	SUV

Variasi tampilan layer gold untuk top 1000 car_key

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'pt-mobilita-nusantara'. The central query window shows the following SQL query:

```
SELECT TOP (1000) [date_key]
FROM [pt-mobilita-nusantara].[gold].[dim_date]
```

The Results pane at the bottom displays the top 10 rows of the query results:

date_key	date	year	month	day	weekday	month_name	quarter
1	20220103	2022	1	3	Monday	January	1
2	20220105	2022	1	5	Wednesday	January	1
3	20220106	2022	1	6	Thursday	January	1
4	20220108	2022	1	8	Saturday	January	1
5	20220109	2022	1	9	Sunday	January	1
6	20220110	2022	1	10	Monday	January	1
7	20220111	2022	1	11	Tuesday	January	1
8	20220112	2022	1	12	Wednesday	January	1
9	20220201	2022	2	1	Tuesday	February	1
10	20220203	2022	2	3	Thursday	February	1

The status bar indicates the query executed successfully, returning 238 rows.

Variasi tampilan layer gold untuk top 1000 date_key

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'pt-mobilita-nusantara'. The central query window shows the following SQL query:

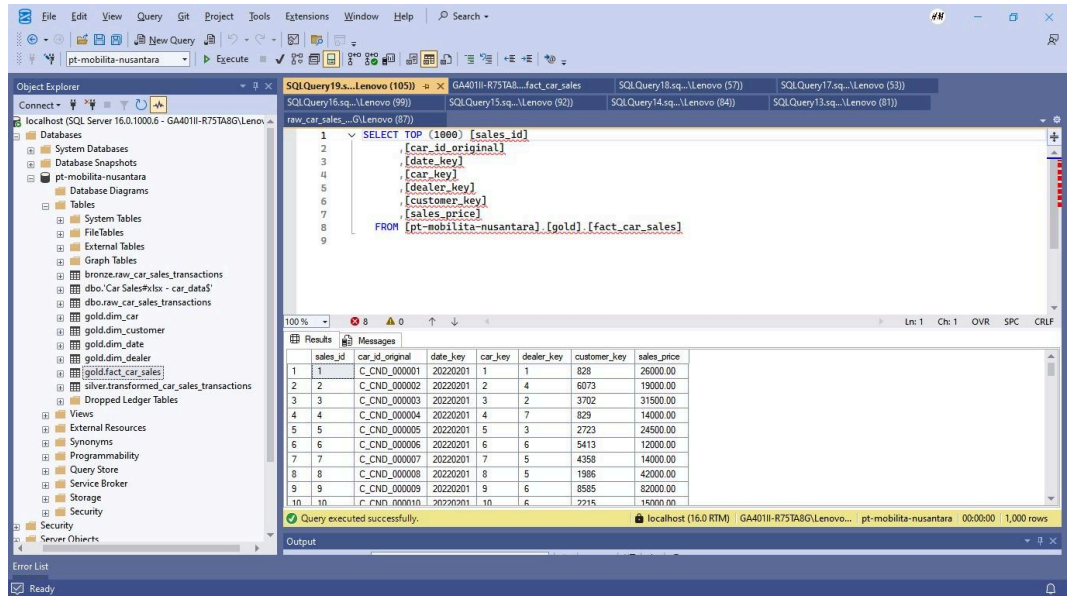
```
SELECT TOP (1000) [dealer_key]
FROM [pt-mobilita-nusantara].[gold].[dim_dealer]
```

The Results pane at the bottom displays the top 7 rows of the query results:

dealer_key	dealer_number	dealer_name	dealer_region
1	06457-3834	Buddy Storbeck's Diesel Service Inc	Aurora
2	38701-8047	Capitol KIA	Aurora
3	53546-9427	Chrysler Plymouth	Aurora
4	60504-7114	C & M Motors Inc	Aurora
5	70788-7841	Clay Johnson Auto Sales	Aurora
6	85257-3102	Classic Chevy	Aurora
7	99301-3882	Chrysler of Tri-Cities	Aurora

The status bar indicates the query executed successfully, returning 7 rows.

Variasi tampilan layer gold untuk top 1000 dealer_key



Variasi tampilan layer gold untuk top 1000 sales_id

Table: 'Car Sales#xlsx - car_data' (23 906 rows) (23 906 rows)

Date	Customer Name	Gender	Annual Income	Dealer Name	Company	Model	Engine
C_CND_000203	Mathis	Male	13500	Saab-Belle Dodge	Dodge	Viper	DoubleAA Over
C_CND_000221	Zoey	Male	13500	Ryder Truck Rental and Leasing	Pontiac	Grand Am	DoubleAA Over
C_CND_000498	Kenneth	Male	13500	Saab-Belle Dodge	BMW	528i	DoubleAA Over
C_CND_000499	Nicholas	Male	13500	Nebo Chevrolet	Mercury	Grand Marquis	Overhead Cams
C_CND_000501	Nico	Male	13500	New Castle Ford Lincoln Mercury	BMW	528i	Overhead Cams
C_CND_012106	Nikolas	Male	13500	Pitre Buick-Pontiac-Gmc of Scottsdale	BMW	528i	DoubleAA Over
C_CND_012276	Brandon	Male	13500	Motor Vehicle Branch Office	Honda	Passport	Overhead Cams
C_CND_012277	Chase	Male	13500	Enterprise Rent A Car	Mercury	Grand Marquis	Overhead Cams
C_CND_012335	Hadassah	Male	13500	Gartner Buick Hyundai Saab	Volkswagen	GTI	DoubleAA Over
C_CND_003164	Claire	Male	13500	Rabun Used Car Sales	Chevrolet	Prizm	Overhead Cams
C_CND_003180	Leon	Male	13500	Chrysler of Tri-Cities	Volvo	S70	Overhead Cams
C_CND_003343	Jacqueline	Male	13500	Pitre Buick-Pontiac-Gmc of Scottsdale	Plymouth	Neon	DoubleAA Over
C_CND_003365	Rohan	Male	13500	Star Enterprises Inc	Dodge	Ram Pickup	DoubleAA Over
C_CND_003422	Imani	Male	13500	Pitre Buick-Pontiac-Gmc of Scottsdale	Plymouth	Neon	DoubleAA Over
C_CND_003451	Haley	Male	13500	U-Haul CO	Plymouth	Neon	DoubleAA Over
C_CND_002614	Aaron	Male	13500	Ryder Truck Rental and Leasing	Chevrolet	Cavalier	Overhead Cams
C_CND_002729	Mamadou	Male	13500	Progressive Shippers Cooperative Association No	Dodge	Durango	Overhead Cams
C_CND_002749	Lyllian	Male	13500	Progressive Shippers Cooperative Association No	Volkswagen	GTI	Overhead Cams
C_CND_002787	Aidan	Male	13500	Race Car Help	Mercedes-B	S-Class	DoubleAA Over
	Allen	Male	13500	Tri-State Mack Inc	Mercedes-B	SL-Class	Overhead Cams

Berhasil menghubungkan Power BI ke SQL Server dan berhasil menggunakan data dari layer gold. Di sini, Power BI berhasil membuat tabel yang berisi informasi tentang transaksi penjualan mobil, termasuk Car_id, Customer Name, Gender, Annual Income, dan dealer mobil terkait. Penggunaan layer gold di Power BI berarti data yang digunakan sudah dalam bentuk yang telah dibersihkan, terstruktur, dan siap untuk analisis lebih lanjut.

8. Evaluasi

Selama Pelaksanaan proyek Data Warehouse untuk PT Mobilita Nusantara, terdapat sejumlah pencapaian penting yang berhasil dicapai:

1. Implementasi Arsitektur Medallion (Bronze, Silver, Gold): Proyek berhasil mengadopsi arsitektur tiga lapis yang memisahkan data mentah, data tertransformasi, dan data siap analisis secara sistematis. Hal ini memudahkan pelacakan alur data (data lineage) dan meningkatkan kualitas manajemen data.
2. Proses ETL Berhasil Dilaksanakan: Proses Extract, Transform, Load dilakukan dengan baik di setiap lapisan, memastikan bahwa data dari file CSV dapat dimuat, dibersihkan, dan disiapkan untuk analisis dengan akurasi tinggi.
3. Model Star Schema yang Fungsional di Gold Layer: Skema bintang yang terdiri dari satu tabel fakta dan beberapa tabel dimensi berhasil dibangun untuk mendukung analisis multidimensi dan pelaporan yang efisien.
4. Eksekusi Query Analitik Berjalan Lancar: Query OLAP sederhana, seperti total penjualan per bulan, model mobil terlaris, hingga segmentasi pelanggan berdasarkan demografi, berhasil dijalankan dan memberikan insight bisnis yang bernilai.

Beberapa kendala dan keterbatasan juga ditemukan selama pelaksanaan proyek:

1. Masalah Format Data Awal: Data mentah dari file CSV memiliki format yang tidak konsisten, seperti nilai numerik yang disimpan sebagai string dan adanya nilai NULL yang tersembunyi. Ini memerlukan waktu tambahan dalam proses transformasi.
2. Performa Kueri pada Data Skala Besar: Uji performa belum dilakukan pada data dengan skala besar. Kemungkinan perlu adanya optimasi lebih lanjut, terutama pada struktur indeks dan partisi tabel.

3. Penanganan Data Historis Terbatas: Proyek saat ini hanya menggunakan pendekatan SCD (Slowly Changing Dimensions) Tipe 1. Hal ini belum cukup untuk melacak perubahan historis dalam data dimensi yang kompleks, seperti perubahan pendapatan pelanggan atau wilayah dealer dari waktu ke waktu.
4. Proses ETL Masih Manual: Prosedur ETL dijalankan secara manual melalui SSMS, yang berpotensi menimbulkan ketidakkonsistenan jika dilakukan secara berulang tanpa automasi.

Dari pelaksanaan proyek ini, tim memperoleh sejumlah pembelajaran penting:

1. Pentingnya Desain Awal yang Matang: Tahap analisis kebutuhan dan desain konseptual sangat krusial untuk menjamin keberhasilan arsitektur data warehouse secara keseluruhan.
2. Kualitas Data Menentukan Kualitas Analisis: Proses pembersihan dan validasi data (data cleansing) memegang peran besar dalam memastikan data yang digunakan dalam laporan bersifat akurat dan dapat diandalkan.
3. Kolaborasi Tim dan Versi Kode Penting: Penggunaan GitHub sebagai repositori bersama terbukti membantu dalam manajemen versi dan dokumentasi proyek, sekaligus mencegah konflik dalam pengembangan.
4. Integrasi Alat Visualisasi Memperbagus Insight: Menghubungkan SQL Server ke Power BI memberikan nilai tambah dalam pemahaman data serta mempercepat proses penyampaian insight ke stakeholder bisnis.

9. Rencana Pengembangan ke Depan

Proyek data warehouse ini memiliki potensi untuk dikembangkan lebih lanjut guna meningkatkan fungsionalitas dan nilai bisnisnya.

Ide-ide untuk pengembangan lebih lanjut:

- Integrasi Sumber Data Tambahan: Mempertimbangkan untuk mengintegrasikan sumber data lain yang relevan untuk analisis yang lebih kaya.
- Implementasi Dashboard Visualisasi Data: Mengembangkan dashboard interaktif menggunakan perangkat lunak seperti Power BI atau Tableau untuk mempermudah pemahaman data dan tren.
- Otomatisasi Penuh Pipeline ETL: Membuat pipeline ETL yang sepenuhnya otomatis dengan menggunakan alat penjadwalan atau tools ETL khusus seperti SQL Server Integration Services (SSIS), Azure Data Factory, atau Airflow.
- Implementasi SCD Tipe 2: Menerapkan Slowly Changing Dimensions (SCD) Tipe 2 untuk melacak dan mengelola perubahan historis pada data dimensi secara lebih detail.
- Optimasi Performa Lanjutan: Melakukan optimasi performa lebih lanjut untuk menangani volume data yang jauh lebih besar dan memastikan kueri tetap berjalan efisien.

10. Tim Proyek

Nama Anggota	NIM	Pembagian Tugas
Arafi Ramadhan Maulana	122450002	<ul style="list-style-type: none"> - Membuat DDL untuk tabel dimensi dan fakta di Gold Layer (<code>05_Gold_Layer_DDL.sql</code>) yang membentuk Data Mart untuk analisis - Membuat stored procedure untuk memuat data dari Silver Layer ke tabel dimensi dan fakta di Gold Layer (<code>06_Gold_Layer_Load_Procedure.sql</code>).
Rayan Koemi Karuby	122450038	<ul style="list-style-type: none"> - Menghubungkan Power BI ke Data Mart di Gold Layer - Merancang dan membuat dasbor serta visualisasi data yang insightful berdasarkan kebutuhan analisis penjualan mobil - Melakukan eksplorasi data dan menyajikan temuan kunci.
Hermawan Manurung	122450069	<ul style="list-style-type: none"> - Mendefinisikan secara keseluruhan arsitektur Medallion (Bronze, Silver, Gold) yang akan diterapkan - Membuat desain konseptual dan logis untuk setiap lapisan - Menyusun dokumentasi konsep arsitektur, termasuk <code>data_catalog.md</code> awal
Chevando Daffa Pramanda	122450095	<ul style="list-style-type: none"> - Membuat DDL untuk tabel di Silver Layer (<code>03_Silver_Layer_DDL.sql</code>) yang mencakup transformasi, pembersihan tipe data, dan penambahan kolom turunan - Membuat stored procedure untuk memuat dan mentransformasi data dari Bronze Layer ke Silver Layer (<code>04_Silver_Layer_Load_Procedure.sql</code>)
Mirzan Yusuf Rabbani	122450118	<ul style="list-style-type: none"> - Memastikan semua script DDL dan stored procedure (Bronze, Silver, Gold) dapat dijalankan dengan

		<p>benar di SQL Server Management Studio 21</p> <ul style="list-style-type: none"> - Melakukan pengujian aliran data dari awal hingga akhir - Bertanggung jawab atas proses impor data awal jika belum dilakukan
Daffa Ahmad Naufal	122450137	<ul style="list-style-type: none"> - Membuat DDL (Data Definition Language) untuk tabel di Bronze Layer (01_Bronze_Layer_DDL . sql) berdasarkan data sumber - Membuat script untuk memuat data mentah dari sumber (misalnya, tabel hasil impor Excel) ke tabel Bronze (02_Bronze_Layer_Load_Procedure . sql atau script INSERT INTO SELECT)

File Dokumentasi Proyek

- <https://github.com/sains-data/dw-pt-mobilita-nusantara>.