LAPORAN TUGAS KELOMPOK PERGUDANGAN DATA

Perancangan dan Pengembangan Data Warehouse Fit-Tracker

Disusun Oleh Kelompok 6

Muhammad Bagas Kurnia	121450051
Jihan Putri Yani	121450161
Kemas Veriandra R	122450016
Ganiya Syazwa	122450073
Novelia Adinda	122450104

INSTITUT TEKNOLOGI SUMATERA

LAMPUNG SELATAN

2025

1. Pendahuluan

1.1 Latar Belakang

Di era digital saat ini, industri kebugaran semakin berkembang dengan munculnya perangkat *wearable* dan *aplikasi mobile* yang menghasilkan volume data besar dan beragam. Data tersebut mencakup aktivitas latihan pengguna, data langganan, dan preferensi personalisasi yang penting untuk dianalisis. Pengelolaan data yang optimal diperlukan guna menghasilkan wawasan strategis. *Data warehouse* menjadi solusi yang tepat dengan menyediakan *repository* terpusat dan mendukung analisis historis serta *real-time*. Salah satu pendekatan desain yang efisien adalah penggunaan skema bintang (star schema), yang memudahkan eksekusi *query analitik* untuk keperluan evaluasi performa pengguna, efektivitas fitur, dan perencanaan bisnis berbasis data.

1.2 Rumusan Masalah

1. Bagaimana membangun sistem data warehouse untuk aplikasi kebugaran digital (Fit-Tracker)?
2. Bagaimana merancang struktur data menggunakan skema bintang?
3. Bagaimana menerapkan proses ETL yang efektif dan terstruktur?
4. Bagaimana pemanfaatan hasil data warehouse untuk mendukung keputusan strategis dalam industri kebugaran?

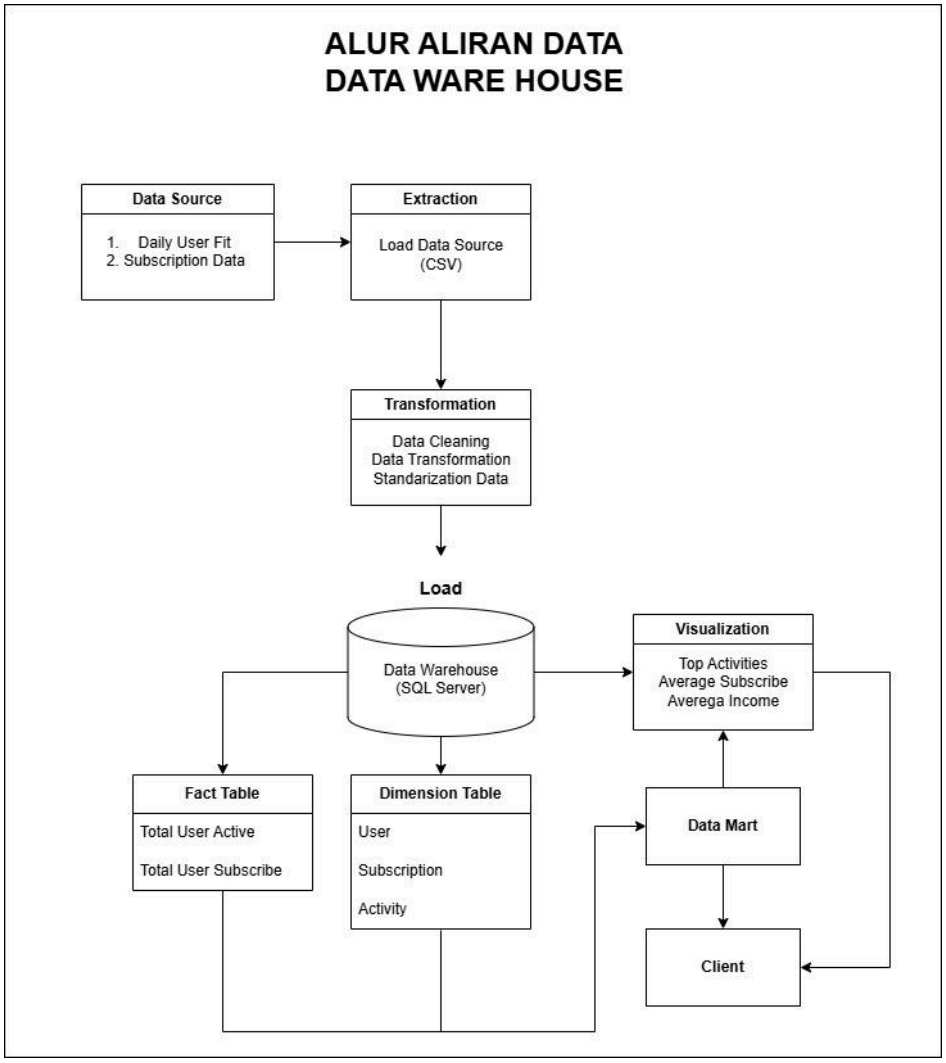
1.3 Ruang Lingkup

Proyek ini mencakup pengolahan data dari aplikasi kebugaran digital fitbit yang meliputi aktivitas latihan, data pengguna, dan informasi langganan. Proses yang dilakukan mencakup tahapan ETL, perancangan data warehouse dengan skema bintang, serta penyajian data melalui visualisasi. Teknologi yang digunakan mencakup *PostgreSQL* untuk penyimpanan data, Python untuk proses ETL, serta *Power BI* atau *Tableau* untuk *dashboard* analitik. Sistem ini dirancang untuk mendukung analisis performa pengguna dan pengambilan keputusan strategis berbasis data.

2. Alur Aliran Data

Alur aliran data dalam sistem *data warehouse Fit-Tracker* dirancang untuk memastikan proses pengolahan data berjalan sistematis, mulai dari sumber data hingga pengguna akhir. Tahapan alur data tersebut dijelaskan sebagai berikut:

1. **Start** – Merupakan tahap inisiasi proses pengolahan data dalam sistem data warehouse.
2. **Data Source** – Data berasal dari sistem aplikasi Fit-Tracker yang mencakup aktivitas latihan pengguna, profil pengguna, dan informasi langganan.
3. **Staging Area** – Pada tahap ini dilakukan proses pembersihan data, termasuk penanganan data null, duplikasi, serta konversi format tanggal dan tipe data lainnya agar sesuai standar.
4. **Transformasi Data** – Data yang telah dibersihkan kemudian distandarisasi dan dipetakan ke dalam struktur tabel dimensi dan tabel fakta sesuai dengan skema multidimensi.
5. **Load ke Data Warehouse** – Data hasil transformasi dimuat ke dalam skema bintang (star schema) yang telah dirancang, guna memastikan keterkaitan antar entitas secara optimal.
6. **Data Mart** – Merupakan subset data yang disusun berdasarkan kebutuhan spesifik dari masing-masing tim pengguna (seperti tim produk, marketing, atau coaching), sehingga memudahkan analisis terfokus.
7. **Client** – Tahap akhir di mana data dapat diakses oleh pengguna untuk keperluan analisis, pembuatan dashboard interaktif, serta pelaporan untuk mendukung pengambilan keputusan strategis.



Gambar 1. *Flowchart* pipeline data

3. Arsitektur Sistem

Rancangan arsitektur sistem yang digunakan dalam pengembangan *data warehouse Fit-Tracker* mengacu pada pendekatan arsitektur tiga lapis (*Three-Tier Architecture*). Pendekatan ini membagi alur sistem menjadi beberapa tahapan yang terstruktur dan saling terintegrasi, yaitu sebagai berikut:

1. **Data Sources** – Merupakan sumber data utama yang terdiri dari data pengguna, data aktivitas latihan, dan data langganan yang diperoleh dari sistem aplikasi Fit-Tracker.
2. **Staging Area** – Berfungsi sebagai tempat untuk melakukan pembersihan, validasi, dan integrasi data dari berbagai sumber. Proses ini mencakup penanganan data tidak konsisten, duplikat, dan transformasi awal sebelum dimuat ke data warehouse.
3. **Data Warehouse** – Merupakan pusat penyimpanan data utama yang terdiri atas tiga komponen:
    - Raw Data: Menyimpan data mentah dari hasil ekstraksi.
    - Metadata: Menyimpan informasi struktural dan deskriptif dari data yang ada.
    - Summary Data: Menyediakan data hasil agregasi dan ringkasan untuk keperluan analitik.
4. **Data Mart** – Menyediakan subset data yang difokuskan untuk kebutuhan analitik tertentu oleh tim yang berbeda, seperti tim produk, tim pelatih/coaching, maupun tim pemasaran.
5. **Client** – Merupakan lapisan akses pengguna akhir, termasuk tim manajemen, analis bisnis, produk, dan pelatih. Mereka dapat mengakses data melalui dashboard, laporan, atau aplikasi analitik lainnya untuk mendukung pengambilan keputusan berbasis data.

Pemisahan arsitektur ini memungkinkan pengelolaan data yang lebih efisien dalam skala besar, meningkatkan keamanan sistem, serta memudahkan proses pemantauan, pemeliharaan, dan pengembangan sistem di masa mendatang seiring pertumbuhan data pengguna dan kompleksitas analisis.

4. ETL Pipeline

4.1 Extract

- Sumber: https://www.kaggle.com/datasets/arashnic/fitbit/data?select=mturkfitbit_export_4.12.16-5.12.16
- Atribut utama: user_id, nama, gender, tinggi, berat, workout_type, intensity_level, durasi, kalori, plan, tanggal, dll.

Tahapan extract ini adalah proses awal dalam menarik data dari berbagai sumber eksternal, dan dalam dataset akan diambil dari platform open-source kaggle yang menyimpan data aktivitas pengguna FitBit, kemudian data akan di extract dengan beberapa atribut penting seperti user_id, gender, workout_type, intensity_level, durasi, kalori, plan, dan tanggal. Proses pengambilan data dilakukan menggunakan Python dengan bantuan pustaka seperti pandas untuk membaca file CSV. Data kemudian disimpan ke dalam staging area sebagai tempat penyimpanan sementara sebelum dilakukan transformasi lebih lanjut.

4.2 Transform

- Pembersihan data: nilai null, duplikat, dan format waktu
- Konversi dan standarisasi tipe data
- Pemetaan ke tabel:
    - dim_user: ID, nama, umur, gender, tinggi, berat
    - dim_workout: jenis latihan, intensitas, otot target
    - dim_subscription: plan, fitur
    - dim_date: hari, bulan, tahun, kuartal
    - fact_fitness_activity: kalori terbakar, durasi, status langganan

Tahapan selanjutnya yaitu Transform, yang merupakan tahapan setelah proses ekstraksi berhasil dilakukan. Tahapan ini akan mencakup berbagai proses pembersihan data seperti penghapusan nilai null, penghapusan data duplikat, serta konversi format tanggal dan tipe data numerik agar konsisten. Selain itu, dilakukan juga pemetaan atribut ke tabel yang sesuai dalam skema bintang, seperti dim_user untuk informasi pengguna, dim_workout untuk detail latihan, dim_subscription untuk langganan, dan dim_date untuk dimensi waktu. Transformasi ini juga mencakup normalisasi data dan pembuatan kolom turunan jika diperlukan, misalnya menghitung umur dari tanggal lahir jika tersedia.

4.3 Load

- Tabel fakta: fact_fitness_activity
- Tabel dimensi: dim_user, dim_workout, dim_subscription, dim_date
- Relasi antar tabel dijaga dengan foreign key

Pada tahap akhir ini, data hasil transformasi akan dimuat ke dalam struktur data warehouse yang terdiri dari tabel fakta dan tabel dimensi. Tabel fact_fitness_activity menyimpan data utama seperti durasi latihan, kalori yang terbakar, dan status langganan. Sementara itu, tabel dimensi (dim_user, dim_workout, dim_subscription, dim_date) diisi terlebih dahulu untuk menjaga konsistensi referensial. Seluruh relasi antar tabel diatur melalui foreign key untuk memastikan integritas data. Proses loading ini bisa dilakukan secara otomatis dengan jadwal tertentu menggunakan script Python atau scheduler seperti Airflow untuk mendukung update berkala (harian atau mingguan).

5. Tools

- PostgreSQL: Penyimpanan dan manajemen DW
- pgAdmin/DBeaver: Antarmuka visual database
- Python (pandas, sqlalchemy): Proses ETL
- Power BI/Tableau: Visualisasi dan dashboard analitik

6. Skrip Query

```
CREATE TABLE dim_user (
    user_id INT PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    gender VARCHAR(10),
    height FLOAT,
    weight FLOAT
);

CREATE TABLE dim_workout (
    workout_id INT PRIMARY KEY,
    workout_type VARCHAR(100),
    intensity_level VARCHAR(50),
    target_muscle VARCHAR(100)
);

CREATE TABLE dim_subscription (
    subscription_id INT PRIMARY KEY,
    plan_name VARCHAR(100),
```

```
    features_included TEXT
);

CREATE TABLE dim_date (
    date_id INT PRIMARY KEY,
    full_date DATE,
    day INT,
    month INT,
    year INT,
    quarter INT
);

CREATE TABLE fact_fitness_activity (
    activity_id INT PRIMARY KEY,
    user_id INT,
    workout_id INT,
    subscription_id INT,
    date_id INT,
    duration_minutes FLOAT,
    calories_burned FLOAT,
    is_subscribed BOOLEAN,
    FOREIGN KEY (user_id) REFERENCES dim_user(user_id),
    FOREIGN KEY (workout_id) REFERENCES dim_workout(workout_id),
    FOREIGN KEY (subscription_id) REFERENCES dim_subscription(subscription_id),
    FOREIGN KEY (date_id) REFERENCES dim_date(date_id)
);
```

Script diatas merupakan **Struktur Skema Bintang** dalam data warehouse system ’Fit Tracker” pertama-tama akan dibangun dengan tabel fakta yaitu “fact_fitness_activity” disini akan merekam beberapa data kualitatif seperti durasi latihan, banyaknya kalori yang terbakar, serta status langganan setiap pengguna. Dalam tabel fakta ini memiliki 4 tabel dimensi yang langsung terhubung yaitu ada “dim_user”, “dim_workout”, “dim_subscription”, dan “dim_date” saling terhubung dengan foreign key, sehingga integritas data terjaga dengan baik dan proses analitik dapat dilakukan secara fleksibel berdasarkan berbagai perspektif, seperti jenis latihan, waktu pelaksanaan, rencana langganan, maupun karakteristik pengguna.

Tiap tabel dimensi mempunyai atribut yang bersifat deskriptif, sebagai contoh dapat dilihat pada “dim_user” yang menyimpan demografis pengguna seperti usia, gender, tinggi, dan berat badan yang berguna untuk analisis segmentasi berdasarkan profil fisik. “Dim_workout” menjelaskan jenis latihan, tingkat intensitas, dan otot yang ditargetkan, sehingga memungkinkan analisis

performa atau preferensi latihan. dim_subscription membantu memahami fitur-fitur yang digunakan oleh pengguna berlangganan, sedangkan dim_date mendukung analisis tren aktivitas berdasarkan waktu (harian, bulanan, kuartalan). Kombinasi skema ini mendukung analisis multidimensi yang dapat digunakan oleh tim bisnis untuk memahami perilaku pengguna, mengevaluasi efektivitas program latihan, dan menyusun strategi personalisasi layanan berbasis data.

Contoh Query Analitik

- **Total Kalori Terbakar per Jenis Latihan**

```
SELECT W.workout_type, SUM(F.calories_burned) AS total_calories
FROM fact_fitness_activity F
JOIN dim_workout W ON F.workout_id = W.workout_id
GROUP BY W.workout_type
ORDER BY total_calories DESC;
```

Query ini akan menghitung jumlah total kalori yang terbakar untuk setiap jenis latihan (workout_type). Dengan mengambil data dari tabel fakta (fact_fitness_activity) lalu mengaitkannya ke (dim_workout) berdasarkan workout_id. Lalu terdapat fungsi agregasi sum() yang akan menjumlahkan jumlah kalori yang terbakar (calories_burned). Setelah itu dikelompokkan dengan (group by) berdasarkan jenis latihan, sehingga kita dapat mengetahui latihan mana yang paling tinggi dalam pembakaran kalori. Dan terakhir akan diurutkan dengan (order by) mulai dari total tertinggi kalori yang terbakar hingga ke yang terendah. Query ini akan membantu mengetahui latihan mana yang paling efektif membakar kalori dan baik untuk melakukan analisis performa fitur latihan.

- **Rata-Rata Durasi Latihan per Pengguna**

```
SELECT U.name, AVG(F.duration_minutes) AS avg_duration
FROM fact_fitness_activity F
JOIN dim_user U ON F.user_id = U.user_id
GROUP BY U.name
ORDER BY avg_duration DESC;
```

Query ini akan menghitung rata-rata durasi latihan tiap pengguna dengan menggunakan fungsi avg() untuk menghitung rata-rata dari kolom durasi latihan per menit atau (duration_minutes). Kemudian melakukan fungsi JOIN dengan tabel (dim_user) untuk mengambil nama pengguna. Setelah itu akan dikelompokkan dengan (group_by) berdasarkan nama pengguna (u.name) agar rata-rata tiap user dihitung secara individu. Hasilnya akan menghasilkan daftar pengguna yang paling rajin atau lambat dalam melakukan latihan. Query ini membantu tim coaching dalam mengidentifikasi pengguna paling aktif atau yang perlu ditingkatkan intensitas latihannya.

- **Distribusi Kalori Bulanan**

```
SELECT D.month, D.year, SUM(F.calories_burned) AS monthly_calories
FROM fact_fitness_activity F
JOIN dim_date D ON F.date_id = D.date_id
GROUP BY D.year, D.month
ORDER BY D.year, D.month;
```

Query ini digunakan untuk menghitung total kalori yang terbakar setiap bulan berdasarkan data aktivitas kebugaran. Langkah pertama kita menggabungkan tabel fakta yang berisi aktivitas kebugaran, dengan tabel dimensi yang menyimpan informasi tanggal, melalui kolom date_id. Setelah itu penggabungan, fungsi agregat SUM() digunakan untuk menjumlahkan nilai calories_burned untuk setiap kombinasi bulan dan tahun. Pengelompokkan dilakukan

berdasarkan kolom D.year dan D.month menggunakan klausa GROUP BY, sehingga setiap baris hasil mempresentasikan total kalori yang terbakar dalam satu bulan tertentu. Hasil akhir diurutkan secara naik berdasarkan tahun dan bulan menggunakan klausa ORDER BY, sehingga distribusi kalori bulanan ditampilkan secara kronologis.

- **Pengguna Berlangganan dengan Aktivitas Tertinggi**

```
SELECT U.name, COUNT(*) AS activity_count
FROM fact_fitness_activity F
JOIN dim_user U ON F.user_id = U.user_id
WHERE F.is_subscribed = TRUE
GROUP BY U.name
ORDER BY activity_count DESC
LIMIT 10;
```

Query ini bertujuan untuk mengidentifikasi 10 pengguna berlangganan yang memiliki jumlah aktivitas kebugaran terbanyak. Proses dimulai dengan menggabungkan tabel fakta (fact_fitness_activity), yang menyimpan data aktivitas kebugaran pengguna, dengan tabel dimensi dim_user, yang berisi informasi pengguna seperti nama, melalui kolom user_id. Selanjutnya, digunakan klausa WHERE F.is_subscribed = TRUE untuk memfilter hanya aktivitas yang dilakukan oleh pengguna yang memiliki status langganan aktif. Fungsi agregat COUNT(*) menghitung jumlah aktivitas yang dilakukan oleh setiap pengguna. Data kemudian dikelompokkan berdasarkan nama pengguna (U.name) menggunakan klausa (group by), sehingga setiap baris hasil merepresentasikan satu pengguna beserta jumlah total aktivitasnya. Hasil akhir diurutkan secara menurun (descending) berdasarkan jumlah aktivitas menggunakan klausa (order_by), dan hanya 10 pengguna teratas yang ditampilkan dengan menggunakan (limit 10). Query ini sangat berguna untuk mengidentifikasi pengguna paling aktif dari kalangan pelanggan berbayar, yang berpotensi menjadi sasaran utama untuk strategi retensi, reward, atau penawaran eksklusif.