

**LAPORAN AKHIR**  
**PERANCANGAN DATA WAREHOUSE PADA BIDANG *FINTECH* ‘DompetImut’**



Disusun oleh Kelompok 27 RC :

Putri Maulida Chairani	121450050
Elilya Octaviani	122450009
Irhamna Mahdi	122450049
Randa Andriana Putra	122450083
Rafly Prabu Darmawan	122450140

**PROGRAM STUDI SAINS DATA**  
**FAKULTAS SAINS**  
**INSTITUT TEKNOLOGI SUMATERA**  
**LAMPUNG SELATAN**  
**2025**

## 1. Latar Belakang

DompelImut adalah perusahaan *fintech* yang menyediakan layanan dompet digital, terutama ditujukan untuk masyarakat urban dan generasi milenial di Indonesia. Layanan yang ditawarkan meliputi kemudahan dalam bertransaksi seperti pembayaran kepada *merchant*, transfer antar pengguna, pembelian pulsa, serta integrasi dengan *e-commerce*. Perusahaan ini memiliki visi untuk menjadi solusi keuangan digital terdepan dengan mengutamakan kecepatan, keamanan, dan kenyamanan dalam setiap transaksi. Namun, kini mereka menghadapi tantangan serius terkait integrasi data yang tersebar di berbagai sistem, yang mengakibatkan kurangnya visibilitas *real-time* dalam analitik pelanggan, efisiensi operasional, dan deteksi penipuan. Untuk terus tumbuh dan meningkatkan layanannya, DompelImut perlu membangun sistem terpusat yang dapat mengkonsolidasikan data secara efektif. Hal ini akan mendukung pengambilan keputusan yang lebih cerdas dan responsif berdasarkan data.

## 2. Tujuan dan ruang lingkup sistem

### 2.1. Tujuan

- Membangun sistem terpusat yang dapat menggabungkan data dari berbagai sumber untuk memastikan konsistensi dan kemudahan akses dalam analisis menyeluruh.
- Mengembangkan analitik lanjutan untuk memahami pola penggunaan, melakukan segmentasi pelanggan secara akurat, serta merancang strategi retensi dan promosi yang lebih efektif dan personal.
- Menyediakan sistem otomatis untuk mendeteksi anomali transaksi secara real-time dan meminimalkan risiko penipuan.
- Membangun *dashboard* interaktif dan komprehensif yang mudah diakses oleh *stakeholder* non-teknis, serta mengotomatiskan proses pelaporan untuk menghemat waktu dan meningkatkan akurasi.
- Mengembangkan infrastruktur data yang scalable dan fleksibel guna mengakomodasi lonjakan jumlah pengguna, transaksi, dan kompleksitas data seiring pertumbuhan perusahaan.

### 2.2. Ruang Lingkup Sistem

Ruang lingkup sistem dalam perancangan data warehouse DompelImut mencakup seluruh proses yang diperlukan untuk mengintegrasikan data, mengolahnya menjadi informasi yang dapat dianalisis, dan menyajikannya kepada berbagai *stakeholder* untuk mendukung pengambilan keputusan yang strategis dan berbasis data.

#### 2.2.1. Integrasi dan Transformasi Data

*Data warehouse* DompelImut dirancang untuk mengintegrasikan berbagai sumber data dari sistem internal seperti basis data transaksi, pengguna, dan layanan,

serta sumber eksternal seperti API pihak ketiga dan data geolokasi. Proses ETL (*Extract, Transform, Load*) menjadi fondasi utama dalam mengelola data ini. Data diekstraksi, kemudian dibersihkan dari duplikasi, ketidaksesuaian format, dan nilai yang hilang, serta ditambahkan atribut turunan seperti *is\_fraud* untuk mendeteksi transaksi mencurigakan. Setelah proses transformasi selesai, data dimuat ke dalam struktur yang telah ditentukan, siap untuk dianalisis dan digunakan sebagai dasar pengambilan keputusan bisnis.

### 2.2.2. Perancangan Arsitektur *Data Warehouse*

Arsitektur data warehouse DompelImut terdiri dari tiga zona utama, yaitu *Raw Zone* untuk menyimpan data mentah, *Silver Zone* untuk data yang telah dibersihkan dan dinormalisasi, dan *Gold Zone* sebagai area data yang telah dikurasi dan dimodelkan menggunakan pendekatan *Star Schema*. *Star Schema* ini terdiri dari tabel fakta (seperti *fact\_transactions*) yang menyimpan data kuantitatif transaksi, serta tabel dimensi (seperti *dim\_user*, *dim\_time*, dan *dim\_location*) yang menyimpan konteks terkait. Desain ini memungkinkan performa kueri yang optimal serta mendukung agregasi dan segmentasi data. Infrastruktur dirancang untuk mendukung skalabilitas tinggi dan memastikan ketersediaan sistem melalui mekanisme seperti partisi waktu, pemrosesan paralel, dan integrasi cloud atau hybrid environment.

### 2.2.3. Pemanfaatan Data untuk Analitik dan Pelaporan

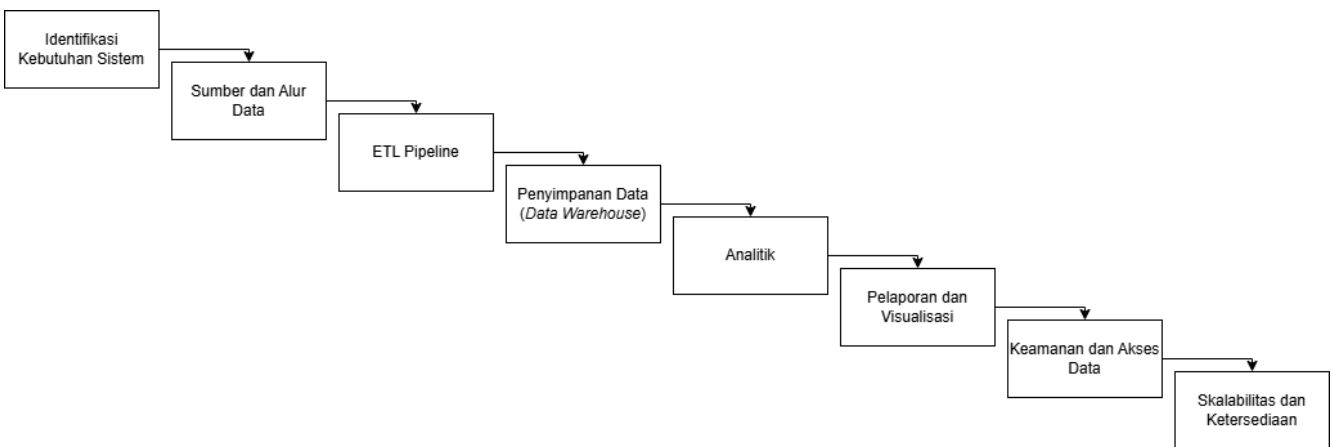
Data yang telah disimpan dalam *warehouse* digunakan untuk mendukung analitik lanjutan dan pelaporan strategis. Sistem ini memungkinkan eksplorasi data mendalam melalui analisis seperti segmentasi pelanggan, prediksi churn, deteksi fraud, dan pemodelan perilaku pengguna. Pelaporan disajikan dalam bentuk dashboard interaktif dan laporan otomatis melalui alat visualisasi seperti Power BI, yang dapat diakses oleh stakeholder non-teknis seperti manajemen, tim pemasaran, dan investor. Selain itu, sistem dilengkapi dengan kontrol keamanan ketat, termasuk otorisasi berbasis peran, enkripsi data, dan audit log, untuk memastikan akses hanya diberikan kepada pihak yang berwenang serta menjaga kerahasiaan dan integritas data pengguna.

## 3. Metodologi

### 3.1. Pendekatan Perancangan

Perancangan *data warehouse* pada *fintech* DompelImut mengikuti pendekatan *waterfall* yang sistematis dimulai dari analisis kebutuhan sistem dengan pendekatan *business/data-driven* untuk memastikan keselarasan dengan tujuan bisnis, diikuti dengan identifikasi *stakeholder* dan perumusan kebutuhan analisis. Tahap selanjutnya adalah

integrasi sumber data internal dan eksternal yang kemudian dipetakan ke dalam skema dimensi dan fakta. Proses ETL *pipeline* mencakup ekstraksi menggunakan SSIS atau Azure Data Factory, transformasi dengan Python, SQL, dan Scikit-learn, serta pemuatan ke dalam *warehouse* dengan struktur *star schema*. Penyimpanan data menggunakan arsitektur tiga lapis (*Raw, Silver, Gold*) dilengkapi partisi waktu dan indeks untuk efisiensi. Fitur analitik mendukung segmentasi, prediksi *churn*, dan deteksi *fraud* dengan bantuan model *machine learning* seperti klasifikasi dan asosiasi. Untuk penyajian informasi, sistem menyediakan dashboard visualisasi interaktif berbasis Power BI atau Tableau. Aspek keamanan dijaga melalui RBAC, enkripsi, MFA, dan audit log, sementara skalabilitas dan ketersediaan didukung oleh partisi, replikasi, elastisitas cloud, dan mekanisme *failover* otomatis.



**Gambar 1.** *Waterfall* perencanaan DompelImut

### 3.2. Tahapan Perancangan

#### 3.2.1. Analisis dan Identifikasi Kebutuhan Sistem

Pada perancangan DompelImut, memastikan perancangan sejalan dengan kebutuhan bisnis serta memastikan teknis dan struktur data terealisasi sesuai dengan sistem dari internal maupun eksternal.

**Tabel 1.** Kebutuhan sistem sesuai tujuan bisnis

Stakeholder	Tujuan Bisnis
Manajemen Eksekutif	Mengamati performa portofolio pinjaman secara keseluruhan untuk mendukung pengambilan keputusan strategis yang tepat dan berdampak besar
Tim Risiko & Kepatuhan	Memahami pola risiko dalam portofolio pinjaman untuk

	menjaga kelangsungan bisnis yang aman dan sesuai regulasi
Tim <i>Marketing</i>	Menciptakan strategi promosi yang lebih terarah dengan memanfaatkan analisis demografi dan perilaku konsumen dari Data Warehouse
Data Analyst	Menggali wawasan mendalam tentang tren pinjaman dari preferensi pengguna hingga faktor geografis untuk membantu bisnis berkembang
Pengembang Produk	Menyusun produk pinjaman yang inovatif dan relevan berdasarkan data pasar yang terintegrasi
Investor & Pemangku Kepentingan	Menilai kesehatan finansial perusahaan serta potensi pertumbuhan untuk pengambilan keputusan investasi yang lebih terpercaya

Dalam perancangan data warehouse DompetImut, sumber data berasal dari sistem operasional internal yang mencakup tabel pengguna, transaksi, dan layanan, serta data eksternal yang diperoleh melalui API dari penyedia layanan transaksi. Tujuan utama dari perancangan ini adalah untuk meningkatkan keuntungan perusahaan dengan menganalisis aktivitas pengguna, seperti tingkat *churn*, identifikasi transaksi mencurigakan, deteksi pengguna potensial, serta pola kebiasaan transaksi guna mendukung strategi rekomendasi. Analisis ini diharapkan dapat meningkatkan retensi pengguna, mengurangi risiko penipuan, mendorong penggunaan fitur premium, dan meningkatkan efisiensi operasional. Berdasarkan tujuan bisnis tersebut, diturunkan sejumlah kebutuhan analisis operasional seperti identifikasi pengguna yang tidak bertransaksi dalam satu tahun terakhir, perbedaan pola transaksi antara pengguna aktif dan tidak aktif, jumlah serta karakteristik transaksi mencurigakan, rata-rata saldo pengguna layanan premium, dan biaya transaksi ke berbagai layanan. Untuk mendukung analisis multidimensi ini, dibangun sebuah tabel fakta utama yang merekam ukuran-ukuran seperti jumlah transaksi, biaya transaksi, dan jumlah transaksi mencurigakan, yang terhubung ke berbagai tabel dimensi yang mencerminkan atribut-atribut seperti pengguna, waktu, layanan, dan jenis transaksi, lengkap dengan hierarki dan derivasi yang relevan.

**Tabel 2.** Informasi dimensi atribut pengguna

<i>Facts</i>	<i>Measure</i>	<i>Dimensions</i>	<i>Hierarchies</i>
Transaksi	Jumlah transaksi, Biaya transaksi, transaksi mencurigakan	Pengguna (1:M)	Pengguna → Segmentasi (Basic/Premium)
		Layanan (1:M)	Layanan → Kategori

		Waktu (1:M)	Detik → Tanggal → Bulan → Tahun
		Tempat (1:M)	Kota/Kabupaten → Provinsi → Negara
		Transaksi deteksi (1:M)	Penanda penipuan → Jenis penipuan

### 3.2.2. Sumber dan Alur Data

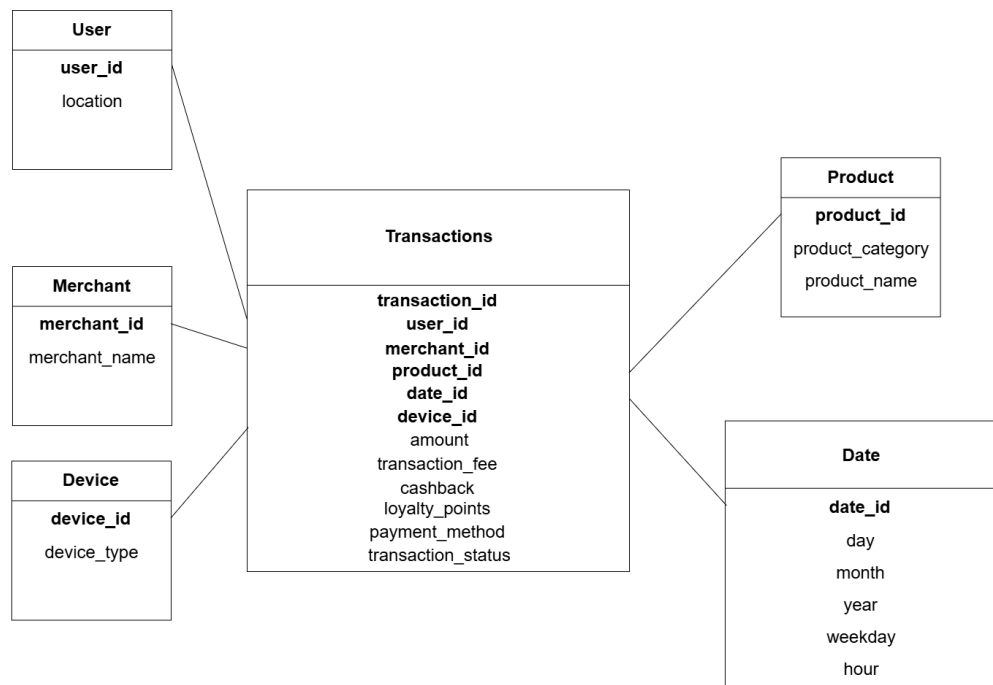
Sumber data yang digunakan DompelImut memuat berbagai atribut penting seperti informasi transaksi, pengguna, merchant, produk, perangkat, dan waktu. Untuk proses pencocokan dilakukan dengan membandingkan elemen-elemen dalam skema yang dirancang dengan atribut-atribut yang tersedia dalam dataset. Hasil evaluasi menunjukkan bahwa seluruh komponen penting yang diperlukan untuk membentuk tabel fakta dan tabel dimensi tersedia secara eksplisit atau dapat diturunkan (*derived*) dari atribut yang ada. Dengan kata lain, skema yang dirancang memiliki kesesuaian yang tinggi dengan sumber data, sehingga dapat langsung diimplementasikan untuk kebutuhan analisis lebih lanjut, seperti analisis perilaku pengguna, efisiensi transaksi, hingga efektivitas program loyalitas. Data ini terdiri dari beberapa fitur penting yang menjelaskan secara menyeluruh setiap transaksi. Atribut dalam dataset mencakup identitas pengguna dan merchant, rincian produk atau layanan yang dibeli, nominal transaksi, biaya, cashback, poin loyalitas, serta informasi perangkat dan lokasi.

Beberapa potensi analisis yang dapat dilakukan menggunakan data ini yaitu analisis pola pengeluaran berdasarkan kategori produk, mengkaji efektivitas program cashback dan loyalitas, menyelidiki hubungan antara metode pembayaran dan tingkat keberhasilan transaksi, mengeksplorasi tren musiman dalam penggunaan dompet digital, mengembangkan model deteksi penipuan berdasarkan pola transaksi, melakukan segmentasi pengguna berdasarkan perilaku pengeluaran, serta menganalisis popularitas berbagai merchant dan layanan.

**Tabel 3.** Tabel fakta

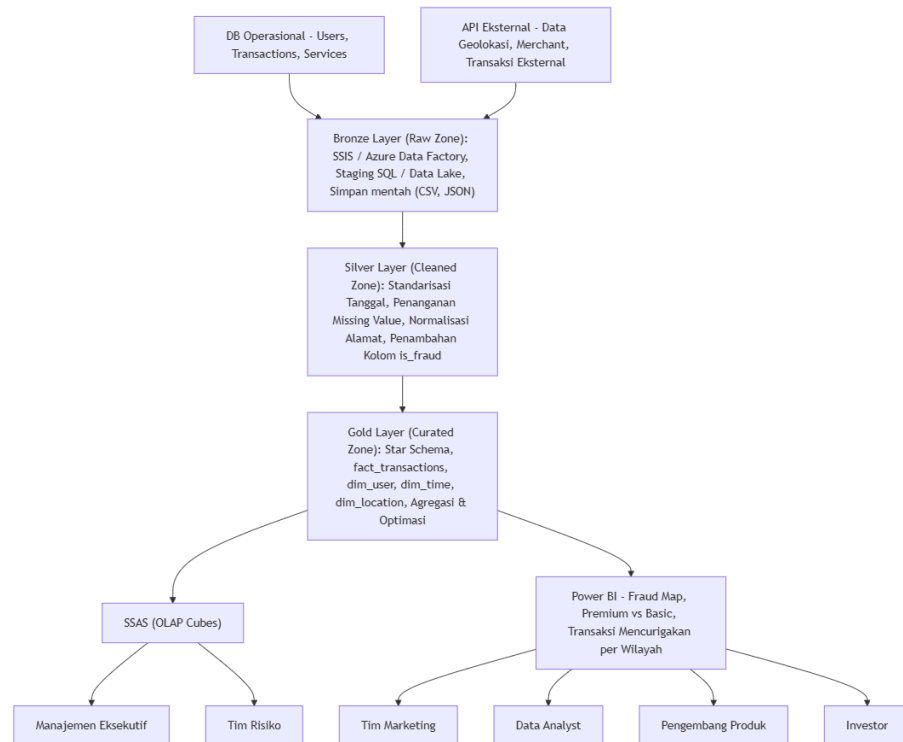
Kolom	Tipe Data	Deskripsi
idx	Integer	Nomor indeks unik untuk setiap entri atau record dalam dataset
transaction_id	String (UUID)	ID unik berbentuk UUID untuk mengidentifikasi setiap transaksi
user_id	String	ID unik pengguna yang melakukan transaksi

transaction_date	Datetime	Tanggal dan waktu saat transaksi dilakukan
product_category	String	Kategori dari produk atau layanan yang dibeli
product_name	String	Nama produk atau layanan spesifik yang dibeli
merchant_name	String	Nama merchant atau penyedia layanan tempat transaksi dilakukan
product_amount	Float	Jumlah nominal dari transaksi dalam mata uang lokal
transaction_fee	Float	Biaya yang dikenakan untuk transaksi tersebut
cashback	Float	Jumlah cashback yang diterima pengguna dari transaksi
loyalty_points	Integer	Jumlah poin loyalitas yang diperoleh dari transaksi
payment_method	String	Metode pembayaran yang digunakan (misalnya: e-wallet, kartu kredit, QRIS)
transaction_status	String	Status transaksi: Successful, Failed, atau Pending
merchant_id	String	ID unik dari merchant terkait transaksi
device_type	String	Jenis perangkat yang digunakan untuk melakukan transaksi (misal: mobile, web)
location	String	Lokasi transaksi secara umum (misalnya: kota, provinsi, atau wilayah besar)



**Gambar 2.** *Star Schema* perencanaan DompetImut

Aliran data menggambarkan proses transformasi data dari sumber mentah menjadi siap digunakan untuk analisis bisnis. Diagram ini secara rinci menjelaskan alur untuk data warehouse DompetImut dengan mengacu pada arsitektur dan metode ETL (Extract, Transform, Load).



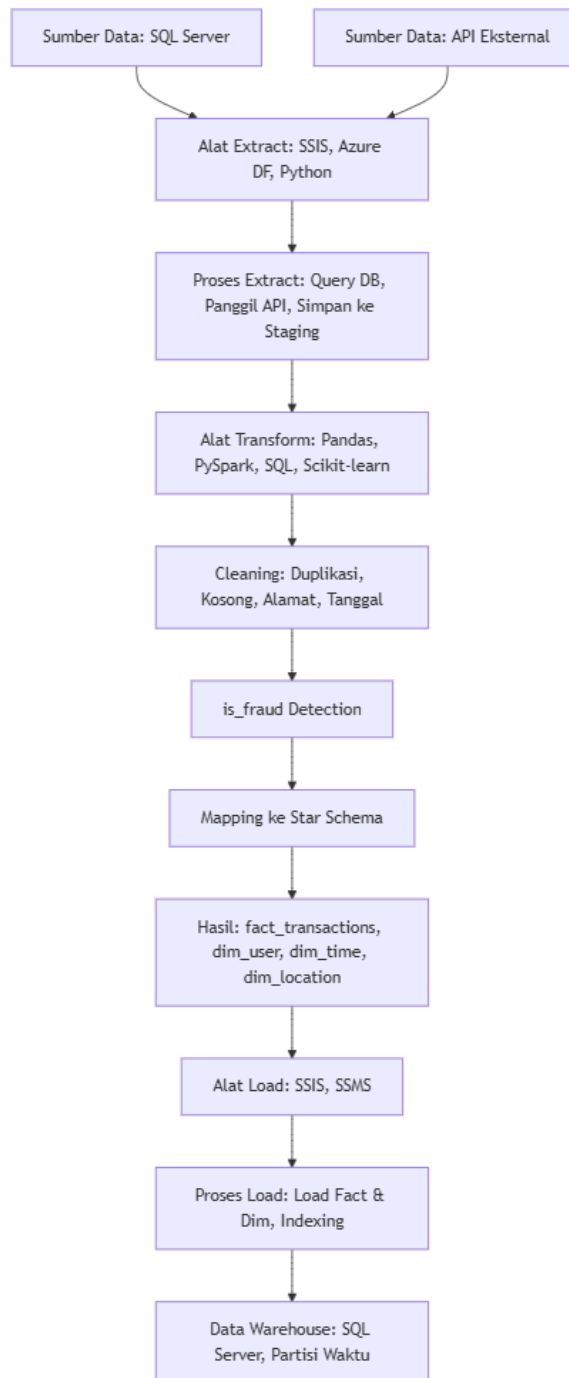
**Gambar 3.** Alur Data

Aliran data DompetImut dimulai dari dua sumber utama, yaitu basis data operasional (tabel Pengguna, Transaksi, Layanan) dan API eksternal (data geolokalisasi, pedagang dan transaksi pihak ketiga). Data dari kedua sumber ini diekstrak melalui SSIS dan kemudian disimpan dalam format mentah di Zona Mentah tanpa modifikasi. Di *Silver Layer* (Zona Bersih), data dibersihkan dan diubah. Proses ini termasuk menormalkan format, menangani data yang hilang, menormalkan alamat, dan menambahkan atribut seperti is\_fraud untuk mendeteksi transaksi yang mencurigakan. Data yang telah dibersihkan dan terstruktur diproses ke *Gold Layer (Curated Zone)* menggunakan model Star Schema, dengan tabel fakta (fact\_transactions) dan tabel dimensi (dim\_user, dim\_time, dim\_location). Fase ini juga mencakup agregasi dan optimisasi, seperti partisi waktu dan pembuatan indeks. Dari *Gold Layer*, data digunakan oleh SSAS untuk analisis multidimensi dan oleh Power BI untuk visualisasi interaktif seperti peta transaksi yang mencurigakan dan analisis perilaku pengguna. Hasil analisis ini menjadi dasar pengambilan keputusan



oleh berbagai pihak, seperti manajemen, tim risiko, pemasaran, analis data, pengembang produk, dan investor, sesuai dengan kebutuhan strategis masing-masing.

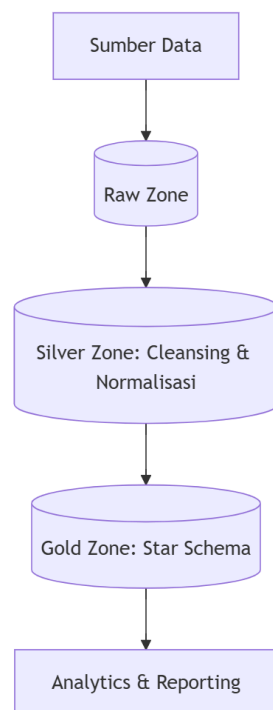
### 3.2.3. ETL (*Extract, Transform, Load*) Pipeline



**Gambar 4.** Proses ETL

Diagram tersebut menggambarkan alur proses perancangan data warehouse DompelImut dari tahap ekstraksi hingga penyimpanan akhir. Data diambil dari dua sumber utama, yaitu SQL Server dan API eksternal, menggunakan alat seperti SSIS, Azure Data Factory, dan Python. Proses ekstraksi dilakukan dengan mengeksekusi query database dan memanggil API untuk menyimpan data ke dalam area staging. Selanjutnya, data ditransformasikan menggunakan Pandas, PySpark, SQL, dan Scikit-learn, termasuk proses cleaning (menghapus duplikasi, data kosong, dan memperbaiki alamat serta tanggal) serta deteksi *fraud*. Setelah itu, data dimapping ke dalam star schema, menghasilkan tabel-tabel. Proses pemuatan (*load*) dilakukan menggunakan SSIS dan SSMS, yang mencakup pemuatan data ke tabel fakta dan dimensi serta pembuatan indeks. Akhirnya, data disimpan dalam data warehouse berbasis SQL Server dengan partisi waktu untuk mendukung efisiensi kueri dan skalabilitas.

#### 3.2.4. Penyimpanan Data (*Data Warehouse*)



**Gambar 5.** Tempat penyimpanan data

Gambar tersebut menunjukkan arsitektur lapisan *data warehouse* yang terdiri dari tiga zona utama, yaitu *Raw Zone*, *Silver Zone*, dan *Gold Zone*. Data pertama kali diambil dari berbagai sumber data, lalu disimpan dalam *Raw Zone*, yaitu area penyimpanan mentah tanpa proses pembersihan. Selanjutnya, data diproses di *Silver Zone* yang berfokus pada *cleansing* (pembersihan data dari duplikasi, kesalahan, atau

inkonsistensi) dan normalisasi (penyesuaian struktur agar konsisten dan seragam). Setelah itu, data yang telah bersih dan terstruktur dimasukkan ke dalam *Gold Zone*, tempat data dimodelkan dalam bentuk star schema untuk mendukung kebutuhan analitik. Zona akhir ini digunakan untuk keperluan *analytics & reporting*, sehingga pengguna bisnis dapat memperoleh *insight* dari data yang telah diolah dengan akurat dan efisien.

### 3.2.5. Analitik

Fitur analitik dalam sistem data warehouse DompetImut dirancang untuk mendukung eksplorasi data mendalam guna memahami perilaku pengguna, pola transaksi, dan tren finansial secara komprehensif. Dengan memanfaatkan teknik seperti *clustering*, *classification*, dan *association rule mining*, sistem ini dapat digunakan untuk segmentasi pelanggan berdasarkan perilaku transaksi, prediksi *churn* (kehilangan pelanggan), serta analisis risiko penipuan. Selain itu, analitik ini memungkinkan perumusan strategi pemasaran yang lebih tepat sasaran, rekomendasi produk atau layanan personalisasi, dan pengambilan keputusan berbasis data yang lebih akurat.

### 3.2.6. Pelaporan dan Visualisasi

Sistem ini mencakup modul pelaporan dan visualisasi data yang mendukung pembuatan laporan otomatis dan dashboard interaktif. Laporan mencakup metrik utama seperti volume transaksi, pertumbuhan pengguna, segmentasi wilayah, serta efektivitas kampanye promosi. Visualisasi disajikan melalui grafik, peta, dan tabel yang mudah dipahami, dirancang agar bisa digunakan oleh stakeholder non-teknis seperti tim manajemen, pemasaran, dan operasional. Dengan integrasi ke *tools* seperti Power BI atau Tableau, pengguna dapat melakukan eksplorasi data secara *real-time*, menyaring informasi sesuai kebutuhan, dan memperoleh *insight* tanpa ketergantungan langsung pada tim IT.

### 3.2.7. Keamanan dan Akses Data

Dalam perancangan data warehouse DompetImut, keamanan data menjadi prioritas utama, mengingat sensitivitas data keuangan dan identitas pengguna. Sistem ini menerapkan autentikasi dan otorisasi berbasis peran (*role-based access control*), enkripsi data baik saat transit maupun saat disimpan, serta pemantauan akses melalui audit log yang terekam secara sistematis. Protokol keamanan tambahan seperti multi-factor authentication (MFA), pembatasan akses berdasarkan lokasi/IP, dan pengelolaan hak akses granular juga diterapkan untuk mengurangi risiko penyalahgunaan data oleh pihak internal maupun eksternal.

### 3.2.8. Skalabilitas dan Ketersediaan

Arsitektur data warehouse dirancang untuk *scalable* dan *highly available*, sehingga dapat menangani pertumbuhan data yang masif akibat meningkatnya jumlah transaksi dan pengguna DompelMut dari waktu ke waktu. Sistem mendukung partisi data berdasarkan waktu, pemrosesan paralel, serta penggunaan infrastruktur *cloud* atau *hybrid* yang elastis. Ketersediaan tinggi dijamin melalui replikasi data, mekanisme *failover* otomatis, serta *monitoring* kinerja dan beban sistem secara *real-time*. Hal ini memastikan layanan analitik tetap responsif dan andal meskipun terjadi lonjakan aktivitas atau gangguan teknis.

## 3.3. Alat dan Teknologi

### 3.3.1. SQL Server Integration Service (SSIS)

SSIS adalah alat utama dalam proses Extract, Transform, dan Load (ETL) yang digunakan dalam pengembangan data warehouse DompelMut. SSIS memungkinkan ekstraksi data dari berbagai sumber data operasional seperti SQL Server dan mendukung pengambilan data dari API eksternal melalui komponen script. Selain itu, SSIS memungkinkan proses transformasi data seperti pembersihan, penghapusan duplicate, dan pengolahan data menjadi format yang lebih terstruktur. Selanjutnya, hasil transformasi dimuat ke area pengujian atau langsung ke data warehouse. SSIS memungkinkan penggunaan SQL Server Agent untuk membuat paket ETL yang terjadwal secara otomatis.

### 3.3.2. Database Engine

Pengelolaan dan penyimpanan data warehouse DompelMut bergantung pada Database Engine. Struktur data dirancang untuk implementasi menggunakan model skema bintang. Di dalam model ini, terdapat tabel fakta utama yang disebut `fact_transactions` yang menyimpan semua data transaksi yang telah diproses melalui proses ETL, termasuk atribut transaksi, jumlah, dan jenis pembayaran, serta flag yang mendeteksi transaksi penipuan (`is_fraud`). Beberapa tabel dimensi terhubung ke tabel fakta seperti `dim_user` menyimpan informasi pengguna lengkap, `dim_time` menyimpan data dimensi tanggal, dan `dim_location` menyimpan data dimensi lokasi. Database Engine juga memiliki fitur partisi data berdasarkan waktu transaksi, yang memungkinkan query dilakukan lebih cepat karena data dapat difokuskan pada rentang waktu tertentu tanpa memindai tabel secara keseluruhan. Selain itu, kolom-kolom penting seperti `transaction_id`, `user_id`, dan `date_id` diindeks untuk mempercepat pencarian dan analisis data. Untuk memastikan data warehouse DompelMut tetap aman dan tersedia meskipun terjadi gangguan sistem, fitur backup

dan restore Database Engine juga digunakan. Database Engine mengatur semua pengaturan keamanan akses data.

### 3.3.3. SQL Server Analysis Service (SSAS)

SSAS digunakan untuk memetakan data yang telah disimpan di data warehouse ke dalam cube. Cube ini memiliki tabel fakta fact\_transactions yang berfungsi sebagai pusat data transaksi, dan tabel dimensi seperti dim\_user, dim\_time, dan dim\_location yang membantu menganalisis data berdasarkan dimensi, waktu, dan lokasi pengguna. SSAS juga memungkinkan pembuatan kalkulasi agregat, seperti total transaksi per jenis pembayaran atau per lokasi, serta hierarki waktu seperti tahun, kuartal, bulan, dan hari. Selain itu, SSAS memungkinkan integrasi fitur deteksi pola transaksi mencurigakan (is\_fraud) sebagai indikator analitik. Cube kemudian digunakan sebagai sumber data untuk Power BI dan Metabase untuk mendukung visualisasi dan proses pengambilan keputusan tim risiko, pemasaran, dan manajemen.

### 3.3.4. SQL Server Reporting Services (SSRS)

Laporan interaktif yang dibuat dengan SSRS dapat diakses secara online melalui portal internal organisasi DompetImut atau secara berkala dikirimkan melalui email kepada pihak yang bertanggung jawab. SSRS digunakan untuk membuat laporan seperti rekapitulasi transaksi bulanan, pengenalan jumlah transaksi mencurigakan per kota, dan laporan performa pengguna aktif bulanan. Dengan menggunakan query SQL yang telah dioptimalkan, laporan ini dibuat dengan mengambil data langsung dari tabel-tabel data warehouse dan dari cube SSAS. Dengan SSRS, DompetImut dapat menyediakan laporan yang terstandarisasi, akurat, dan mudah dipantau untuk membantu proses audit internal dan evaluasi performa bisnis.

### 3.3.5. SQL Server Management Studio (SSMS)

Seluruh infrastruktur data warehouse DompetImut dikelola oleh SQL Server Management Studio (SSMS). SSMS digunakan untuk menjalankan query SQL yang kompleks untuk memantau kualitas data transaksi, mengelola indeks dan partisi pada tabel fact\_transactions, dan menangani kesalahan proses ETL atau kegagalan pemuatan data. Selain itu, SSMS juga digunakan untuk mengelola pembantu SQL Server, yang menjadwalkan eksekusi otomatis untuk memproses data transaksi baru dari API eksterna dan sistem operasional. SSMS juga memudahkan proses pengawasan kinerja query dan penggunaan resource server dalam mengoptimalkan kinerja sistem data warehouse agar tetap responsif dan stabil meskipun beban data terus meningkat.

## 4. Desain konseptual, logikal, dan fisik

### 4.1 Desain Konseptual

Desain Konseptual merupakan tahap awal dalam perancangan data warehouse yang berfokus pada identifikasi entitas utama, hubungan antar entitas, serta fakta dan dimensi yang relevan. Pendekatan yang digunakan dalam sistem DompelImut adalah *star schema*, karena strukturnya yang lebih sederhana, efisien dalam query analitik, dan juga mudah dikembangkan.

#### 4.1.1 Desain Tabel Dimensi

Tabel dimensi berfungsi sebagai referensi dalam sistem data warehouse, menyimpan data kontekstual atau atribut non-numerik. Dalam proyek DompelImut, tabel dimensi mencerminkan berbagai entitas yang relevan terhadap analisis perilaku transaksi pengguna. Tabel-tabel dimensi yang digunakan antara lain:

**Tabel 4.** Tabel Dimensi

Tabel Dimensi	Kolom Utama	Deskripsi	Hierarki (Jika Ada)
User	user_id, location	Identitas pengguna dan lokasi geografis	Lokasi: City → Province → Country
Merchant	merchant_id, merchant_name	Merchant tempat pengguna bertransaksi	-
Product	product_id, product_category, product_name	Produk atau layanan yang dibeli	Nama Produk → Kategori Produk
Device	device_id, device_type	Jenis perangkat yang digunakan untuk transaksi	-
Date	date_id, Day, month, year, weekday, Hour	Informasi waktu lengkap transaksi	Hour → Day → Weekday → Month → Year

#### 4.1.2 Desain Tabel Fakta

Tabel Fakta merupakan tabel yang merekam seluruh transaksi digital pengguna. Tabel fakta menyimpan data numerik dan hubungan foreign key ke dimensi. Tabel utama adalah tabel transaksi, yang mencatat seluruh aktivitas keuangan yang dilakukan pengguna. Setiap baris dalam tabel fakta merepresentasikan satu peristiwa transaksi. Desain tabel fakta dapat dilihat pada **Tabel 3**.

#### 4.1.3 Justifikasi Pemilihan Skema

Dalam perancangan DompelMut, skema yang dipilih adalah *Star Schema* karena karakteristiknya yang sederhana dan cepat dalam eksekusi query. Setiap dimensi langsung terhubung ke tabel fakta tanpa melalui normalisasi lebih lanjut.

**Tabel 5.** Aspek *Star Schema*

Aspek	<i>Star Schema</i>
Struktur	Tabel fakta terhubung langsung ke tabel dimensi
Kinerja Query	Cepat karena join minimal sehingga memudahkan eksplorasi data
Fleksibilitas	Mudah dikembangkan dengan penambahan dimensi baru
Kesesuaian dengan DompelMut	Struktur data dari sistem operasional sudah mendukung segmentasi berdasarkan entitas seperti waktu, perangkat, produk, dan lokasi.

#### 4.2 Desain Logikal

Desain logikal adalah tahap dalam perancangan data warehouse yang mendefinisikan bagaimana entitas data diatur secara logis dalam bentuk tabel dan hubungan antar tabel. Di tahap ini, data dimodelkan tanpa memperhatikan detail teknis penyimpanan, fokus pada struktur relasional, kunci utama dan asing, serta hubungan antar entitas.

##### 4.2.1 Relasi Entitas

Tabel fakta Transactions terhubung dengan lima tabel dimensi utama melalui kolom foreign key. Setiap transaksi hanya merujuk pada satu entitas dalam masing-masing dimensi, yang menciptakan relasi Many-to-One dari fakta ke dimensi.

**Tabel 6.** Relasi Entitas pada Tabel Fakta

Tabel Fakta	Kolom FK	Tabel Dimensi	Jenis Relasi
-------------	----------	---------------	--------------

Transactions	user_id	User	Many-to-One
Transactions	merchant_id	Merchant	Many-to-One
Transactions	product_id	Product	Many-to-One
Transactions	device_id	Device	Many-to-One
Transactions	date_id	Date	Many-to-One

#### 4.2.2 Primary Key dan Relasi

- Setiap tabel dimensi memiliki kunci utama (\*\_id).
- Tabel fakta memiliki kunci utama transaction\_id.
- Semua foreign key pada tabel fakta mengarah ke primary key tabel dimensi terkait.
- Skema ini menjamin integritas data dan memudahkan implementasi query OLAP.

### 4.3 Desain Fisikal

Desain fisik adalah tahap yang menentukan bagaimana data disimpan dan diakses secara teknis di dalam sistem basis data. Ini mencakup pemilihan format file, indexing, partisi tabel, struktur penyimpanan (layer), dan strategi optimasi query. Tujuannya adalah memaksimalkan kinerja, efisiensi ruang, serta skalabilitas sistem. Pada DompetImut, desain fisik dirancang dengan memperhatikan volume transaksi digital yang tinggi serta kebutuhan analitik yang responsif.

#### 4.3.1 Indexing

- Clustered index pada transaction\_id untuk mengakses data transaksi dengan cepat.
- Non-clustered index pada kolom seperti user\_id, date\_id, dan is\_fraud untuk mempercepat filtering dan join.

#### 4.3.2 Partisi

- Tabel Transactions dipartisi berdasarkan date\_id (range partisi waktu).
- Partisi mempercepat analisis berbasis periode dan memudahkan pengelolaan data historis.

#### 4.3.3 Struktur Penyimpanan Layered

- Staging Area: menyimpan data mentah dari berbagai sumber.
- Clean/Silver Layer: menyimpan data hasil pembersihan dan transformasi.
- Curated/Gold Layer: menyimpan data akhir dalam bentuk star schema siap analisis.



#### 4.3.4 Platform

- Data disimpan di SQL Server Data Warehouse.
- Proses ETL menggunakan SSIS, integrasi analitik menggunakan SSAS, dan visualisasi melalui Power BI.

Desain logikal menunjukkan keterkaitan antara entitas dalam bentuk relasi many-to-one dari fakta ke dimensi, sehingga memungkinkan eksplorasi dan segmentasi data secara fleksibel. Desain fisik memperkuat implementasi dengan penggunaan index pada kolom penting, strategi partisi berdasarkan waktu untuk optimasi performa, serta struktur penyimpanan berlapis untuk mendukung ETL yang efisien.

**Tabel 7.** Perbedaan Desain Logikal dan Fisik

Aspek	Desain Logikal	Desain Fisik
Fokus	Struktur relasional dan hubungan antar entitas	Implementasi teknis penyimpanan dan akses data
Bentuk Data	Tabel logis dan relasi (FK, PK)	Index, partisi, file format, storage layer
Teknologi Terkait	Skema entitas-relasi	SQL Server, SSIS, indexing, OLAP, partitioning
Tujuan	Menjamin integritas dan kejelasan hubungan data	Mengoptimalkan performa query dan efisiensi penyimpanan

## 5. Proses implementasi

### Realisasi Struktur Tabel dan Proses ETL

Implementasi skema bintang direalisasikan dengan SQL DDL sebagai berikut :

```
-- Tabel Dimensi: Pengguna (User)
CREATE TABLE dim_user (
  user_id INT PRIMARY KEY,
  user_name VARCHAR(100),
  user_email VARCHAR(100),
  user_phone VARCHAR(20)
  -- kolom tambahan jika perlu
);

-- Tabel Dimensi: Waktu (Time)
CREATE TABLE dim_time (
  time_id INT PRIMARY KEY,
```

```

    date_value DATE,
    year INT,
    month INT,
    day INT,
    quarter INT
    -- kolom lain (mis. nama hari, minggu)
);

-- Tabel Dimensi: Lokasi (Location)
CREATE TABLE dim_location (
    location_id INT PRIMARY KEY,
    city VARCHAR(100),
    province VARCHAR(100),
    country VARCHAR(100)
    -- kolom lain (mis. kode pos)
);

-- Tabel Fakta: Transaksi
CREATE TABLE fact_transactions (
    transaction_id INT PRIMARY KEY,
    user_id INT,
    time_id INT,
    location_id INT,
    amount DECIMAL(18,2),
    is_fraud BIT,
    -- kolom fakta tambahan (mis. jenis transaksi)
    FOREIGN KEY (user_id) REFERENCES dim_user(user_id),
    FOREIGN KEY (time_id) REFERENCES dim_time(time_id),
    FOREIGN KEY (location_id) REFERENCES dim_location(location_id)
);

```

## **DompelImut diproses melalui tiga tahapan utama:**

### **(a) Extract**

Pada tahap ini, data dikumpulkan dari berbagai sumber untuk disiapkan masuk ke dalam sistem data warehouse. Data diekstraksi dari sumber utama seperti file CSV atau API eksternal, yang mencakup informasi pengguna, layanan, transaksi, lokasi, serta data tambahan dari mitra pihak ketiga.

Sumber data mencakup:

- Database internal DompelImut, seperti tabel transaksi, pengguna, merchant, dan produk.
- File CSV eksternal yang berisi riwayat transaksi digital dengan 5000 baris data sintetis.
- API eksternal, misalnya untuk informasi geolokasi, perangkat pengguna, atau merchant pihak ketiga.

Data diekstrak menggunakan Python, SQL, dan tools ETL seperti SSIS (SQL Server Integration Services). Dalam script ekstraksi, dilakukan koneksi ke database sumber, pembacaan file CSV, serta penarikan data API menggunakan modul requests.

### **(b) Transform**

Data yang telah diekstrak kemudian dibersihkan, dinormalisasi, dan diklasifikasikan ke dalam entitas-entitas seperti pengguna, waktu, dan lokasi untuk membentuk tabel dimensi. Proses ini juga melibatkan penggabungan data, parsing tanggal, penambahan atribut *is\_fraud*, serta agregasi data transaksi untuk disusun dalam struktur skema bintang. Transformasi dilakukan agar data yang tadinya mentah menjadi bersih, seragam, dan siap dianalisis.

Langkah-langkah transformasi meliputi:

- Pembersihan Data: Menghilangkan duplikasi, memperbaiki format tanggal, menangani nilai kosong (NULL), dan validasi tipe data.
- Normalisasi Atribut: Misalnya, menyamakan penulisan metode pembayaran (e-wallet, Ewallet, E-wallet) menjadi satu format standar.
- Derivasi Kolom Baru:

- `is_fraud`: kolom baru untuk mendeteksi anomali berdasarkan aturan sederhana, seperti transaksi di luar lokasi biasa, nilai transaksi tidak wajar, atau pergantian perangkat.
- `date_id`: ID waktu unik yang dikonversi dari timestamp transaksi ke dimensi waktu.
- Mapping: Menghubungkan setiap baris transaksi ke entitas dimensi seperti `user_id`, `product_id`, `merchant_id`, dan `device_id`.

Transformasi ini dilakukan di SQL Server menggunakan skrip SQL dan di Python (jika perlu pemrosesan lanjutan seperti flagging fraud).

### (c) Load

Setelah transformasi selesai, data dari masing-masing entitas dimuat ke dalam tabel dimensi, kemudian data transaksi dimuat ke dalam tabel fakta dengan relasi ke tabel dimensi melalui kunci asing, guna memastikan konsistensi dan integritas data dalam data warehouse. Setelah data bersih dan terstruktur, data dimuat ke dalam SQL Server Data Warehouse yang telah didesain menggunakan skema bintang (Star Schema).

Tahapan loading meliputi:

- Memasukkan data ke dalam tabel dimensi: `dim_user`, `dim_merchant`, `dim_product`, `dim_device`, `dim_date`.
- Memasukkan data transaksi ke dalam tabel fakta: `fact_transactions`.
- Penerapan indexing dan partisi saat proses loading untuk memastikan efisiensi akses data dan kecepatan query.

## 6. Hasil Implementasi

### 6.1 Implementasi data warehouse di SQL Server

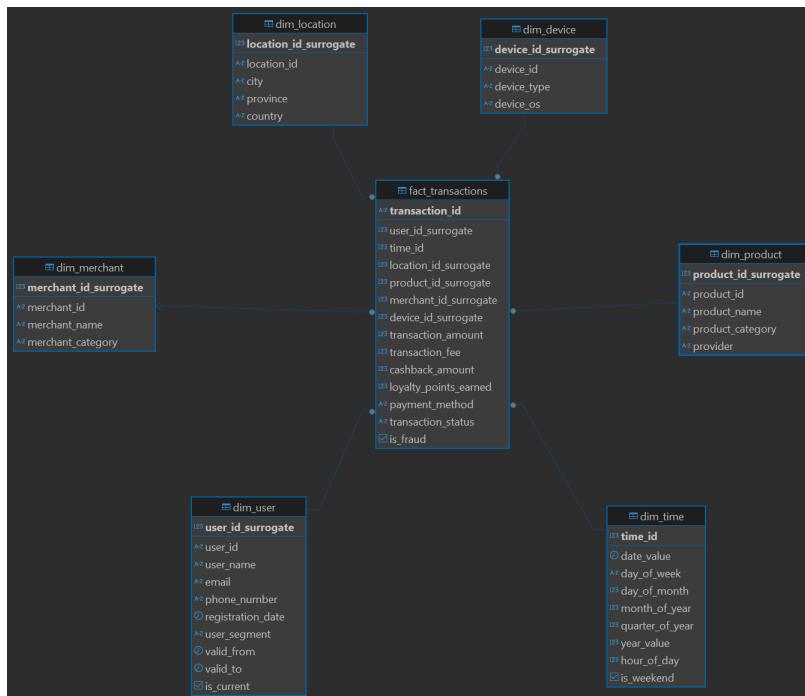
Pembuatan tabel fakta dan dimensi dengan script SQL di dalam skema database yang dibuat:

```

--CREATE TABLE dim_location (
1  location_id INT PRIMARY KEY IDENTITY(1,1),
2  location_id_surrogate INT PRIMARY KEY IDENTITY(1,1),
3  city VARCHAR(100),
4  province VARCHAR(100),
5  country VARCHAR(100) DEFAULT 'Indonesia'
6 );
7
8 --CREATE TABLE dim_user (
9  user_id INT PRIMARY KEY IDENTITY(1,1),
10 user_id_surrogate INT PRIMARY KEY IDENTITY(1,1),
11 user_name VARCHAR(255) NOT NULL UNIQUE,
12 email VARCHAR(255) NOT NULL,
13 phone_number VARCHAR(20) UNIQUE,
14 registration_date DATE,
15 user_segment VARCHAR(10),
16 valid_from DATETIME DEFAULT GETDATE(),
17 valid_to DATETIME DEFAULT '9999-12-31',
18 is_current BIT DEFAULT 1
19 );
20
21 --CREATE TABLE dim_device (
22 device_id INT PRIMARY KEY IDENTITY(1,1),
23 device_id_surrogate INT PRIMARY KEY IDENTITY(1,1),
24 device_type VARCHAR(255) NOT NULL UNIQUE,
25 device_os VARCHAR(255) NOT NULL
26 );
27
28 --CREATE TABLE dim_product (
29 product_id INT PRIMARY KEY IDENTITY(1,1),
30 product_id_surrogate INT PRIMARY KEY IDENTITY(1,1),
31 product_name VARCHAR(255) NOT NULL UNIQUE,
32 product_category VARCHAR(100),
33 provider VARCHAR(100)
34 );
35
36 --CREATE TABLE fact_transactions (
37 transaction_id INT PRIMARY KEY IDENTITY(1,1),
38 user_id_surrogate INT,
39 time_id INT,
40 location_id_surrogate INT,
41 product_id_surrogate INT,
42 merchant_id_surrogate INT,
43 device_id_surrogate INT,
44 transaction_amount DECIMAL(15,2),
45 transaction_fee DECIMAL(15,2),
46 cashback_amount DECIMAL(15,2),
47 loyalty_points_earned DECIMAL(15,2),
48 payment_method VARCHAR(50),
49 transaction_status VARCHAR(50),
50 is_fraud BIT
51 );

```

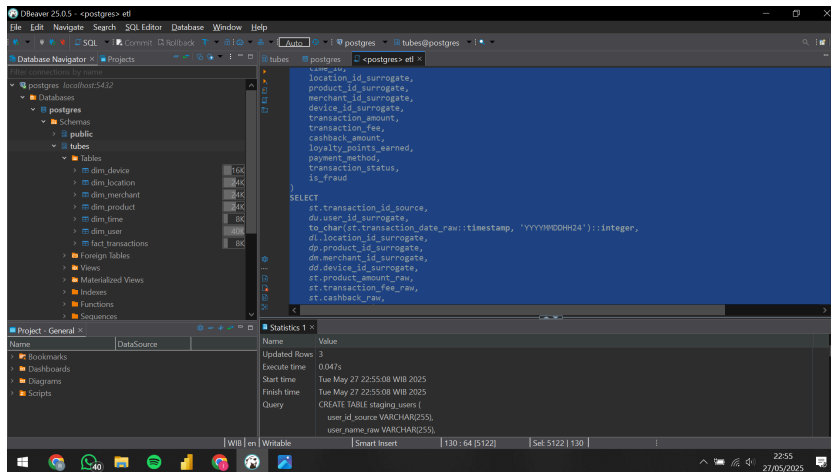
Skema:



Dengan demikian, hasil keseluruhan dari Bagian 6.1.1 adalah terciptanya sebuah data warehouse dengan struktur fisik yang solid dan terdefinisi dengan baik di SQL Server. Fondasi ini siap untuk tahap selanjutnya, yaitu pengisian data melalui proses ETL (Extract, Transform, Load), dan kemudian dimanfaatkan untuk kueri analitik serta visualisasi data guna mendukung pengambilan keputusan strategis di DompertImut. Skrip

SQL yang digunakan untuk implementasi ini juga siap untuk didokumentasikan dan dikelola versinya, misalnya melalui GitHub.

### 6.1.2 ETL



Skrip-skrip SQL yang diberikan (baik untuk transformasi maupun pemuatan) adalah contoh bagaimana operasi-operasi spesifik dalam pipeline ETL dapat diimplementasikan. Dalam praktiknya, skrip-skrip ini seringkali menjadi bagian dari alur kerja yang lebih besar yang dikelola oleh alat ETL seperti SSIS. SSIS akan menangani orkestrasi, penjadwalan, penanganan kesalahan, dan integrasi dengan berbagai sumber dan tujuan data. SQL tetap menjadi bahasa yang sangat penting untuk manipulasi dan transformasi data secara efisien di dalam RDBMS.

### 6.1.3 OLAP

```
SELECT
    p.product_category AS Kategori_Produk,
    COUNT(ft.transaction_id) AS Jumlah_Transaksi,
    SUM(ft.transaction_amount) AS Total_Nominal_Transaksi
FROM
    fact_transactions ft
JOIN
    dim_product p ON ft.product_id_surrogate = p.product_id_surrogate
WHERE
    ft.transaction_status = 'Successful' -- Hanya memperhitungkan transaksi yang berhasil
GROUP BY
    p.product_category
ORDER BY
    Total_Nominal_Transaksi DESC;
```

Berdasarkan data transaksi terkini DompetImut, terlihat jelas bahwa investasi besar kita pada fitur dan promosi di sektor 'Hiburan' telah membuahkan hasil yang fenomenal. Meskipun dari segi jumlah transaksi kategori ini mungkin tidak menempati posisi puncak, namun data Total Nominal Transaksi menunjukkan potensi pendapatan yang sangat signifikan dan tingkat pengeluaran per transaksi yang tinggi dari pengguna di

kategori ini. Ini mengindikasikan bahwa pengguna layanan hiburan adalah high-value customer yang tidak ragu membelanjakan uangnya.

## **7. Evaluasi (apa yang berhasil, apa yang belum, kendala teknis)**

Ada beberapa hal yang telah dilakukan dengan baik selama pelaksanaan proyek ini. Pendekatan tiga lapisan utama Raw, Silver, dan Gold Zone telah digunakan dalam perancangan arsitektur data warehouse, yang memungkinkan proses ekstraksi, transformasi, dan pemuatan data berjalan secara efisien dan terorganisir. Karena menyederhanakan hubungan antar tabel, mempercepat eksekusi permintaan, dan sangat mendukung analisis multidimensi yang dibutuhkan oleh berbagai stakeholder. Star Schema telah terbukti menjadi struktur utama yang tepat. Selain itu, pipeline ETL yang dikembangkan telah dirancang dengan baik. Hal tersebut dapat dilihat dalam proses ekstraksi dari berbagai sumber, pembersihan dan normalisasi data, dan pemuatan akhir ke dalam data warehouse yang telah dioptimalkan melalui partisi berdasarkan waktu dan indeks.

Penggunaan indeks dan partisi waktu sangat membantu meningkatkan performa query analitik dari sisi penyimpanan dan akses data, menjadi sangat penting karena mengingat volume transaksi yang tinggi dan kebutuhan insights real-time yang menjadi prioritas utama bisnis fintech seperti DompetImut. Sistem ini juga mendukung pelaporan dan visualisasi melalui dashboard interaktif yang ramah bagi pengguna non-teknis. Laporan otomatis yang dibuat oleh Power BI atau SSRS mempercepat pengambilan keputusan di level manajemen dan operasional. Tak kalah penting, aspek keamanan sistem ini sangat diperhatikan, termasuk penggunaan otorisasi berbasis peran, enkripsi data, audit log, dan autentikasi multi-faktor untuk memastikan bahwa data pengguna tetap aman dan rahasia.

Namun, masih ada beberapa aspek yang belum ideal dan dapat ditingkatkan lebih lanjut. Salah satu kekurangan utama adalah implementasi yang kurang detail terhadap model machine learning yang direncanakan, seperti deteksi penipuan dan prediksi churn. Model analitik canggih tersebut masih bersifat konseptual karena penjelasan tentang algoritma yang digunakan, hasil pelatihan model, dan evaluasi kinerjanya belum tersedia dalam laporan. Laporan juga tidak menyertakan hasil uji validasi teknis terhadap kinerja sistem.

Hasil uji tersebut meliputi kecepatan proses ETL, waktu eksekusi query dalam kondisi data besar, dan ketahanan terhadap lonjakan beban data. Selain itu, catatan penting adalah kurangnya analisis berbasis kasus nyata karena laporan tersebut lebih berfokus pada desain teknis daripada aplikasi nyata yang menunjukkan bagaimana sistem ini membantu menyelesaikan masalah khusus yang muncul dalam pengoperasian Dompelmut.

Laporan juga tidak menampilkan hasil analitik secara menyeluruh dalam bentuk dashboard atau grafik. Padahal, visualisasi konkret akan sangat membantu dalam menunjukkan bahwa sistem yang dibangun berfungsi dengan baik. Selain itu, metode untuk mendeteksi anomali transaksi masih berbasis aturan dan belum sepenuhnya memanfaatkan potensi teknologi pengajaran mesin yang dapat menyesuaikan diri dengan pola dinamis. Menggabungkan data dari berbagai sumber, baik internal maupun eksternal, dapat menjadi masalah tambahan yang membutuhkan proses transformasi harus lebih teliti dan konsisten.

#### **8. Rencana pengembangan ke depan (jika ingin dikembangkan lagi)**

Pengembangan lanjutan dapat difokuskan pada beberapa arah utama, seperti teknis, analitik, dan integrasi bisnis untuk meningkatkan kapabilitas dan nilai strategis sistem data warehouse Dompelmut. Implementasi praktis dari model analitik lanjutan adalah salah satu prioritas utama, khususnya untuk prediksi churn dan deteksi penipuan. Untuk mengimplementasikan model yang sebelumnya secara operasional, perlu dilakukan proses pelatihan, validasi, dan penyetelan parameter algoritma machine learning seperti decision tree, random forest, atau gradient boosting. Hal tersebut dapat memungkinkan sistem untuk secara otomatis mengenali pola transaksi tidak wajar, memperkirakan kemungkinan pengguna akan berhenti menggunakan layanan, dan memberikan notifikasi atau saran untuk tindak lanjut secara real-time.

Selain itu, untuk mendukung skalabilitas yang lebih elastis, integrasi sistem data warehouse dengan platform cloud seperti Microsoft Azure, Google Cloud Platform, atau AWS dapat menjadi langkah penting. Pemanfaatan cloud memungkinkan pengolahan data secara terdistribusi, pemrosesan paralel, dan akses yang lebih cepat dan fleksibel terhadap sumber daya komputasi tinggi.

Fokus utama lainnya adalah meningkatkan visualisasi. Dashboard saat ini dapat diperluas untuk mencakup analisis prediktif dan insight berbasis AI. Fitur seperti anomaly heatmaps, predictive trend lines, dan user segmentation real-time memungkinkan manajemen untuk memperoleh informasi yang lebih tajam untuk pengambilan keputusan. Selain itu, memasukkan notifikasi otomatis ke platform internal seperti Slack, email, atau WhatsApp Business dapat mempermudah penyebaran informasi penting kepada tim yang relevan tanpa harus membuka dashboard secara manual.




Pengembangan sistem dapat memungkinkan pengembangan data warehouse untuk mencakup domain bisnis lain selain transaksi pembayaran, seperti integrasi dengan sistem CRM (Customer Relationship Management), log aktivitas pengguna, data sosial media, dan data keuangan internal perusahaan. Dengan langkah ini, jangkauan analisis yang tersedia akan diperluas, dan ini akan memungkinkan strategi bisnis yang lebih komprehensif dan berbasis data lintas departemen.

## 9. Tim Proyek

Nama	NIM	Kelompok 27
RANDA ANDRIANA PUTRA	122450083	Ketua
PUTRI MAULIDA CHAIRANI	121450050	Anggota
ELILYA OCTAVIANI	122450009	Anggota
IRHAMNA MAHDI	122450049	Anggota
RAFLY PRABU DARMAWAN	122450140	Anggota

### LAMPIRAN:

Link Google Drive (script, data) :  tubes dw

Link GitHub : <https://github.com/sains-data/Kelompok27-DW-RC>