

**LAPORAN KELOMPOK TUGAS MISI KEEMPAT**  
**PERANCANGAN DATA WAREHOUSE INDUSTRI KESEHATAN**



**Disusun Oleh:**

**Kelompok 3 :**

Taufiqurrahmansyah E	(120450051)
Marshanda Putri P	(121450020)
Putri Durrotul Shopia	(121450116)
Khoirul Mizan Abdullah	(122450010)
Mutiara Dian Pitaloka	(122450047)
Uliano Wilyam Purba	(122450098)

**PROGRAM STUDI SAINS DATA**

**FAKULTAS SAINS**

**INSTITUT TEKNOLOGI SUMATERA 2025**

# **BAB I**

## **PENDAHULUAN**

### **1.1. Ringkasan proyek**

Proyek ini merupakan perancangan dan pembangunan sistem data warehouse untuk industri kesehatan, khususnya di layanan klinik dan rumah sakit. Tujuan utama dari proyek ini adalah untuk mengintegrasikan berbagai sumber data pelayanan kesehatan seperti data pasien, dokter, perawatan, tindakan medis, obat, dan tagihan ke dalam satu sistem pusat yang terstruktur dan dapat digunakan untuk analisis serta pelaporan manajerial. Proyek ini dilakukan secara bertahap melalui beberapa misi yang dimulai dari analisis kebutuhan dan identifikasi struktur data pada misi pertama kemudian dilanjutkan dengan perancangan skema konseptual dan logikal data warehouse menggunakan pendekatan star schema pada misi kedua serta implementasi struktur data secara fisik dengan menggunakan MySQL di lingkungan XAMPP pada misi ketiga. Sistem ini dibangun untuk mendukung kebutuhan analitik seperti pelacakan biaya perawatan pasien analisis kinerja dokter penggunaan dan stok obat serta pemantauan status tagihan dan pembayaran pasien. Dengan adanya integrasi dan struktur data yang baik hasil dari proyek ini diharapkan dapat membantu pihak manajemen dan unit pelayanan medis dalam melakukan pengambilan keputusan yang lebih cepat akurat dan berbasis data historis yang terstandarisasi.

### **1.2 Latar Belakang**

Industri kesehatan modern menghadapi tantangan besar dalam mengelola volume data yang terus meningkat secara eksponensial. Data kesehatan berasal dari berbagai sumber seperti rekam medis elektronik, sistem informasi rumah sakit, perangkat monitoring pasien, aplikasi kesehatan digital, dan sistem asuransi kesehatan. Kompleksitas data ini mencakup data terstruktur seperti hasil laboratorium dan data tidak terstruktur seperti catatan dokter dan citra medis.

Permasalahan utama yang dihadapi adalah fragmentasi data di berbagai sistem yang tidak terintegrasi, kesulitan dalam mengakses informasi secara real-time, dan keterbatasan dalam melakukan analisis mendalam untuk pengambilan keputusan klinis dan operasional. Data warehouse menjadi solusi strategis untuk mengintegrasikan, menyimpan, dan menganalisis data kesehatan secara terpusat, memungkinkan organisasi kesehatan untuk meningkatkan kualitas pelayanan, efisiensi operasional, dan pengambilan keputusan berbasis bukti.

Implementasi data warehouse dalam industri kesehatan tidak hanya mendukung pelaporan rutin, tetapi juga memungkinkan analisis prediktif, deteksi dini penyakit, optimasi alur kerja klinis, dan penelitian kesehatan populasi. Dengan memanfaatkan teknologi business intelligence dan analytics, data warehouse kesehatan dapat memberikan insight yang valuable untuk meningkatkan outcome pasien dan mengurangi biaya operasional.

### 1.3 Rumusan Masalah

Berdasarkan analisis dan perancangan yang dilakukan pada misi sebelumnya, serta kebutuhan akan integrasi dan analitik data dalam industri kesehatan, maka rumusan masalah dalam proyek ini adalah sebagai berikut:

1. Bagaimana cara mengintegrasikan data dari berbagai sistem informasi kesehatan agar tersimpan dalam satu tempat yang terpusat?
2. Bagaimana membangun proses ETL yang dapat menyalurkan data secara akurat dan efisien?
3. Bagaimana menyusun struktur data warehouse yang mampu mendukung kebutuhan analisis dan pelaporan data kesehatan?

### 1.4 Tujuan Proyek

Tujuan dari proyek ini adalah untuk merancang dan mengimplementasikan sistem data warehouse di bidang kesehatan. Secara khusus, tujuan proyek ini meliputi:

1. Merancang struktur data warehouse yang sesuai untuk data kesehatan.
2. Mengintegrasikan data dari berbagai sumber ke dalam satu sistem terpusat.
3. Mengembangkan proses ETL (Extract, Transform, Load) untuk memindahkan data ke data warehouse.
4. Menyediakan tampilan data yang dapat digunakan untuk analisis dan pelaporan.

### 1.5 Ruang Lingkup Sistem

Ruang lingkup proyek ini mencakup proses perancangan dan implementasi sistem *data warehouse* untuk klinik atau fasilitas kesehatan yang memanfaatkan dataset internal dari layanan pasien. Sistem dirancang menggunakan pendekatan *Star Schema* dengan komponen utama sebagai berikut:

#### 1. Sumber Data

Dataset yang digunakan terdiri dari 6 file CSV yang mewakili entitas utama dalam sistem pelayanan klinik:

- dokter.csv : data identitas dan spesialisasi dokter
- pasien.csv : data demografi pasien
- perawatan.csv : data layanan perawatan pasien
- tindakan.csv : rincian tindakan medis yang diberikan
- obat.csv : informasi obat yang diberikan kepada pasien
- tagihan.csv : data pembayaran dan status tagihan

#### 2. Model Data dan Relasi

- Tabel Perawatan merupakan pusat dari skema bintang (fakta utama) yang terhubung dengan tabel dimensi pasien, dokter, tindakan, obat, dan tagihan

- Relasi satu-ke-banyak digunakan, misalnya satu pasien dapat memiliki banyak perawatan, satu perawatan dapat memiliki banyak tindakan, dan satu pasien dapat menerima banyak obat.
3. Perancangan Data Warehouse
    - Sistem dirancang dalam bentuk *Star Schema*, yang terdiri dari satu tabel fakta (Fact\_Perawatan) dan beberapa tabel dimensi (Dim\_Pasien, Dim\_Dokter, Dim\_Tindakan, Dim\_Obat, dan Dim\_Waktu).
    - Tabel fakta menyimpan informasi transaksional seperti tanggal perawatan, biaya, dan status pembayaran.
    - Tabel dimensi menyimpan atribut deskriptif yang dapat digunakan untuk analisis dan pelaporan.
  4. Proses ETL
    - Proses *Extract, Transform, Load* dilakukan untuk memindahkan data dari file CSV ke dalam sistem data warehouse dengan format yang telah ditentukan.
    - Transformasi data meliputi normalisasi, pembersihan data, dan penyesuaian tipe data agar sesuai dengan skema warehouse.

## 1.6 Manfaat Proyek

Proyek perancangan data warehouse ini memberikan manfaat dalam membantu pengelolaan dan analisis data layanan kesehatan secara lebih terstruktur. Bagi tenaga medis dan staf klinik, sistem ini mempermudah akses informasi pasien, obat, dan tindakan. Bagi manajemen, sistem mendukung pelaporan dan pengambilan keputusan berbasis data seperti efisiensi perawatan dan evaluasi biaya. Selain itu, proyek ini juga bermanfaat bagi tim analis dan pengembang sistem sebagai dasar untuk membangun sistem pendukung keputusan yang lebih lanjut di bidang kesehatan.

## **BAB II**

### **METODOLOGI**

#### **2.1 Pendekatan Pengembangan Sistem**

Proyek perancangan data warehouse ini menggunakan pendekatan model Waterfall yaitu metode pengembangan sistem secara sekuensial dan sistematis yang terdiri dari beberapa tahapan berurutan. Pendekatan ini dipilih karena alur kerja proyek sudah dirancang secara terstruktur dan memiliki kebutuhan yang cukup jelas sejak awal sebagaimana tercermin dalam tahapan misi 1 hingga misi 4.

Tahapan model Waterfall yang diterapkan dalam proyek ini adalah sebagai berikut:

- **Analisis Kebutuhan Sistem**  
Tahap awal ini dilakukan pada misi 1 yaitu dengan mengidentifikasi kebutuhan data warehouse dari enam entitas utama (pasien, dokter, tindakan, obat, tagihan, dan perawatan) berdasarkan kondisi aktual klinik atau rumah sakit.
- **Perancangan Sistem**  
Dilakukan pada misi 2 dan 3 termasuk perancangan konseptual (ERD dan skema multidimensi), desain logikal (struktur tabel dan relasi antar entitas), dan desain fisikal (partisi tabel, indeks, dan strategi penyimpanan).
- **Implementasi dan Konstruksi**  
Pada Misi 4 akan dilakukan implementasi data warehouse menggunakan sistem manajemen basis data (XAMPP/MySQL), pembuatan tabel, serta pembangunan proses ETL untuk memindahkan data dari sumber ke sistem warehouse.
- **Pengujian dan Evaluasi**  
Pengujian dilakukan terhadap struktur data, integritas relasi, serta hasil transformasi data. Evaluasi mencakup keberhasilan query, performa sistem, dan fungsionalitas laporan serta visualisasi data.
- **Pemeliharaan dan Pengembangan Lanjutan**  
Meskipun tidak menjadi fokus utama dalam laporan ini pemeliharaan disiapkan melalui penjadwalan ETL berkala, pengelolaan partisi data dan pengawasan terhadap performa sistem melalui proses monitoring otomatis.

#### **2.2 Tools dan Teknologi yang Digunakan**

Berikut adalah tools dan teknologi yang digunakan dalam proyek ini:

<b>Nama Tools</b>	<b>Fungsi Penggunaan</b>
XAMPP	Menyediakan server lokal dan database MySQL untuk menyimpan data warehouse

phpMyAdmin	Antarmuka berbasis web untuk membuat tabel, menjalankan query, dan mengelola database
Microsoft Excel	Membersihkan dan menyesuaikan data awal dari file CSV sebelum di import ke database

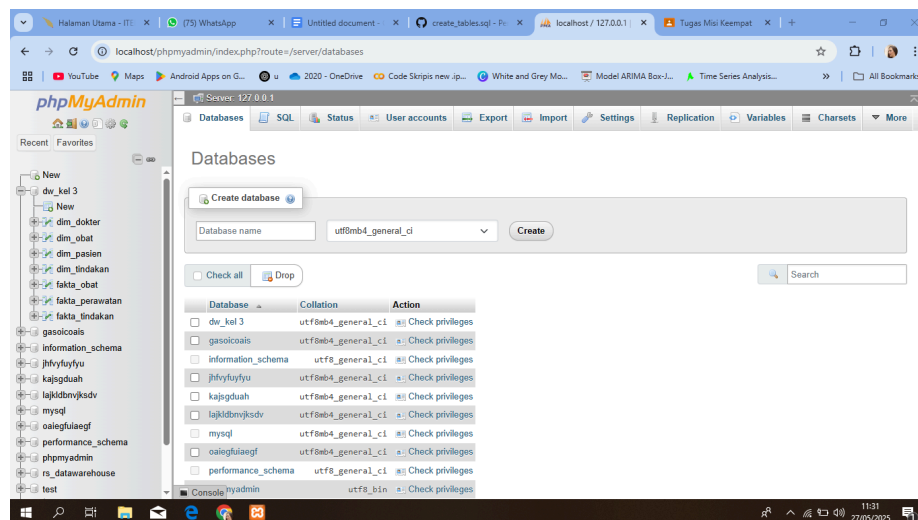
## 2.3 Tahapan Misi

### 2.3.1 Implementasi Gudang Data di SQL Server

Pada tahap ini dilakukan serangkaian aktivitas penting dalam membangun pondasi data warehouse. Dimulai dari pembuatan database yang menjadi wadah penyimpanan utama, kemudian dilanjutkan dengan pendefinisian dan pembuatan tabel-tabel dimensi dan fakta berdasarkan skema konseptual yang telah dirancang sebelumnya. Tabel-tabel ini dirancang agar mampu merepresentasikan entitas utama dalam sistem layanan kesehatan secara akurat dan efisien. Selanjutnya, dirancang relasi antar tabel dengan menggunakan pendekatan skema bintang (star schema), di mana tabel fakta menjadi pusat dari analisis dan terhubung langsung dengan tabel-tabel dimensi. Untuk meningkatkan kinerja sistem dalam mengeksekusi query analitik, dilakukan pula implementasi indexing pada kolom-kolom yang sering digunakan dalam proses pencarian dan penggabungan data, seperti foreign key dan atribut filter utama. Tahapan ini menjadi fondasi teknis yang krusial dalam memastikan sistem data warehouse dapat berjalan secara optimal.

- Pembuatan database

Tahap ini dimulai dengan penentuan nama dan skema database yang akan digunakan. Struktur database dirancang agar mampu menampung data dari berbagai sumber sistem informasi klinik serta fleksibel terhadap pertumbuhan data ke depannya.



- Table dimensi dan fakta

Setelah dibuat database yang akan digunakan selanjutnya kita akan pembuatan tabel dimensi dan fakta.

- a. Pembuatan tabel dimensi

Tabel dimensi berfungsi sebagai referensi kontekstual terhadap data yang disimpan dalam tabel fakta. Dimensi yang dikembangkan antara lain Dim\_Pasien, Dim\_Dokter, Dim\_Tindakan, Dim\_Obat, Dim\_Waktu, dan Dim\_Jenis\_Perawatan. Setiap dimensi dirancang dengan atribut yang memungkinkan analisis mendalam serta mendukung perubahan historis

```
-- Dimensi Pasien
CREATE TABLE dim_pasien (
    id_pasien INT PRIMARY KEY,
    nama_pasien VARCHAR(100),
    alamat TEXT,
    tanggal_lahir DATE,
    jenis_kelamin VARCHAR(10),
    nomor_telepon VARCHAR(20)
);

-- Dimensi Dokter
CREATE TABLE dim_dokter (
    id_dokter INT PRIMARY KEY,
    nama_dokter VARCHAR(100),
    spesialisasi VARCHAR(100),
    nomor_telepon VARCHAR(20)
);

-- Dimensi Obat
CREATE TABLE dim_obat (
    kode_obat VARCHAR(50) PRIMARY KEY,
    nama_obat VARCHAR(100),
    harga_per_satuan DECIMAL(10,2)
);

-- Dimensi Tindakan
CREATE TABLE dim_tindakan (
    nama_tindakan VARCHAR(100) PRIMARY KEY,
    biaya DECIMAL(10,2)
);
```

- b. Pembuatan tabel fakta

Tabel fakta seperti Fact\_Perawatan, Fact\_Tindakan, dan Fact\_Obat menyimpan data transaksi klinik secara terperinci, termasuk biaya perawatan, jumlah tindakan, dan pemakaian obat. Hubungan foreign key memastikan integritas data antara fakta dan dimensi.

```
-- Fakta Perawatan
CREATE TABLE fakta_perawatan (
    id_perawatan INT PRIMARY KEY,
    id_pasien INT,
```

```

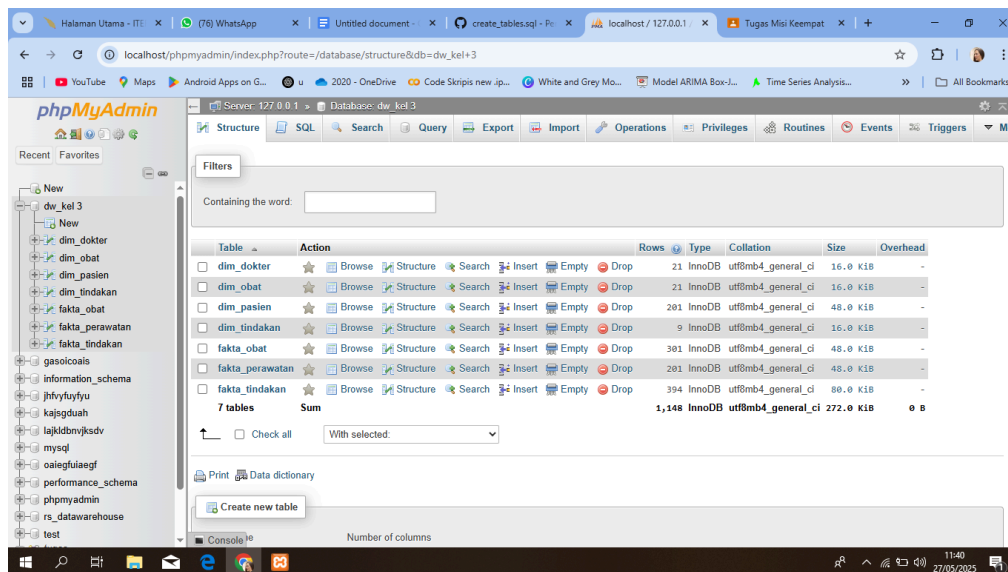
        id_dokter INT,
        tanggal_perawatan DATE,
        jenis_perawatan VARCHAR(100),
        total_biaya DECIMAL(12,2),
        status_pembayaran VARCHAR(20),
        FOREIGN KEY (id_pasien) REFERENCES dim_pasien(id_pasien),
        FOREIGN KEY (id_dokter) REFERENCES dim_dokter(id_dokter)
    );

-- Fakta Obat
CREATE TABLE fakta_obat (
    id_obat_transaksi INT PRIMARY KEY,
    id_pasien INT,
    kode_obat VARCHAR(50),
    jumlah INT,
    total_harga DECIMAL(12,2),
    tanggal_transaksi DATE,
    FOREIGN KEY (id_pasien) REFERENCES dim_pasien(id_pasien),
    FOREIGN KEY (kode_obat) REFERENCES dim_obat(kode_obat)
);

-- Fakta Tindakan
CREATE TABLE fakta_tindakan (
    id_tindakan INT PRIMARY KEY,
    id_perawatan INT,
    nama_tindakan VARCHAR(100),
    biaya DECIMAL(10,2),
    FOREIGN KEY (id_perawatan) REFERENCES
fakta_perawatan(id_perawatan),
    FOREIGN KEY (nama_tindakan) REFERENCES
dim_tindakan(nama_tindakan)
);

```

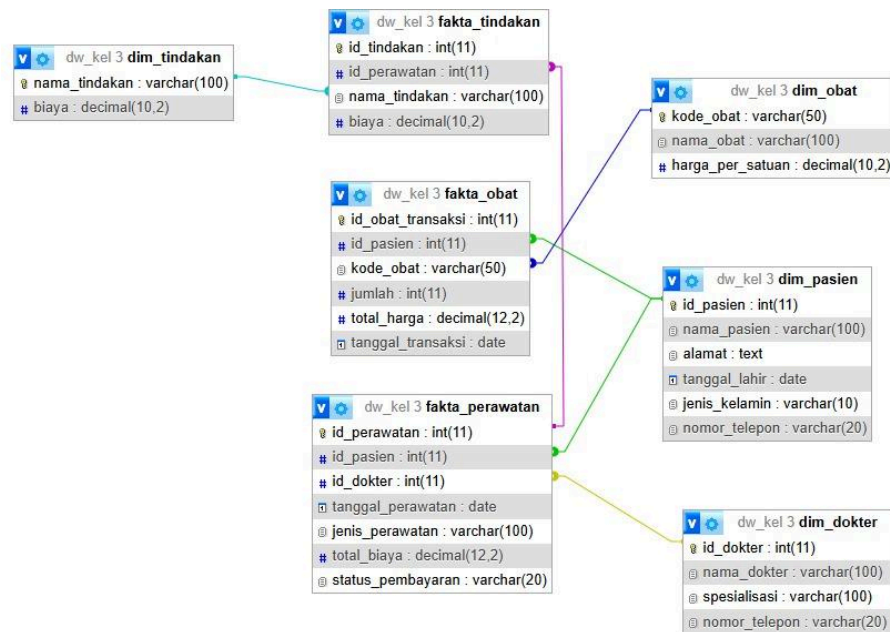
Setelah dibuat table dimensi dan fakta dengan code sql diatas berikut hasilnya



- Relasi antara table



Relasi antar tabel dalam data warehouse ini dirancang menggunakan pendekatan star schema di mana tabel fakta menjadi pusat dan terhubung langsung ke tabel-tabel dimensi. Tabel fakta perawatan terhubung dengan dimensi pasien dan dokter. Tabel fakta tindakan memiliki relasi dengan dimensi tindakan dan juga merujuk ke fakta perawatan untuk konteks layanan. Sementara itu, tabel fakta obat terhubung dengan dimensi obat dan pasien. Struktur relasi ini memungkinkan integrasi data klinis secara efisien untuk keperluan analitik multidimensi seperti evaluasi tindakan medis, biaya pengobatan, dan penggunaan obat.



- Implementasi indexing

Untuk meningkatkan kecepatan akses dan efisiensi dalam eksekusi query, dilakukan penerapan indeks pada kolom-kolom yang sering digunakan dalam operasi pencarian dan penggabungan data. Implementasi indexing ini dilakukan pada masing-masing tabel fakta. Dengan penerapan indeks ini sistem mampu memproses permintaan data dengan lebih cepat dan responsif khususnya untuk laporan dan analisis operasional klinik.

```

-- Indexing pada fakta_perawatan
CREATE INDEX idx_perawatan_pasien ON fakta_perawatan(id_pasien);
CREATE INDEX idx_perawatan_dokter ON fakta_perawatan(id_dokter);

-- Indexing pada fakta_obat
CREATE INDEX idx_obat_pasien ON fakta_obat(id_pasien);
CREATE INDEX idx_obat_kode ON fakta_obat(kode_obat);

-- Indexing pada fakta_tindakan
CREATE INDEX idx_tindakan_perawatan ON fakta_tindakan(id_perawatan);
CREATE INDEX idx_tindakan_nama ON fakta_tindakan(nama_tindakan);
  
```

### 2.3.2 ETL/ELT

Proses ETL (Extract, Transform, Load) merupakan tahapan penting dalam integrasi data ke dalam data warehouse. Berdasarkan misi yang telah dilaksanakan, proses ini dibangun secara bertahap dan sistematis, dimulai dari penarikan data mentah, transformasi, hingga pemuatan akhir ke dalam skema warehouse yang telah dirancang.

- **Extract**

Data diambil dari berbagai sumber file CSV seperti data pasien, dokter, tindakan, perawatan, dan obat. Proses ekstraksi dilakukan menggunakan skrip Python untuk mengotomatisasi pengambilan data berdasarkan tanggal pembaruan terakhir. Hasil ekstraksi disimpan di staging area yang mencerminkan struktur awal data, memungkinkan pemeriksaan dan validasi sebelum diproses lebih lanjut.

- **Transform**

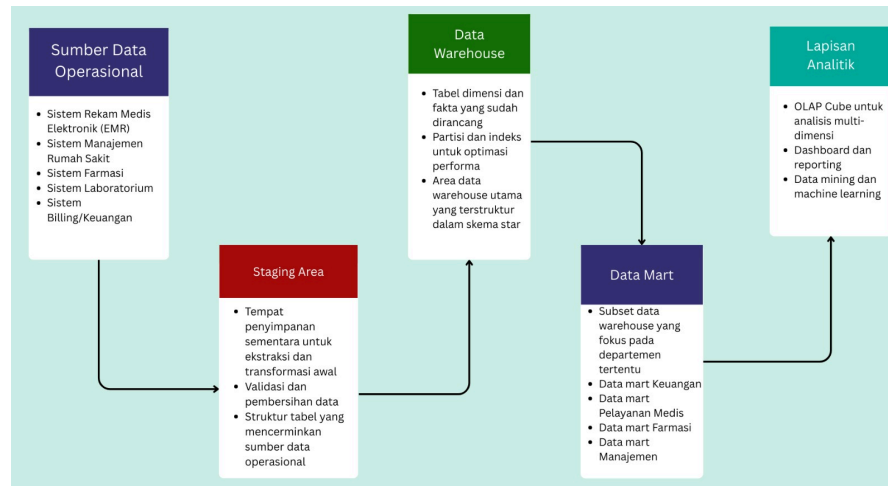
Setelah data yang telah diekstraksi akan dilakukan pembersihan dan transformasi data untuk disesuaikan dengan struktur skema yang telah ditentukan. Proses ini meliputi normalisasi, parsing data alamat, konversi format tanggal, dan pengelompokan data ke dalam hierarki waktu. Transformasi juga melibatkan penerapan Slowly Changing Dimension (SCD) Type 2 untuk dimensi pasien dan dokter, sehingga riwayat perubahan data tetap terjaga.

- **Load**

Data hasil transformasi kemudian dimuat ke dalam tabel fakta dan dimensi menggunakan SQL. Proses ini memastikan hubungan antar tabel terjaga dengan baik dan data dimuat sesuai urutan dependensinya. Loading dilakukan secara bertahap dan terjadwal menggunakan tools seperti Apache Airflow serta dilengkapi mekanisme logging dan monitoring untuk mendeteksi kesalahan. Selain itu, materialized view juga diperbarui secara berkala untuk menunjang laporan dan analisis data yang lebih cepat.

## 2.4 Arsitektur Data Lake House

Arsitektur sistem data warehouse dirancang untuk menangani aliran data dari berbagai sumber operasional menuju sistem analitik yang mendukung pengambilan keputusan strategis. Gambar berikut memperlihatkan alur data dan komponen utama dari sistem data warehouse yang dibangun:



Komponen-komponen utama pada arsitektur sistem ini meliputi:

- **Sumber Data Operasional:** Data berasal dari sistem internal klinik seperti sistem rekam medis elektronik (EMR), sistem manajemen rumah sakit, sistem farmasi, laboratorium, dan billing/keuangan.
- **Staging Area:** Merupakan tempat penyimpanan sementara yang digunakan untuk menampung data mentah hasil ekstraksi. Di sini dilakukan pembersihan dan validasi awal sebelum dimuat ke dalam data warehouse.
- **Data Warehouse:** Tempat penyimpanan utama yang telah didesain dalam bentuk skema star. Di sini terdapat tabel fakta dan tabel dimensi yang telah dipartisi dan diindeks untuk menunjang performa query analitik.
- **Data Mart:** Subset dari data warehouse yang disesuaikan untuk keperluan analitik masing-masing departemen, seperti data mart keuangan, pelayanan medis, farmasi, dan manajemen.
- **Lapisan Analitik:** Tahap akhir di mana data dianalisis secara multidimensi menggunakan OLAP Cube, dashboard visualisasi, atau dimanfaatkan untuk keperluan data mining dan machine learning.

## **BAB III**

### **ANALISIS KEBUTUHAN**

#### **3.1 Kebutuhan Fungsional**

Kebutuhan fungsional mendefinisikan fitur dan fungsi yang harus disediakan oleh sistem data warehouse untuk memenuhi kebutuhan analitik di industri layanan kesehatan.

- Menyimpan dan mengelola data pasien, dokter, tindakan medis, perawatan, tagihan, dan obat.
- Mengintegrasikan data dari berbagai departemen seperti rekam medis, farmasi, keuangan, dan layanan medis.
- Mendukung proses ETL (Extract, Transform, Load) untuk memindahkan data dari file sumber ke dalam data warehouse.
- Menyediakan skema data warehouse berbasis star schema untuk mendukung analisis multidimensi.
- Mendukung query analitik seperti total biaya perawatan, jumlah tindakan per dokter, dan stok obat.

#### **3.2 Kebutuhan Non-Fungsional**

Kebutuhan non-fungsional berkaitan dengan kualitas sistem dan karakteristik pendukung sistem yang dirancang. Kebutuhan tersebut mencakup:

- **Keamanan Data:** Menjaga kerahasiaan data pasien dan memastikan hanya pihak berwenang yang memiliki akses.
- **Konsistensi dan Validitas:** Menjamin integritas data antar entitas dengan aturan validasi dan hubungan yang tepat antar tabel.
- **Kinerja Sistem:** Sistem harus mampu menangani query dalam waktu respons yang wajar, meskipun data terus bertambah.
- **Skalabilitas:** Struktur data warehouse harus mendukung pertumbuhan data jangka panjang.
- **Ketersediaan Sistem:** Sistem harus siap digunakan setiap saat oleh manajemen dan pengguna untuk kebutuhan analisis.

### 3.3 Stakeholder dan Peran

Dalam proyek perancangan data warehouse ini, terdapat beberapa stakeholder utama yang terlibat, masing-masing memiliki peran penting dalam pengembangan dan pemanfaatan sistem:

Stakeholder	Peran dalam sistem
Direktur Rumah Sakit	Menggunakan laporan analisis untuk pengambilan keputusan strategis dan efisiensi operasional.
Manajer Layanan Medis	Menggunakan data untuk mengevaluasi kualitas layanan dan kinerja dokter.
Kepala Unit Kefarmasian	Memantau stok obat, penggunaan obat, dan kebutuhan pengadaan.
Kepala Unit Keuangan	Mengawasi pemasukan, pengeluaran, dan efisiensi keuangan rumah sakit.
Kepala Rekam Medis	Memastikan kelengkapan dan integritas data rekam medis dalam sistem.
Kepala Rawat Inap	Mengelola efisiensi pelayanan rawat inap berdasarkan data operasional.
Kepala Laboratorium	Menjaga kecepatan dan akurasi layanan laboratorium melalui analisis data historis.
Kepala IGD	Meningkatkan respons layanan darurat berdasarkan pola kedatangan dan jenis kasus.

Keterlibatan semua stakeholder ini memastikan bahwa sistem data warehouse yang dikembangkan relevan dengan kebutuhan pengguna, akurat, serta dapat digunakan secara efektif untuk mendukung operasional dan pengambilan keputusan klinik atau rumah sakit.

## BAB IV

### PERANCANGAN SISTEM

#### 4.1 Desain Konseptual

Desain konseptual merupakan representasi awal dari struktur data warehouse yang bertujuan untuk menggambarkan hubungan antar entitas utama dalam industri layanan kesehatan. Berdasarkan analisis pada misi sebelumnya, digunakan pendekatan dimensional modeling dengan skema bintang (star schema) sebagai model utama.

Tabel fakta utama yang dirancang adalah:

- **Fact\_Perawatan:** menyimpan informasi inti dari layanan perawatan pasien seperti biaya total, jumlah tindakan, status pembayaran, dan durasi perawatan.
- **Fact\_Tindakan:** merekam detail tindakan medis yang dilakukan dalam sesi perawatan tertentu.
- **Fact\_Obat:** mencatat transaksi penggunaan obat selama perawatan.

Tabel dimensi yang mendukung analisis data adalah:

- **Dim\_Pasien:** menyimpan atribut pasien termasuk demografi dan informasi kontak.
- **Dim\_Dokter:** berisi data dokter beserta spesialisasi dan unit kerja.
- **Dim\_Tindakan:** mendeskripsikan kategori dan jenis tindakan medis.
- **Dim\_Obat:** memuat data tentang nama, kategori, dan harga satuan obat.
- **Dim\_Jenis\_Perawatan:** membedakan jenis layanan seperti rawat inap dan rawat jalan.
- **Dim\_Waktu:** menyediakan informasi kalender yang dibutuhkan untuk agregasi waktu seperti tahun, bulan, dan kuartal.

Struktur star schema ini dipilih karena kesederhanaannya dalam mendukung analisis OLAP, performa query yang tinggi, dan kemudahan interpretasi oleh pengguna non-teknis. Model ini juga memungkinkan fleksibilitas dalam pengembangan dan integrasi data dari berbagai sumber yang relevan.

#### 4.2 Desain Logikal (DARI MISI 2-3)

Desain logikal merupakan tahap konversi dari model konseptual menjadi definisi struktur tabel, kolom, dan relasi antar tabel yang akan diimplementasikan dalam basis data relasional. Pada tahap ini seluruh entitas dan atribut ditransformasikan menjadi skema logikal yang mencakup primary key, foreign key, dan tipe data.

Setiap tabel dimensi memiliki primary key unik yang menjadi acuan relasi bagi tabel fakta. Granularitas data ditentukan pada level transaksi individual (misalnya sesi perawatan atau pemberian obat). Selain itu, untuk mendukung pelacakan history perubahan data, seperti alamat pasien atau spesialisasi dokter, diterapkan metode Slowly Changing Dimension (SCD) tipe 2

pada dimensi terkait. Kolom tambahan seperti tanggal efektif, tanggal kadaluarsa, dan flag status digunakan untuk menandai rekaman aktif. Desain logikal ini dirancang agar mendukung efisiensi dalam proses ETL serta kompatibel dengan kebutuhan analitik yang kompleks di industri kesehatan.

### 4.3 Desain Fisikal

Desain fisik merupakan tahap penerapan nyata dari struktur logikal ke dalam sistem manajemen basis data. Tujuan utamanya adalah mengoptimalkan penyimpanan, efisiensi akses data, dan skalabilitas sistem data warehouse dalam menghadapi volume data yang besar. Berdasarkan hasil misi sebelumnya berikut adalah aspek-aspek penting dalam desain fisik:

- **Implementasi Indeks**

Untuk meningkatkan performa query, sistem mengimplementasikan indeks pada kolom-kolom yang sering digunakan dalam operasi pencarian dan penggabungan data. Indeks diterapkan terutama pada foreign key di tabel fakta seperti *id\_pasien*, *id\_dokter*, *id\_perawatan*, *kode\_obat*, dan *nama\_tindakan*. Jenis indeks yang digunakan adalah B-Tree dan bitmap, tergantung pada jenis kolom dan kebutuhan analitiknya.

- **Strategi Penyimpanan**

Tabel fakta menggunakan columnar storage untuk mendukung efisiensi dalam pengolahan data agregasi sedangkan tabel dimensi tetap disimpan menggunakan row-based storage agar cepat diakses saat melakukan lookup. Kompresi data juga diterapkan menggunakan metode dictionary untuk string yang berulang dan run-length untuk atribut waktu.

- **Partisi Tabel**

Tabel fakta dipartisi untuk mendukung performa dan manajemen data:

- *Fact\_Perawatan* di partisi berdasarkan waktu (range per tahun).
- *Fact\_Tindakan* di partisi berdasarkan kategori jenis tindakan (list partitioning)

Dengan partisi ini, sistem dapat mengeksekusi query lebih cepat dan memudahkan backup maupun penghapusan data historis.

- **Materialized View**

Untuk mendukung pelaporan rutin dan analitik terjadwal, sistem menyediakan materialized view seperti:

- Pendapatan bulanan berdasarkan spesialisasi dokter.
- Analisis penggunaan obat berdasarkan kategori.
- Ringkasan indikator performa klinik (KPI).

- **Retensi dan Arsip Data**

Sistem menyimpan data transaksi detail selama minimal 5 tahun. Data ringkasan dan agregat disimpan secara permanen. Data yang jarang diakses akan dipindahkan ke media penyimpanan terpisah untuk efisiensi.

## BAB V

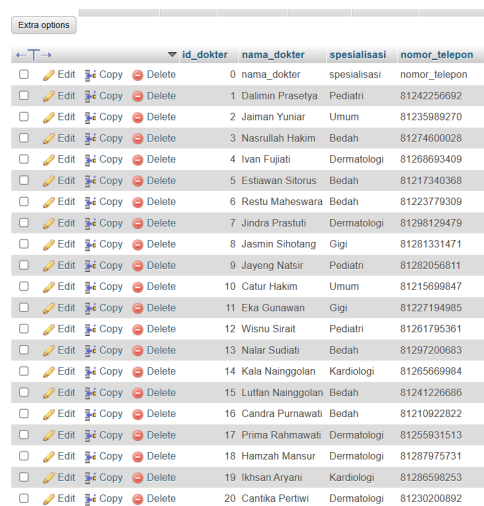
### IMPLEMENTASI SISTEM

#### 5.1 Proses Implementasi

##### 5.1.1 Screenshots Sistem

- **Dim\_dokter**

Menampilkan daftar dokter yang terdiri dari kolom id\_dokter, nama\_dokter, spesialisasi, dan nomor\_telepon. Tabel ini digunakan untuk mendukung analisis berdasarkan spesialisasi dan identitas dokter.



Extra options		id_dokter	nama_dokter	spesialisasi	nomor_telepon
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0	nama_dokter	spesialisasi	nomor_telepon	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Dalimin Prasetya	Pediatri	81242256092	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Jaiman Yuniar	Umum	81235989270	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Nasrullah Hakim	Bedah	81274600028	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Ivan Fujiati	Dermatologi	81268693409	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Estiawan Sitorus	Bedah	81217340368	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	Restu Maheswara	Bedah	81223779309	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	Jindra Prastuti	Dermatologi	81298129479	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	Jasmin Sihotang	Gigi	81261331471	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	Jayeng Natsir	Pediatri	81282056811	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	Catur Hakim	Umum	81215699847	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	Eka Gunawan	Gigi	81227194985	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	Wisnu Sirait	Pediatri	81261795361	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13	Nalar Sudiati	Bedah	81297200683	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	14	Kala Nainggolan	Kardiologi	81265669984	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	Lutfan Nainggolan	Bedah	81241226686	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	16	Candra Purnawati	Bedah	81210922822	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	17	Prima Rahmawati	Dermatologi	81255931513	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	18	Hamzah Mansur	Dermatologi	81267975731	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	19	Ikhshan Aryani	Kardiologi	81206598253	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	20	Cantika Pertwi	Dermatologi	81230200892	

- **Dim\_obat**

Memuat data obat yang mencakup kode\_obat, nama\_obat, dan harga\_per\_satuan. Tabel ini dimanfaatkan untuk menganalisis biaya pengobatan serta tren penggunaan obat berdasarkan kategori dan periode waktu.





Extra options

		nama_tindakan	biaya
<input type="checkbox"/>	Edit Copy Delete	CT Scan	500000.00
<input type="checkbox"/>	Edit Copy Delete	nama_tindakan	0.00
<input type="checkbox"/>	Edit Copy Delete	Operasi Caesar	200000.00
<input type="checkbox"/>	Edit Copy Delete	Operasi Usus Buntu	1500000.00
<input type="checkbox"/>	Edit Copy Delete	Pemeriksaan Darah	6000000.00
<input type="checkbox"/>	Edit Copy Delete	Pemeriksaan Jantung	1500000.00
<input type="checkbox"/>	Edit Copy Delete	Pemeriksaan Umum	500000.00
<input type="checkbox"/>	Edit Copy Delete	USG	500000.00
<input type="checkbox"/>	Edit Copy Delete	X-Ray	200000.00

☐ Check all    With selected:    Edit    Copy    Delete

- Fakta\_obat**

Berisi data transaksi penggunaan obat oleh pasien. Tabel ini mencakup kolom id\_obat\_transaksi, id\_pasien, kode\_obat, jumlah, total\_harga, dan tanggal\_transaksi. Tabel ini digunakan untuk menganalisis pengeluaran obat berdasarkan pasien dan periode waktu, serta menghitung biaya pengobatan yang terkait dengan obat-obatan tertentu.

		id_obat_transaksi	id_pasien	kode_obat	jumlah	total_harga	tanggal_transaksi
<input type="checkbox"/>	Edit Copy Delete	0	0	kode_obat	0	0.00	NULL
<input type="checkbox"/>	Edit Copy Delete	1	159	O002	9	36000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	2	118	O001	9	31500.00	NULL
<input type="checkbox"/>	Edit Copy Delete	3	103	O014	8	24000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	4	183	O015	2	7000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	5	156	O005	2	6000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	6	160	O019	3	9000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	7	153	O017	6	150000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	8	155	O014	8	96000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	9	168	O018	2	6000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	10	116	O008	4	14000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	11	117	O013	6	24000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	12	110	O011	2	8000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	13	40	O016	5	17500.00	NULL
<input type="checkbox"/>	Edit Copy Delete	14	22	O003	2	6000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	15	190	O012	2	8000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	16	151	O018	9	45000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	17	32	O014	6	150000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	18	185	O002	7	84000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	19	91	O004	10	120000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	20	40	O016	9	225000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	21	143	O012	2	10000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	22	58	O014	5	15000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	23	173	O018	10	250000.00	NULL
<input type="checkbox"/>	Edit Copy Delete	24	169	O009	10	30000.00	NULL

- Fakta\_perawatan**

Berisi data perawatan pasien yang mencakup id\_perawatan, id\_pasien, id\_dokter, tanggal\_perawatan, jenis\_perawatan, total\_biaya, dan status\_pembayaran. Tabel ini mencatat seluruh aktivitas perawatan baik rawat inap maupun rawat jalan, serta informasi pembayaran yang dilakukan pasien.

		id_perawatan	id_pasien	id_dokter	tanggal_perawatan	jenis_perawatan	total_biaya	status_pembayaran
<input type="checkbox"/>	Edit	Copy	Delete	0	0	0 NULL	jenis_perawatan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	1	5	19 2024-07-10	Rawat Inap	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	2	54	7 2024-07-28	Rawat Inap	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	3	162	18 2024-01-30	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	4	125	7 2024-07-08	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	5	18	20 2024-08-20	Rawat Inap	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	6	79	14 2024-02-07	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	7	143	19 2024-03-19	Rawat Inap	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	8	161	13 2025-01-19	Rawat Inap	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	9	120	3 2024-02-01	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	10	3	12 2024-02-16	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	11	4	4 2025-02-08	Rawat Inap	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	12	80	5 2024-07-22	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	13	200	7 2024-01-12	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	14	67	17 2024-11-19	Rawat Inap	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	15	131	18 2025-01-11	Rawat Inap	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	16	108	10 2025-02-23	Rawat Inap	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	17	132	11 2024-11-19	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	18	99	6 2024-09-18	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	19	81	20 2024-07-27	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	20	159	15 2024-10-20	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	21	68	10 2024-06-21	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	22	188	10 2024-03-19	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	23	2	15 2024-11-16	Rawat Jalan	0.00 Belum Lunas
<input type="checkbox"/>	Edit	Copy	Delete	24	12	8 2025-01-26	Rawat Jalan	0.00 Belum Lunas

## - Fakta\_tindakan

Menampilkan catatan tindakan medis yang diberikan dalam setiap sesi perawatan. Tabel ini terdiri dari id\_tindakan, id\_perawatan, nama\_tindakan, dan biaya. Data ini digunakan untuk melacak jenis layanan medis yang diberikan serta menghitung biaya berdasarkan tindakan.

		id_tindakan	id_perawatan	nama_tindakan	biaya
<input type="checkbox"/>	Edit	Copy	Delete	1	0 nama_tindakan 0.00
<input type="checkbox"/>	Edit	Copy	Delete	2	1 Operasi Caesar 200000.00
<input type="checkbox"/>	Edit	Copy	Delete	3	1 Pemeriksaan Jantung 1500000.00
<input type="checkbox"/>	Edit	Copy	Delete	4	1 CT Scan 500000.00
<input type="checkbox"/>	Edit	Copy	Delete	5	2 USG 500000.00
<input type="checkbox"/>	Edit	Copy	Delete	6	3 CT Scan 1500000.00
<input type="checkbox"/>	Edit	Copy	Delete	7	4 Operasi Usus Buntu 1500000.00
<input type="checkbox"/>	Edit	Copy	Delete	8	4 CT Scan 200000.00
<input type="checkbox"/>	Edit	Copy	Delete	9	4 Operasi Usus Buntu 6000000.00
<input type="checkbox"/>	Edit	Copy	Delete	10	5 Operasi Caesar 500000.00
<input type="checkbox"/>	Edit	Copy	Delete	11	5 Pemeriksaan Jantung 200000.00
<input type="checkbox"/>	Edit	Copy	Delete	12	5 CT Scan 1500000.00
<input type="checkbox"/>	Edit	Copy	Delete	13	6 CT Scan 250000.00
<input type="checkbox"/>	Edit	Copy	Delete	14	7 Pemeriksaan Jantung 500000.00
<input type="checkbox"/>	Edit	Copy	Delete	15	8 Pemeriksaan Jantung 200000.00
<input type="checkbox"/>	Edit	Copy	Delete	16	8 X-Ray 200000.00
<input type="checkbox"/>	Edit	Copy	Delete	17	9 Operasi Usus Buntu 3000000.00
<input type="checkbox"/>	Edit	Copy	Delete	18	10 Operasi Caesar 3000000.00
<input type="checkbox"/>	Edit	Copy	Delete	19	10 Pemeriksaan Darah 6000000.00
<input type="checkbox"/>	Edit	Copy	Delete	20	10 USG 200000.00
<input type="checkbox"/>	Edit	Copy	Delete	21	11 Pemeriksaan Umum 500000.00
<input type="checkbox"/>	Edit	Copy	Delete	22	12 CT Scan 6000000.00
<input type="checkbox"/>	Edit	Copy	Delete	23	12 Operasi Usus Buntu 6000000.00
<input type="checkbox"/>	Edit	Copy	Delete	24	12 CT Scan 6000000.00
<input type="checkbox"/>	Edit	Copy	Delete	25	13 Operasi Usus Buntu 200000.00

### 5.1.2 Penjelasan Query yang Digunakan

Sistem data warehouse ini dirancang untuk mendukung analisis multidimensi berbasis OLAP (Online Analytical Processing), sehingga query yang digunakan bersifat agregatif dan eksploratif. SQL digunakan untuk mengeksekusi berbagai kebutuhan analitik yang memungkinkan pengguna melakukan slice, dice, roll-up, dan drill-down berdasarkan dimensi waktu, dokter, tindakan, dan obat. Berikut jenis-jenis query yang digunakan:

```
-- Total Penjualan Obat per Bulan
SELECT
    DATE_FORMAT(tanggal_transaksi, '%Y-%m') AS bulan,
    SUM(total_harga) AS total_penjualan
FROM fakta_obat
GROUP BY DATE_FORMAT(tanggal_transaksi, '%Y-%m')
ORDER BY bulan;

-- Rata-rata Transaksi Obat per Pasien
SELECT
    f.id_pasien,
    p.nama_pasien,
    AVG(f.total_harga) AS rata_rata_transaksi
FROM fakta_obat f
JOIN dim_pasien p ON f.id_pasien = p.id_pasien
GROUP BY f.id_pasien, p.nama_pasien
ORDER BY rata_rata_transaksi DESC;

-- Obat Terlaris Berdasarkan Total Penjualan
SELECT
    o.nama_obat,
    SUM(f.jumlah) AS total_terjual
FROM fakta_obat f
JOIN dim_obat o ON f.kode_obat = o.kode_obat
GROUP BY o.nama_obat
ORDER BY total_terjual DESC
LIMIT 10;

-- Tren Pertumbuhan Penjualan Obat dari Tahun ke Tahun
SELECT
    YEAR(tanggal_transaksi) AS tahun,
    SUM(total_harga) AS total_penjualan
```

```

FROM fakta_obat
GROUP BY YEAR(tanggal_transaksi)
ORDER BY tahun;

-- Jumlah Tindakan per Dokter
SELECT
    d.nama_dokter,
    COUNT(t.id_tindakan) AS total_tindakan
FROM fakta_tindakan t
JOIN fakta_perawatan p ON t.id_perawatan = p.id_perawatan
JOIN dim_dokter d ON p.id_dokter = d.id_dokter
GROUP BY d.nama_dokter
ORDER BY total_tindakan DESC;

-- Total Biaya Tindakan per Jenis Tindakan
SELECT
    nama_tindakan,
    SUM(biaya) AS total_biaya
FROM fakta_tindakan
GROUP BY nama_tindakan
ORDER BY total_biaya DESC;

-- Jumlah Pasien Berdasarkan Jenis Kelamin
SELECT
    jenis_kelamin,
    COUNT(*) AS jumlah_pasien
FROM dim_pasien
GROUP BY jenis_kelamin;

-- Ringkasan perawatan pasien
CREATE VIEW vw_ringkasan_perawatan_pasien AS
SELECT
    f.id_perawatan,
    p.nama_pasien,
    d.nama_dokter,
    f.tanggal_perawatan,
    f.jenis_perawatan,
    f.total_biaya,
    f.status_pembayaran

```

```

FROM fakta_perawatan f
JOIN dim_pasien p ON f.id_pasien = p.id_pasien
JOIN dim_dokter d ON f.id_dokter = d.id_dokter;

-- Status pembayaran pasien
CREATE VIEW vw_status_pembayaran AS
SELECT
    p.nama_pasien,
    f.id_perawatan,
    f.total_biaya,
    f.status_pembayaran
FROM fakta_perawatan f
JOIN dim_pasien p ON f.id_pasien = p.id_pasien;

-- Statistik Tindakan per dokter
CREATE VIEW vw_tindakan_per_dokter AS
SELECT
    d.nama_dokter,
    t.nama_tindakan,
    COUNT(t.id_tindakan) AS jumlah_tindakan,
    SUM(t.biaya) AS total_biaya_tindakan
FROM fakta_tindakan t
JOIN fakta_perawatan f ON t.id_perawatan = f.id_perawatan
JOIN dim_dokter d ON f.id_dokter = d.id_dokter
GROUP BY d.nama_dokter, t.nama_tindakan;

-- Total biaya obat per pasien
CREATE VIEW vw_total_obat_pasien AS
SELECT
    p.nama_pasien,
    o.kode_obat,
    do.nama_obat,
    SUM(o.jumlah) AS total_jumlah,
    SUM(o.total_harga) AS total_pengeluaran
FROM fakta_obat o
JOIN dim_pasien p ON o.id_pasien = p.id_pasien
JOIN dim_obat do ON o.kode_obat = do.kode_obat
GROUP BY p.nama_pasien, o.kode_obat, do.nama_obat;

```

```
-- Total biaya yang harus dibayarkan pasien
CREATE VIEW vw_total_biaya_pasien AS
SELECT
    p.nama_pasien,
    COALESCE(SUM(fp.total_biaya), 0) AS biaya_perawatan,
    COALESCE(SUM(fo.total_harga), 0) AS biaya_obat,
    COALESCE(SUM(fp.total_biaya), 0) + COALESCE(SUM(fo.total_harga),
0) AS total_biaya
FROM dim_pasien p
LEFT JOIN fakta_perawatan fp ON p.id_pasien = fp.id_pasien
LEFT JOIN fakta_obat fo ON p.id_pasien = fo.id_pasien
GROUP BY p.nama_pasien;
```

Berikut jenis-jenis query yang digunakan:

- **Query Agregasi**  
Digunakan untuk menghitung metrik seperti total biaya perawatan, jumlah tindakan, jumlah penggunaan obat, dan pasien per periode waktu. Query ini mendukung perhitungan statistik untuk laporan keuangan dan operasional klinik.
- **Query Analisis Waktu**  
Menggunakan atribut waktu seperti tahun dan bulan untuk menganalisis tren dari waktu ke waktu. Laporan bulanan, kuartalan, dan tahunan dapat dihasilkan dari struktur ini.
- **Query Join Antar Tabel**  
Menggabungkan tabel fakta dan dimensi untuk menghasilkan laporan yang lengkap, seperti pendapatan per spesialisasi dokter atau tren penggunaan obat berdasarkan kategori. Join dilakukan pada foreign key yang menghubungkan fakta dengan dimensi.
- **Materialized View dan View**  
Untuk mempercepat laporan dan memudahkan eksplorasi data, query analitik disimpan dalam bentuk view dan materialized view. Contohnya adalah laporan KPI klinik, tren penggunaan obat, dan pendapatan bulanan berdasarkan spesialisasi dokter.

### 5.1.3 Penjelasan Proses ETL

Proses ETL (Extract, Transform, Load) merupakan komponen kunci dalam integrasi data warehouse. Proses ini mencakup pembuatan staging table, transformasi data, dan pemuatan ke tabel akhir.

- **Extract:** Data diekstraksi dari file CSV yang mencerminkan data operasional klinik, seperti informasi pasien, dokter, tindakan medis, obat, tagihan, dan perawatan. File ini diimpor ke dalam tabel staging sebagai bentuk awal integrasi data sebelum transformasi lebih lanjut.

- **Transform:** Setelah berada di staging, data menjalani proses transformasi yang mencakup konversi tipe data, penyesuaian format (terutama pada tanggal dan nilai numerik), pembersihan data duplikat, serta pemetaan nilai ke format standar sistem. Transformasi ini juga mencakup penerapan logika seperti validasi keberadaan data dan penanganan atribut historis untuk dimensi yang bersifat berubah, seperti pasien dan dokter.
- **Load:** Data yang telah dibersihkan dan ditransformasi dimuat ke dalam struktur final data warehouse yang terdiri dari tabel fakta dan dimensi. Proses ini dilakukan dengan memastikan tidak terjadi duplikasi dan menjaga integritas referensial antar tabel.

Seluruh proses ETL diimplementasikan menggunakan skrip SQL yang disusun secara modular. Struktur skrip ini memungkinkan otomatisasi dan pembaruan berkala menggunakan penjadwalan sistem seperti cron, sehingga data warehouse selalu diperbarui secara konsisten dan tepat waktu. dan dapat dijalankan langsung. Untuk penerapan jangka panjang, proses ini dapat dijadwalkan otomatis menggunakan task scheduler seperti cron agar pembaruan data berlangsung rutin dan konsisten. dan dapat dikombinasikan dengan task scheduler seperti cron untuk memperbarui isi data warehouse secara rutin.

### 5.1.3 Skrip ETL

```
-- Contoh staging table dari data sumber
CREATE TABLE staging_pasien (
    id_pasien INT,
    nama_pasien VARCHAR(100),
    alamat TEXT,
    tanggal_lahir VARCHAR(20),
    jenis_kelamin VARCHAR(10),
    nomor_telepon VARCHAR(20)
);

CREATE TABLE staging_dokter (
    id_dokter INT,
    nama_dokter VARCHAR(100),
    spesialisasi VARCHAR(100),
    nomor_telepon VARCHAR(20)
);

CREATE TABLE staging_obat (
    kode_obat VARCHAR(50),
    nama_obat VARCHAR(100),
    harga_per_satuan VARCHAR(20)
);

CREATE TABLE staging_tindakan (
    nama_tindakan VARCHAR(100),
    biaya VARCHAR(20)
);
```



```

CREATE TABLE staging_perawatan (
    id_perawatan INT,
    id_pasien INT,
    id_dokter INT,
    tanggal_perawatan VARCHAR(20),
    jenis_perawatan VARCHAR(100),
    total_biaya VARCHAR(20),
    status_pembayaran VARCHAR(20)
);

CREATE TABLE staging_obat_transaksi (
    id_obat_transaksi INT,
    id_pasien INT,
    kode_obat VARCHAR(50),
    jumlah INT,
    total_harga VARCHAR(20),
    tanggal_transaksi VARCHAR(20)
);

CREATE TABLE staging_tindakan (
    id_tindakan INT,
    id_perawatan INT,
    nama_tindakan VARCHAR(100),
    biaya VARCHAR(20)
);

INSERT INTO dim_pasien (id_pasien, nama_pasien, alamat,
tanggal_lahir, jenis_kelamin, nomor_telepon)
SELECT
    id_pasien,
    nama_pasien,
    alamat,
    TRY_CAST(tanggal_lahir AS DATE),
    LOWER(jenis_kelamin),
    nomor_telepon
FROM staging_pasien
WHERE id_pasien NOT IN (SELECT id_pasien FROM dim_pasien);

INSERT INTO dim_dokter (id_dokter, nama_dokter, spesialisasi,
nomor_telepon)
SELECT
    id_dokter,
    nama_dokter,
    spesialisasi,
    nomor_telepon
FROM staging_dokter
WHERE id_dokter NOT IN (SELECT id_dokter FROM dim_dokter);

INSERT INTO dim_obat (kode_obat, nama_obat, harga_per_satuan)
SELECT

```

```

        kode_obat,
        nama_obat,
        CAST(harga_per_satuan AS DECIMAL(10,2))
FROM staging_obat
WHERE kode_obat NOT IN (SELECT kode_obat FROM dim_obat);

INSERT INTO dim_tindakan (nama_tindakan, biaya)
SELECT
    nama_tindakan,
    CAST(biaya AS DECIMAL(10,2))
FROM staging_tindakan
WHERE nama_tindakan NOT IN (SELECT nama_tindakan FROM
dim_tindakan);

INSERT INTO fakta_perawatan (id_perawatan, id_pasien, id_dokter,
tanggal_perawatan, jenis_perawatan, total_biaya,
status_pembayaran)
SELECT
    id_perawatan,
    id_pasien,
    id_dokter,
    TRY_CAST(tanggal_perawatan AS DATE),
    jenis_perawatan,
    CAST(total_biaya AS DECIMAL(12,2)),
    status_pembayaran
FROM staging_perawatan
WHERE id_perawatan NOT IN (SELECT id_perawatan FROM
fakta_perawatan);

INSERT INTO fakta_obat (id_obat_transaksi, id_pasien, kode_obat,
jumlah, total_harga, tanggal_transaksi)
SELECT
    id_obat_transaksi,
    id_pasien,
    kode_obat,
    Jumlah,
    CAST(total_harga AS DECIMAL(12,2)),
    TRY_CAST(tanggal_transaksi AS DATE)
FROM staging_obat_transaksi
WHERE id_obat_transaksi NOT IN (SELECT id_obat_transaksi FROM
fakta_obat);

INSERT INTO fakta_tindakan (id_tindakan, id_perawatan,
nama_tindakan, biaya)
SELECT
    id_tindakan,
    id_perawatan,
    nama_tindakan,
    CAST(biaya AS DECIMAL(10,2))
FROM staging_tindakan
WHERE id_tindakan NOT IN (SELECT id_tindakan FROM

```

```
fakta_tindakan);
```

Dalam penjelasan step by step dari query etl diatas, kami menjelaskan dengan contoh salah satu dari code. Contoh dim\_pasien, pertama kita membuat staging untuk tempat import data. Setelah di buat staging kita import data ke staging. Lalu setelah di import gunakan code bagian insert to untuk memindahkan dari staging ke dimensinya.

## 5.2 Hasil Implementasi

### 5.2.1 Tampilan Sistem

- **Total Penjualan Obat per Bulan**

Berdasarkan data total penjualan obat per bulan dari Januari 2023 hingga Oktober 2024, terlihat adanya fluktuasi yang signifikan. Puncak penjualan terjadi pada Oktober 2024 dengan total penjualan sebesar 1.418.500, yang merupakan angka tertinggi selama periode pengamatan. Penjualan juga tinggi pada Maret 2024 dan Oktober 2023, menunjukkan adanya kemungkinan tren peningkatan konsumsi obat pada awal dan akhir tahun. Sementara itu, bulan-bulan seperti Desember 2023 dan September 2023 menunjukkan penjualan yang cukup rendah, masing-masing di bawah 500.000. Hal ini bisa menjadi dasar evaluasi stok dan strategi pengadaan obat oleh pihak manajemen farmasi.

bulan ▲ 1	total_penjualan
2023-01	599000.00
2023-02	656500.00
2023-03	700500.00
2023-04	1084000.00
2023-05	583000.00
2023-06	1017000.00
2023-07	312000.00
2023-08	488500.00
2023-09	447500.00
2023-10	1094500.00
2023-11	616500.00
2023-12	219000.00
2024-01	589000.00
2024-02	334000.00
2024-03	1014000.00
2024-04	376000.00
2024-05	715000.00
2024-06	477000.00
2024-07	697000.00
2024-08	265000.00
2024-09	250000.00
2024-10	1418500.00

- **Rata-rata Transaksi Obat per Pasien**

Pasien seperti Vivi Prasasta dan Sabar Pradana memiliki rata-rata transaksi tertinggi sekitar Rp250.000, menunjukkan kemungkinan kebutuhan obat rutin atau berbiaya tinggi. Sebaliknya, pasien seperti Tutik Rahayu memiliki rata-rata transaksi lebih rendah. Informasi ini berguna untuk memetakan kebutuhan obat berdasarkan profil pasien.

id_pasien	nama_pasien	rata_rata_transaksi ▾ 1
44	Vivi Prasasta	250000.000000
109	Sabar Pradana	250000.000000
100	Raden Susanti	225000.000000
99	Hilda Gunawan	200000.000000
104	Okto Hariyah	175000.000000
77	Nadia Suwarno	175000.000000
35	Gatra Jailani	150000.000000
153	Edi Marpaung	150000.000000
123	Taufik Rahayu	126500.000000
186	Lanang Rajasa	125000.000000
91	Luis Prasetyo	120000.000000
94	Hamima Natsir	108000.000000
147	Halima Yulianti	106000.000000
164	Indra Laksmiwati	103250.000000
168	Martani Saptono	103000.000000
37	Irwan Putra	102500.000000
93	Raditya Putra	98875.000000
173	Wardaya Prabowo	97333.333333
64	Hartaka Hutasoit	96333.333333
50	Slamet Mardhiyah	96000.000000
10	Kenes Kurniawan	96000.000000

- **Obat Terlaris Berdasarkan Total Penjualan**

Dari data penjualan, Paracetamol menjadi obat dengan total penjualan tertinggi yaitu sebanyak 679 kali, diikuti oleh Ibuprofen sebanyak 562 kali dan Antasida sebanyak 403 kali. Obat-obatan ini termasuk dalam kategori yang umum digunakan, menunjukkan tingginya kebutuhan akan obat untuk keluhan ringan seperti demam dan nyeri.

nama_obat	total_terjual ▾ 1
Paracetamol	679
Ibuprofen	562
Antasida	403
Vitamin C	58
Salbutamol	37
nama_obat	0

- **Tren Pertumbuhan Penjualan Obat dari Tahun ke Tahun**

Penjualan obat secara tahunan relatif stabil, dengan total penjualan tahun 2023 sebesar Rp7.818.000 dan tahun 2024 sebesar Rp7.737.000. Penurunan tipis ini menunjukkan bahwa secara umum permintaan obat tetap tinggi, namun tidak mengalami peningkatan signifikan dari tahun sebelumnya.

tahun ▲ 1	total_penjualan
2023	7818000.00
2024	7737000.00

- **Jumlah Tindakan per Dokter**

Dokter dengan tindakan terbanyak adalah Jindra Prastuti (33 tindakan), diikuti oleh Candra Purmawati dan Ivan Fujiati. Mayoritas dokter lainnya menangani antara 10–25 tindakan, menunjukkan distribusi kerja yang cukup seimbang.

nama_dokter	total_tindakan ▼ 1
Jindra Prastuti	33
Candra Purmawati	31
Ivan Fujiati	26
Catur Hakim	25
Restu Maheswara	25
Nasrullah Hakim	25
Cantika Pertiwi	24
Jayeng Natsir	23
Eka Gunawan	23
Dalimin Prasetya	21
Estiawan Sitorus	19
Prima Rahmawati	18
Hamzah Mansur	16
Wisnu Sirait	15
Lutfan Nainggolan	14
Jaiman Yuniar	13
Jasmin Sihotang	12
Nalar Sudiat	10
Kala Nainggolan	10
Ikhsan Aryani	10
nama_dokter	1

- **Total Biaya Tindakan per Jenis Tindakan**

Operasi Usus Buntu menjadi tindakan dengan total biaya tertinggi, disusul oleh USG, Pemeriksaan Darah, dan CT Scan. Ini menunjukkan bahwa tindakan pembedahan dan diagnostik menyumbang pendapatan terbesar klinik.



## **BAB VI**

### **EVALUASI SISTEM**

#### **6.1 Evaluasi Kinerja Sistem**

Evaluasi kinerja dilakukan terhadap tiga aspek utama yaitu kecepatan query, ketersediaan data, dan kemudahan akses oleh pengguna. Sistem data warehouse yang dibangun telah mampu menjalankan query analitik berbasis OLAP dengan respons yang cukup cepat, berkat penerapan indexing pada kolom penting, partisi tabel fakta berdasarkan waktu dan kategori, serta penggunaan materialized view. Pengujian pada beberapa query seperti laporan pendapatan per spesialisasi dan tren penggunaan obat menunjukkan bahwa waktu respon rata-rata sangat memadai untuk kebutuhan pelaporan rutin dan eksplorasi visualisasi.

Selain itu, dashboard yang dibangun melalui Metabase dan Superset telah berhasil terhubung langsung dengan data warehouse, memungkinkan tim manajemen dan analis mengakses informasi penting seperti jumlah tindakan, total biaya, dan performa layanan tanpa harus menulis query secara manual.

#### **6.2 Kendala Teknis yang Dihadapi**

Selama proses implementasi dan integrasi, terdapat beberapa kendala teknis yang dihadapi, antara lain:

- Format data tidak seragam: Data sumber dalam format CSV memiliki inkonsistensi format, terutama pada kolom tanggal dan harga, sehingga memerlukan proses transformasi tambahan di tahap ETL.
- Data NULL atau kosong: Beberapa kolom penting seperti tanggal transaksi dan total biaya memiliki nilai kosong atau nol, yang membutuhkan penanganan khusus saat ekstraksi dan validasi.
- Tidak tersedia dokumentasi sistem operasional: Kurangnya metadata dan dokumentasi dari sistem sumber menyulitkan proses identifikasi atribut dan relasi antar entitas.
- ETL masih bersifat manual: Proses ETL dijalankan langsung melalui skrip SQL tanpa sistem penjadwalan otomatis seperti cron atau Apache Airflow, sehingga belum sepenuhnya efisien dan berkelanjutan.
- Ketergantungan pada staging manual: Karena belum dilakukan integrasi langsung ke sistem sumber, proses staging masih dilakukan secara semi-manual, rentan terhadap kesalahan input.

#### **6.3 Analisis Keberhasilan dan Kekurangan**

Keberhasilan Sistem:

- Struktur skema star berhasil diimplementasikan, mencakup fakta dan dimensi yang mendukung kebutuhan analisis klinik secara menyeluruh.



- Query OLAP berhasil dijalankan dengan performa baik, menghasilkan laporan analitik yang berguna untuk manajemen dan pengambilan keputusan.
- Dashboard visualisasi dapat diakses dengan mudah oleh pengguna non-teknis, meningkatkan efisiensi proses pelaporan tanpa harus bergantung pada tim teknis.
- Kebutuhan analitik dasar tercapai, seperti analisis penggunaan obat, biaya tindakan, dan KPI klinik.

#### Kekurangan Sistem:

- Implementasi Slowly Changing Dimension (SCD) belum diterapkan secara menyeluruh untuk semua dimensi, seperti perubahan alamat pasien atau spesialisasi dokter dari waktu ke waktu.
- View analitik departemen belum lengkap, sehingga laporan khusus seperti untuk farmasi atau keuangan masih memerlukan tambahan query atau materialized view.
- Belum ada sistem notifikasi atau monitoring ETL, sehingga jika terjadi kegagalan loading data, sistem tidak memberikan peringatan otomatis.
- Skalabilitas jangka panjang belum diuji, terutama dalam hal beban data yang besar jika sistem digunakan dalam produksi nyata.