

# ANALISIS KEBUTUHAN BISNIS DAN TEKNIS UNTUK MERANCANG DATA WAREHOUSE PADA INDUSTRI RIDE-HAILING OKI (OJEK KAMPUS ITERA)

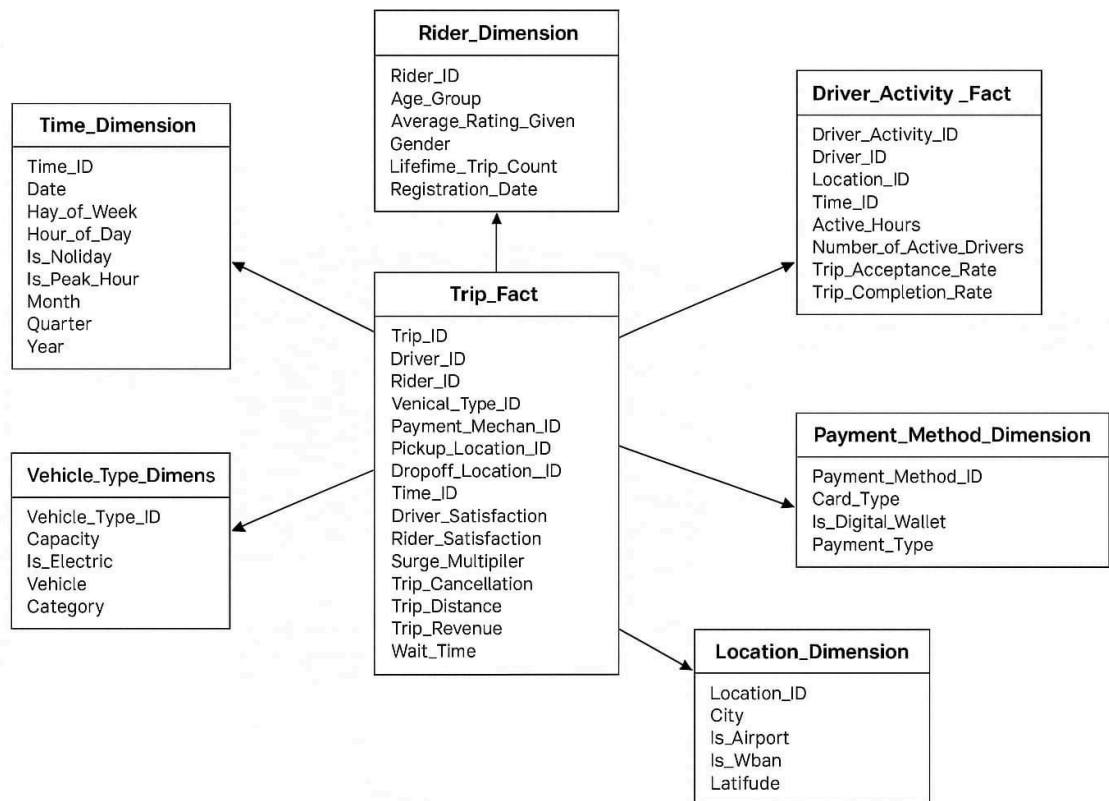
## Anggota Kelompok 26 :

- |                           |           |
|---------------------------|-----------|
| 1. Muhammad Rendy Saputra | 121450045 |
| 2. Asa Do'a Uyi           | 122450005 |
| 3. Tessa Kania Sagala     | 122450040 |
| 4. Presilia               | 122450081 |
| 5. Ahmad Rizqi            | 122450138 |

## SKEMA KONSEPTUAL MULTIDIMENSI

### A. Diagram *Star Schema* (Fakta–Dimensi)

Diagram *star schema* berikut menggambarkan hubungan antara tabel fakta dan dimensi dalam data warehouse OKI. Tabel fakta seperti *Trip Fact Table* menyimpan metrik utama (misalnya, kepuasan dan *surge multiplier*), sementara dimensi seperti *Time* dan *Location* memberikan konteks untuk analisis, mendukung pengambilan keputusan yang lebih baik sesuai kebutuhan bisnis.



## DESAIN LOGIKAL

### A. Struktur Tabel

Data warehouse OKI menggunakan skema *star schema*, yang terdiri dari tabel dimensi (menyimpan data deskriptif) dan tabel fakta (menyimpan metrik transaksi). Berikut adalah rincian tabel:

### B. Tabel Dimensi

#### - Time Dimension

Peran : Menyimpan informasi waktu untuk analisis operasional dan pola penggunaan layanan OKI berdasarkan tanggal dan jam.

Kolom	Tipe Data	Peran
Time_ID	varchar(10)	Primary key, sebagai identifikasi unik setiap waktu (misalnya gabungan tanggal dan jam).
Date	date	Tanggal spesifik dari waktu tersebut. Berguna untuk mengelompokkan data perjalanan.
Hour_of_Day	int	Menunjukkan jam (0-23) dari waktu tersebut.
Day_of_Week	varchar(10)	Nama hari (Senin–Minggu), berguna untuk melihat perbedaan aktivitas antara hari kerja dan akhir pekan.
Month	varchar(10)	Nama bulan dari waktu tersebut.
Quarter	varchar(2)	Kuartal dalam tahun (Q1, Q2, Q3, Q4).
Year	int	Tahun dari waktu tersebut.
Is_Holiday	char(1)	Menandai apakah hari tersebut hari libur ('Y' atau 'N').
Is_Peak_Hour	char(1)	Menandai apakah jam tersebut merupakan jam sibuk OKI (misalnya jam masuk/kuliah/istirahat).
Is_Current	bit	Menunjukkan apakah waktu ini adalah waktu “aktif” atau waktu saat ini dalam konteks analisis.

#### - Location Dimension

Peran : Menyimpan informasi geografis untuk analisis berbasis lokasi dalam sistem OKI (Ojek Kampus ITERA), seperti lokasi titik jemput/antar, area operasional, dan hub penting (termasuk kemungkinan perluasan ke bandara terdekat).

Kolom	Tipe Data	Peran
Location_ID	varchar(5)	Primary Key, sebagai identifikasi unik untuk setiap lokasi geografis.

City	varchar(50)	Nama kota atau area dari lokasi tersebut.
Latitude	decimal(9,4)	Koordinat garis lintang lokasi untuk keperluan pemetaan.
Longitude	decimal(9,4)	Koordinat garis bujur lokasi untuk keperluan pemetaan.
Is_Airport	char(1)	Menandai apakah lokasi tersebut merupakan bandara (Y = ya, N = tidak).
Is_Urban	char(1)	Menandai apakah lokasi tersebut berada di area perkotaan atau pusat aktivitas kampus.
Is_Current	bit	Menunjukkan apakah lokasi ini sedang menjadi fokus analisis atau yang aktif saat ini (1 = ya, 0 = tidak).

- **Rider Dimension**

Peran : Menyimpan data pengguna layanan OKI, yang mencakup informasi demografis dan perilaku perjalanan mahasiswa atau civitas kampus.

Kolom	Tipe Data	Peran
Rider_ID	varchar(5)	Primary Key, identifikasi unik untuk setiap pengguna OKI.
Age_Group	varchar(10)	Kelompok usia (misalnya: “<18”, “18–22”, “23–27”), bisa dipakai untuk analisis segmentasi pengguna.
Gender	char(1)	Jenis kelamin pengguna OKI (M = laki-laki, F = perempuan).
Registration_Date	date	Tanggal saat rider pertama kali menggunakan atau mendaftar OKI.
Lifetime_Trip_Count	int	Total perjalanan rider sejak pertama kali menggunakan OKI.
Average_Rating_Given	decimal(3,1)	Rata-rata rating yang diberikan rider untuk pengemudi, berguna untuk monitoring kualitas layanan.
Is_Current	bit	Menunjukkan apakah rider masih aktif dalam sistem OKI atau sedang dianalisis saat ini.

- **Driver Dimension**

Peran : Menyimpan informasi pengemudi OKI untuk analisis kinerja, status keaktifan, dan pola operasional mereka.

Kolom	Tipe Data	Peran
Driver_ID	varchar(5)	Primary Key, sebagai identifikasi unik setiap pengemudi OKI.
City_of_Operation	varchar(50)	Nama kota atau wilayah operasional, misalnya "Bandar Lampung" untuk cakupan analisis geografis.
Registration_Date	date	Tanggal pengemudi mendaftar sebagai bagian dari layanan OKI.
Average_Rating	decimal(3,1)	Rata-rata penilaian dari penumpang, untuk evaluasi kualitas layanan pengemudi.
Is_Full_Time	char(1)	Menandai apakah pengemudi bekerja penuh waktu (Y = ya, N = tidak).
Is_Current	bit	Menunjukkan apakah pengemudi masih aktif dalam layanan OKI (1 = aktif, 0 = tidak).

- **Vehicle\_Type\_Dimension**

Peran : Menyimpan informasi tentang jenis kendaraan yang digunakan dalam layanan OKI.

Kolom	Tipe Data	Peran
Vehicle_Type_ID	varchar(5)	Sebagai Primary Key yang mengidentifikasi secara unik setiap jenis kendaraan dalam sistem.
Vehicle_Category	varchar(50)	Menyediakan kategori kendaraan (misalnya "Ojek", "Mobil", "Motor") untuk pengelompokan kendaraan dalam aplikasi.
Capacity	int	Menyimpan informasi mengenai kapasitas kendaraan, yaitu jumlah penumpang yang dapat diangkut.
Is_Electric	char(1)	Menyatakan apakah kendaraan tersebut listrik atau bukan (misalnya 'Y' untuk Ya, 'N' untuk Tidak).
Is_Current	bit	Menunjukkan status apakah kendaraan tersebut masih aktif atau tidak dalam sistem.

- **Payment\_Method\_Dimension**

Peran : Mengelola dan menyimpan informasi terkait dengan berbagai metode pembayaran yang tersedia untuk pengguna.

Kolom	Tipe Data	Peran
-------	-----------	-------

Payment_Method_ID	varchar(5)	Sebagai Primary Key yang mengidentifikasi secara unik setiap metode pembayaran dalam sistem OKI.
Payment_Type	varchar(50)	Menyediakan informasi mengenai jenis pembayaran (misalnya "Tunai", "Kartu Kredit", "Dompot Digital").
Card_Type	varchar(50)	Menyimpan informasi tentang jenis kartu yang digunakan dalam pembayaran kartu (misalnya "Visa", "MasterCard").
Is_Digital_Wallet	char(1)	Menyatakan apakah metode pembayaran tersebut menggunakan dompet digital (misalnya 'Y' untuk Ya, 'N' untuk Tidak).
Is_Current	bit	Menunjukkan apakah metode pembayaran tersebut masih aktif atau tidak dalam sistem OKI (1 untuk aktif, 0 untuk tidak aktif).

### C. Tabel Fakta

Tabel fakta menyimpan metrik kuantitatif seperti pendapatan dan jarak, disimpan di filegroup FG\_Fact dengan partisi berdasarkan Time\_ID. Berikut adalah tabel fakta:

#### - Transaction Fact

Peran : Merekam dan mengelola seluruh informasi transaksi yang terjadi dalam sistem OKI

Kolom	Tipe Data	Peran
Transaction_ID	int	Primary Key yang mengidentifikasi secara unik setiap transaksi dalam sistem.
Time_ID	varchar(10)	Foreign Key yang merujuk pada tabel waktu, menghubungkan transaksi dengan waktu terjadinya transaksi (misalnya, tanggal atau jam transaksi).
Rider_ID	varchar(5)	Foreign Key yang merujuk pada pengendara atau pengguna yang melakukan transaksi. Ini membantu melacak transaksi yang dilakukan oleh rider tertentu.
Payment_Method_ID	varchar(5)	Foreign Key yang merujuk pada metode pembayaran yang digunakan dalam transaksi, seperti kartu kredit atau dompet digital.
Trip_ID	int	Menyimpan ID perjalanan yang terkait dengan transaksi tersebut. Dapat digunakan untuk menghubungkan transaksi dengan detail perjalanan.
Transaction_Amount	decimal(10,2)	Menyimpan jumlah uang yang terlibat dalam

		transaksi. Ini berfungsi untuk menghitung total pendapatan atau biaya dari transaksi tertentu.
Fraud_Probability_Score	int	Menyimpan skor probabilitas terjadinya penipuan pada transaksi. Ini membantu sistem untuk mendeteksi transaksi yang mencurigakan dan perlu ditindaklanjuti.
Transaction_Frequency	int	Menyimpan informasi tentang frekuensi transaksi oleh pengguna atau rider tertentu, yang dapat digunakan untuk analisis pola transaksi atau perilaku pengguna.
BatchID	int	Menyimpan ID batch yang berfungsi untuk mengelompokkan transaksi-transaksi yang diproses bersama dalam satu waktu atau proses.
InsertedDate	date	Menyimpan tanggal ketika transaksi dimasukkan ke dalam sistem untuk keperluan pencatatan dan audit.
SourceSystemID	int	Menyimpan ID sistem sumber yang memproses transaksi tersebut, membantu melacak asal transaksi (misalnya, aplikasi atau sistem eksternal yang digunakan).

#### - Trip Fact

Peran: Mencatat dan mengelola seluruh informasi perjalanan yang terjadi dalam sistem transportasi OKI, termasuk pendapatan, jarak, kepuasan, dan atribut perjalanan lainnya.

Kolom	Tipe data	Peran
Trip_ID	int	Primary key yang mengidentifikasi secara unik setiap perjalanan dalam sistem
Time_ID	varchar(10)	FK yang merujuk pada tabel waktu, menghubungkan perjalanan dengan waktu terjadinya perjalanan
Pickup_Location_ID	varchar(5)	FK yang merujuk pada lokasi penjemputan, mengidentifikasi lokasi awal perjalanan
Dropoff_Location_ID	varchar(5)	FK yang merujuk pada lokasi pengantaran, mengidentifikasi lokasi tujuan pengantaran
Rider_ID	varchar(5)	FK yang merujuk pada pengguna (rider) yang melakukan perjalanan, membantu melacak riwayat pengguna

Driver_ID	varchar(5)	FK yang merujuk pada mengemudi yang membawa rider, membantu analisis performa dan riwayat driver.
Vehicle_Type_ID	varchar(5)	FK yang merujuk pada jenis kendaraan yang digunakan dalam perjalanan
Payment_Method_ID	varchar(5)	FK yang merujuk pada metode pembayaran yang digunakan dalam perjalanan
Trip_Revenue	decimal(10,2)	Mencatat pendapatan yang diperoleh dari perjalanan
Trip_Distance	decimal(5,2)	Mencatat jarak tempuh perjalanan
Wait_Time	int	Mencatat waktu tunggu antara pemesanan hingga perjalanan dimulai
Rider_Satisfaction	int	Mencatat skor kepuasan dari rider terhadap perjalanan
Driver_Satisfaction	int	Mencatat skor kepuasan dari driver terhadap perjalanan
Surge_Multiplier	decimal(4,2)	Mencatat multiplier harga
Trip_Cancellation	char(1)	Mencatat status pembatalan perjalanan(Y/N)
Batch_ID	int	Mencatat batch ID untuk keperluan ETL atau integrasi data antar sistem.
InsertedDate	date	Mencatat tanggal data dimasukkan ke dalam sistem data warehouse.
SourceSystemID	int	Mencatat ID sumber sistem yang mengirimkan data, untuk memudahkan integrasi dari berbagai sumber data.

#### D. Hubungan Antar Tabel

Dalam sistem data warehouse OKI, Transaction Fact dan Trip Fact adalah tabel fakta yang menyimpan data utama transaksi dan perjalanan yang terjadi di sistem OKI. Kedua tabel fakta berhubungan dengan tabel dimensi yang sama, yaitu:

- Time Dimension: Untuk melacak kapan transaksi atau perjalanan terjadi.
- Rider Dimension: Untuk mengidentifikasi siapa pengguna yang melakukan transaksi atau perjalanan.
- Payment Method Dimension: Untuk mencatat metode pembayaran yang digunakan.
- Location Dimension: Khusus untuk Trip Fact, digunakan untuk melacak lokasi penjemputan dan pengantaran.
- Driver Dimension dan Vehicle Dimension: Khusus untuk Trip Fact, untuk mencatat data driver dan jenis kendaraan.

Meskipun keduanya menggunakan beberapa dimensi yang sama (seperti Time, Rider, Payment), namun Transaction Fact lebih fokus ke transaksi pembayaran secara umum, sedangkan Trip Fact fokus pada perjalanan yang lebih detail termasuk jarak, pendapatan, dan kepuasan.

### Tabel ringkasan relasi

Tabel Fakta	Primary Key	Dimensi yang terhubung
Transaction Fact	Transaction_ID	Time_ID, Rider_ID, Payment_Method_ID
Trip Fact	Trip_ID	Time_ID, Rider_ID, Payment_Method_ID, Pickup_Location_ID, Dropoff_Location_ID, Driver_ID, Vehicle_Type_ID

## DESAIN FISIKAL

Desain fisikal mencakup konfigurasi penyimpanan (filegroup dan file fisik), skrip SQL untuk pembuatan objek, strategi partisi, indeks, dan pertimbangan performa untuk mendukung analitik bisnis OKI, seperti pelaporan pendapatan per zona, deteksi fraud, dan analisis kepuasan pelanggan.

### A. Konfigurasi Penyimpanan

Desain fisikal OKI\_DW memanfaatkan filegroup dan file fisik untuk mengelompokkan dan menyimpan data secara efisien:

Filegroup:

- FG\_Dimension: Digunakan untuk tabel dimensi (Time\_Dimension, Location\_Dimension, dll.) yang bersifat statis dan berukuran kecil.
- FG\_Fact: Digunakan untuk tabel fakta (Transaction\_Fact, Trip\_Fact) yang bersifat dinamis dan berukuran besar.
- Alasan: Pemisahan ini memungkinkan manajemen backup per filegroup dan optimasi performa berdasarkan jenis data.

File Fisik:

- FG\_Fact\_File: Disimpan di C:\SQLData\FG\_Fact.ndf dengan ukuran awal 10 MB, maksimum tak terbatas, dan pertumbuhan 5 MB.
- FG\_Dimension\_File: Disimpan di C:\SQLData\FG\_Dimension.ndf dengan ukuran awal 5 MB, maksimum tak terbatas, dan pertumbuhan 5 MB.
- File Database Utama: OKI\_DW.mdf (data) dan OKI\_DW\_Log.ldf (log) disimpan di C:\SQLData dengan konfigurasi awal 10 MB dan 5 MB, masing-masing.
- Alasan: Penyimpanan fisik di lokasi khusus mempermudah pemantauan dan skalabilitas.



## Skrip Pembuatan Filegroup dan File:

```
SQLQuery11.sql - ...E31L86\admin (53))* X SQLQuery10.sql - ...E31L86\admin (77))* SQLQuery9.sql - D...E31L86\admin (75))* SQLQuery7.sql - D...E31L86\admin (56))*

-- Tambah filegroup
ALTER DATABASE OKI_DW ADD FILEGROUP FG_Fact;
ALTER DATABASE OKI_DW ADD FILEGROUP FG_Dimension;
GO

-- Tambah file fisik
ALTER DATABASE OKI_DW
ADD FILE (NAME = 'FG_Fact_File', FILENAME = 'C:\SQLData\FG_Fact.ndf', SIZE = 10MB, MAXSIZE = UNLIMITED, FILEGROWTH = 5MB)
TO FILEGROUP FG_Fact;
GO

ALTER DATABASE OKI_DW
ADD FILE (NAME = 'FG_Dimension_File', FILENAME = 'C:\SQLData\FG_Dimension.ndf', SIZE = 5MB, MAXSIZE = UNLIMITED, FILEGROWTH = 5MB)
TO FILEGROUP FG_Dimension;
GO
```

## B. Skrip SQL untuk Objek

Desain fisik diimplementasikan melalui skrip SQL yang mencakup pembuatan database, tabel, partisi, dan indeks.

Pembuatan Database :

```
SQLQuery12.sql - ...E31L86\admin (62))* X SQLQuery11.sql - ...E31L86\admin (53))* SQLQuery10.sql - ...E31L86\admin (77))* SQLQuery9.sql - D...E31L86\admin (75))*

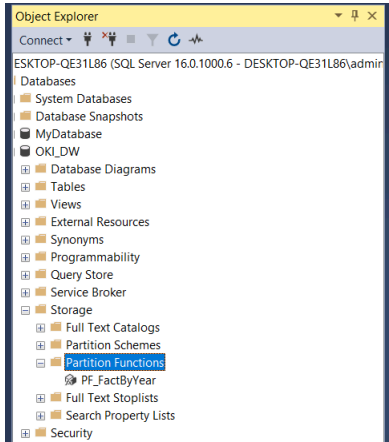
CREATE DATABASE OKI_DW
ON (NAME = OKI_DW_Data, FILENAME = 'C:\SQLData\OKI_DW.mdf', SIZE = 10MB, MAXSIZE = UNLIMITED, FILEGROWTH = 5MB)
LOG ON (NAME = OKI_DW_Log, FILENAME = 'C:\SQLData\OKI_DW_Log.ldf', SIZE = 5MB, MAXSIZE = UNLIMITED, FILEGROWTH = 5MB);
GO
USE OKI_DW;
GO
```

Pembuatan Partisi :

```
SQLQuery12.sql - ...E31L86\admin (62))* X SQLQuery11.sql - ...E31L86\admin (53))* SQLQuery10.sql - ...E31L86\admin (77))* SQLQuery9.sql - D...E31L86\admin (75))*

CREATE PARTITION FUNCTION PF_FactByYear (INT)
AS RANGE RIGHT FOR VALUES (20240101, 20250101, 20260101);
GO

CREATE PARTITION SCHEME PS_FactByYear
AS PARTITION PF_FactByYear TO (FG_Fact, FG_Fact, FG_Fact, FG_Fact);
GO
```



Pembuatan Tabel Fakta :

```
SQLQuery12.sql - ...E31L86\admin (62))* X SQLQuery11.sql - ...E31L86\admin (53))* SQLQuery10.sql - ...E31L86\admin (77))* SQLQuery9.sql - D...E31L86\admin (75))*

CREATE TABLE Transaction Fact (
    Transaction_ID INT PRIMARY KEY,
    Time_ID INT,
    Rider_ID INT,
    Payment_Method_ID INT,
    Trip_ID INT,
    Transaction_Amount DECIMAL(10,2),
    Fraud_Probability_Score INT,
    Transaction_Frequency INT,
    BatchID INT,
    InsertedDate DATE,
    SourceSystemID INT,
    CONSTRAINT FK_Transaction_Time FOREIGN KEY (Time_ID) REFERENCES Time_Dimension(Time_ID),
    CONSTRAINT FK_Transaction_Rider FOREIGN KEY (Rider_ID) REFERENCES Rider_Dimension(Rider_ID),
    CONSTRAINT FK_Transaction_Payment FOREIGN KEY (Payment_Method_ID) REFERENCES Payment_Method_Dimension(Payment_Method_ID)
) ON PS_FactByYear(Time_ID);
GO
```

	TableName	ColumnName	DataType	MaxLength	IsNullable	IsPrimaryKey	IsForeignKey
1	Transaction_Fact	Transaction_ID	int	4	0	1	0
2	Transaction_Fact	Time_ID	varchar	10	0	1	1
3	Transaction_Fact	Rider_ID	varchar	5	1	0	1
4	Transaction_Fact	Payment_Method_ID	varchar	5	1	0	1
5	Transaction_Fact	Trip_ID	int	4	1	0	0
6	Transaction_Fact	Transaction_Amount	decimal	9	1	0	0
7	Transaction_Fact	Fraud_Probability_Score	int	4	1	0	0
8	Transaction_Fact	Transaction_Frequency	int	4	1	0	0
9	Transaction_Fact	BatchID	int	4	1	0	0
10	Transaction_Fact	InsertedDate	date	3	1	0	0
11	Transaction_Fact	SourceSystemID	int	4	1	0	0

Pembuatan Indeks :

```
SQLQuery12.sql -...E31L86\admin (62))* SQLQuery11.sql -...E31L86\admin (53))* SQLQuery10.sql -...E31L86\admin (77))*
CREATE CLUSTERED INDEX IX_Transaction_Fact_Transaction_ID ON Transaction_Fact(Transaction_ID);
CREATE NONCLUSTERED COLUMNSTORE INDEX IX_Transaction_Fact_Columnstore
ON Transaction_Fact (Transaction_Amount, Fraud_Probability_Score, Transaction_Frequency);
GO
```

	IndexName	IndexType	ColumnName
1	PK_Transaction_Fact	CLUSTERED	Transaction_ID
2	PK_Transaction_Fact	CLUSTERED	Time_ID

	IndexName	IndexType	ColumnName
1	PK_TripFact	CLUSTERED	Trip_ID
2	PK_TripFact	CLUSTERED	Time_ID
3	IX_TripID	NONCLUSTERED	Trip_ID
4	IX_TripID	NONCLUSTERED	Time_ID

## C. Strategi Partisi

Partisi Berdasarkan Time\_ID:

- Menggunakan PF\_FactByYear untuk membagi data berdasarkan tahun (batas: 20240101, 20250101, 20260101).
- PS\_FactByYear menetapkan semua partisi ke FG\_Fact.

Manfaat:

- Performa: Query hanya memindai partisi relevan (misalnya, data 2025).
- Manajemen: Data lama dapat diarsipkan atau dihapus per partisi.

## D. Indeks dan Performa

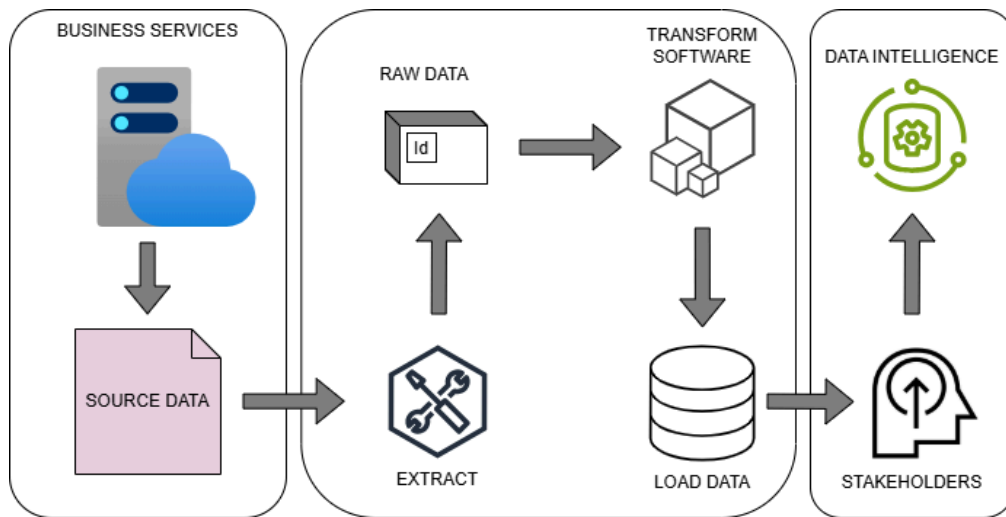
Jenis Indeks:

- Clustered Index: Pada *primary key* (misalnya, Transaction\_ID di Transaction\_Fact).
- Nonclustered Index: Pada kolom sering difilter (misalnya, Time\_ID di Trip\_Fact).
- Columnstore Index: Pada kolom analitik (misalnya, Trip\_Revenue di Trip\_Fact) untuk agregasi cepat.

Manfaat:

- Meningkatkan kecepatan query untuk laporan (misalnya, SUM pendapatan per zona).
- Mengurangi beban pemindaian tabel penuh.

## ETL PIPELINE



### A. Definisi dan Alur Proses ETL

ETL (Extract, Transform, Load) merupakan proses fundamental dalam pembangunan data warehouse seperti *OKI\_DW*. Berdasarkan diagram yang digunakan dalam penelitian ini, alur ETL dirancang untuk mengekstrak data dari sumber, mengolahnya, dan menyajikannya dalam bentuk yang siap digunakan untuk keperluan analisis bisnis. Berikut adalah rincian dari setiap tahapannya:

- **Source Data**  
Merupakan titik awal dari proses ETL. Data biasanya bersumber dari sistem operasional (OLTP) atau file eksternal seperti CSV. Dalam kasus *OKI\_DW*, sumber data berasal dari file CSV yang berisi data dummy, baik untuk tabel dimensi (contohnya: *Time\_Dimension.csv*) maupun tabel fakta (contohnya: *Transaction\_Fact.csv* dan *Trip\_Fact.csv*).
- **Extract**  
Tahap ini bertujuan untuk mengambil data mentah dari sumber dan menyimpannya di lingkungan *staging*. Proses ini mencakup pembacaan file CSV dari direktori *C:\OKI\_DW\_Project\data*.
- **Raw Data**  
Data hasil *extract* masih dalam bentuk mentah dan belum mengalami proses pembersihan atau transformasi. Data ini bersifat sementara dan akan diproses lebih lanjut pada tahap berikutnya.
- **Transform Software**  
Pada tahap ini, data diproses menggunakan perangkat lunak transformasi untuk dibersihkan, divalidasi, dan disesuaikan dengan struktur data warehouse *OKI\_DW*. Proses ini mencakup validasi foreign key serta penyesuaian format tanggal dan atribut lainnya.
- **Load Data**  
Setelah melalui tahap transformasi, data dimuat ke dalam tabel-tabel pada database *OKI\_DW*, seperti *Time\_Dimension* dan *Trip\_Fact*. Proses ini memanfaatkan partisi dan indeks yang telah dirancang sebelumnya untuk mengoptimalkan performa.
- **Data Intelligence**  
Merupakan output akhir dari pipeline ETL, yaitu data yang telah siap dianalisis dan digunakan untuk pembuatan laporan atau visualisasi, guna mendukung pengambilan keputusan bisnis yang lebih tepat.
- **Stakeholders**

Para pemangku kepentingan, seperti manajer operasional OKI atau tim deteksi kecurangan (*fraud*), memanfaatkan *data intelligence* ini untuk berbagai keperluan, termasuk pelaporan pendapatan, deteksi anomali, dan evaluasi kepuasan pelanggan