

## **Topik:** Identifikasi Arsitektur Big Data & Simulasi Pipeline Ekosistem Ride-Hailing

- **Nama Lengkap:** Josia Given Santoso
- **NIM:** 36230035
- **Program Studi:** Sains Data
- **Dosen Pengampu:** Eko Wahyu Prasetyo, S.T., M.Eng.

### **Tautan**

- **Repository Github:** <https://github.com/josgiv/bdmp-pt2-geolocation>

### **A. Pendahuluan & Arsitektur**

Laporan ini membahas perancangan dan simulasi ekosistem Big Data untuk perusahaan layanan *on-demand* (seperti Gojek/Grab). Simulasi ini dirancang untuk menangani volume data masif (skala 30-40 juta baris data) menggunakan pendekatan *Modern Data Stack* berbasis Python. Tujuan utamanya adalah membangun *pipeline* data dari hulu ke hilir yang efisien dan hemat memori.

Berikut adalah pemetaan 5 komponen utama dalam arsitektur yang dibangun:

#### **1. Data Sources (Sumber Data)**

Sistem mensimulasikan tiga jenis data utama yang merepresentasikan aktivitas nyata operasional perusahaan:

- **Transactional Data (Orders)**

Data terstruktur yang mencakup informasi krusial seperti ID pesanan, ID pengguna, ID pengemudi, nilai transaksi (GMV), serta status akhir pesanan (selesai atau dibatalkan).

- **Telemetry Data (GPS)**

Data semi-terstruktur berupa titik koordinat (latitude/longitude) dan kecepatan pengemudi yang dikirimkan secara *real-time*.

- **System Logs**

Data tidak terstruktur dari server yang mencatat kesehatan sistem, termasuk latensi layanan dan tingkat kesalahan (*error rate*) dari berbagai *microservices*.

## 2. Data Ingestion (Ingesti Data)

Proses masuknya data ke dalam sistem dilakukan melalui dua metode simulasi:

- **Batch Ingestion**

Menggunakan skrip optimasi numerik untuk membangkitkan data historis dalam jumlah besar dan langsung menyimpannya ke penyimpanan disk.

- **Streaming Simulation**

Menggunakan mekanisme *generator* untuk meniru aliran data yang masuk terus-menerus (seperti Apache Kafka). Data ini ditangkap dan disimpan sementara dalam format JSON Lines (JSONL) untuk mensimulasikan antrian pesan (*message queue*).

## 3. Data Storage (Penyimpanan)

Penyimpanan data menerapkan konsep Data Lake dengan format file Apache Parquet. Pemilihan Parquet didasarkan pada efisiensi penyimpanan berbasis kolom (*columnar storage*) dan penggunaan kompresi 'Snappy'. Ini terbukti jauh lebih hemat ruang penyimpanan dan lebih cepat saat proses pembacaan (I/O) dibandingkan format teks biasa seperti CSV.

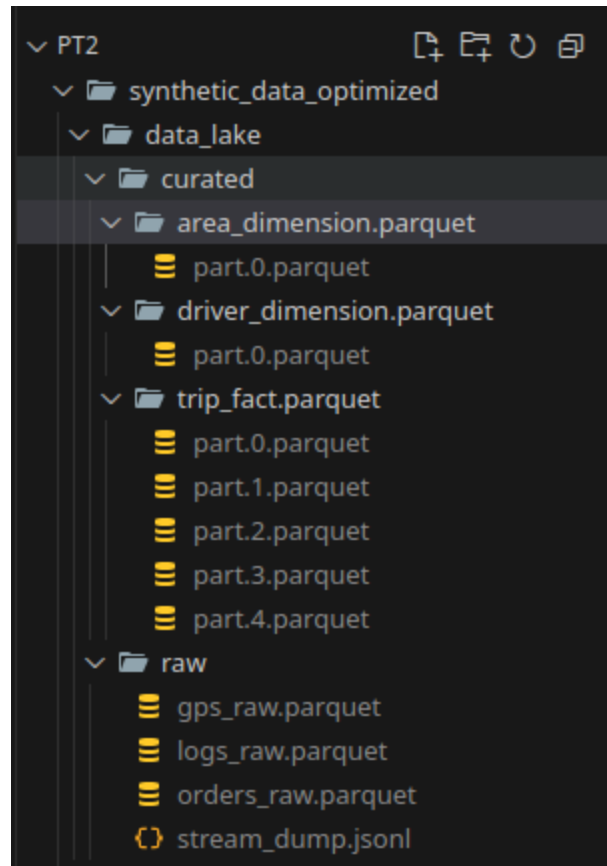
Struktur penyimpanan dibagi menjadi dua zona utama:

- **Raw Zone**

Zona pendaratan untuk data mentah yang belum diubah (contoh: `orders_raw.parquet`).

- **Curated Zone**

Zona penyimpanan untuk data bersih yang telah melalui proses transformasi dan siap dianalisis (contoh: `trip_fact.parquet`).



#### 4. Data Processing (Pemrosesan)

Pemrosesan data dilakukan menggunakan mesin komputasi terdistribusi (Dask) dan manajemen memori efisien (PyArrow). Metode utama yang diterapkan adalah Lazy Evaluation.

Dalam metode ini, sistem tidak memuat seluruh 30 juta baris data ke dalam RAM secara bersamaan, yang akan menyebabkan kegagalan sistem (*crash*). Sebaliknya, sistem membangun grafik tugas (*task graph*) dan hanya melakukan eksekusi komputasi saat hasil akhir benar-benar diminta. Hal ini memungkinkan pengolahan Big Data dilakukan pada perangkat dengan spesifikasi RAM terbatas.

#### 5. Data Visualization (Visualisasi)

Komponen terakhir bertujuan mengubah data mentah menjadi wawasan bisnis. Alat visualisasi digunakan untuk membuat:

- **Grafik Statistik**

Untuk memantau tren performa dan KPI.

- **Peta Geospasial (Heatmap)**

Untuk memetakan persebaran permintaan pelanggan dan posisi mitra pengemudi.

## B. Alur Simulasi Data

Bagian ini menjelaskan langkah-langkah teknis yang dilakukan dalam *notebook* simulasi, mulai dari penyiapan lingkungan hingga pembentukan tabel analitik.

```
Generating orders_raw... (30,000,000 rows)
13%|██████| 4000000/30000000 [00:05<00:36, 716512.08row/s]
100%|██████████| 30000000/30000000 [00:37<00:00, 794965.92row/s]
Done: d:\Dev_Drive\Coding Project Files\Uni_Assignment\Data-Management\PT2\synthetic_data_optimized\data_lake\raw\orders_raw.parquet
Generating gps_raw... (15,000,000 rows)
100%|██████████| 15000000/15000000 [00:05<00:00, 2577086.30row/s]
Done: d:\Dev_Drive\Coding Project Files\Uni_Assignment\Data-Management\PT2\synthetic_data_optimized\data_lake\raw\gps_raw.parquet
Generating logs_raw... (40,000,000 rows)
100%|██████████| 40000000/40000000 [00:31<00:00, 1274360.24row/s]
Done: d:\Dev_Drive\Coding Project Files\Uni_Assignment\Data-Management\PT2\synthetic_data_optimized\data_lake\raw\logs_raw.parquet

All Data Generated Successfully!
```

### 1. Inisialisasi & Data Generation

Pada tahap awal, sistem menyiapkan struktur direktori *Data Lake* dan melakukan pembangkitan data sintetis. Data dibuat dalam potongan-potongan kecil (*chunks*) untuk menjaga stabilitas memori.

#### Output Tahap Ini:

Terbentuknya tiga file utama di zona *Raw*:

- File riwayat transaksi pesanan.
- File data telemetri/GPS driver.
- File log aktivitas sistem backend.

### 2. Proses ETL (Extract, Transform, Load)

Tahap ini merupakan inti dari pengolahan data, di mana data mentah diubah menjadi tabel-tabel yang bermakna bisnis:

## 1) Pembentukan Tabel Fakta (Trip Fact)

Sistem membaca data transaksi mentah dan melakukan transformasi waktu. Durasi perjalanan dihitung dengan menselisihkan waktu selesai pesanan dengan waktu pemesanan. Selain itu, dilakukan ekstraksi fitur "jam" untuk keperluan analisis waktu sibuk.

```
... Processing Analytics with Dask...
Saving Trip Fact...
Trip Fact Sample:
```

	order_id	t_created	t_end	final_status	payment_method	gmv_idr	user_id	driver_id	pickup_lat	pickup_lon	trip_minutes	hour	is_cancelled
0	0	2021-04-27 13:04:23	2021-04-27 13:46:10	completed	gopay	60565	972196	25618	-0.538719	113.686440	41.783333	13	False
1	1	2025-04-11 19:06:55	2025-04-11 19:46:06	completed	gopay	140374	118564	40740	-1.893331	111.903938	39.183333	19	False
2	2	2021-04-23 11:02:41	2021-04-23 11:45:20	completed	gopay	25017	447746	5855	1.766638	112.729195	42.650000	11	False
3	3	2023-09-12 03:29:56	2023-09-12 04:00:28	completed	gopay	28286	309191	3763	-3.018361	112.875572	30.533333	3	False
4	4	2025-12-15 17:56:31	2025-12-15 18:53:29	completed	gopay	75808	51476	37281	-2.512621	112.115448	56.966667	17	False

- Hasil: Tabel `trip_fact` yang berisi metrik durasi perjalanan (menit) dan waktu transaksi.

## 2) Pembentukan Dimensi Driver (Driver Dimension)

Dilakukan proses agregasi (pengelompokan) data berdasarkan identitas pengemudi. Sistem menghitung profil kinerja setiap mitra, meliputi:

```
Building Driver Dimension...
Driver Dimension Sample:
```

	driver_id	total_orders	cancelled_orders	avg_trip_minutes	avg_gmv	cancel_rate	avg_speed
0	25618	622	88	32.507985	80405.139871	0.141479	30.019986
1	40740	600	79	33.871361	79911.235000	0.131667	30.318771
2	5855	644	86	31.756030	82872.998447	0.13354	29.343283
3	3763	632	76	33.710153	77791.806962	0.120253	30.601243
4	37281	609	76	34.000575	81286.903120	0.124795	28.948815

- Total pesanan yang diselesaikan vs dibatalkan.
- Rata-rata pendapatan (GMV).
- Rata-rata kecepatan berkendara (diambil dari data GPS).
- Hasil: Tabel `driver_dimension` yang memberikan gambaran lengkap kinerja individu mitra.

### 3) Pembentukan Dimensi Area (Area Dimension)

Sistem membagi peta operasional menjadi kotak-kotak grid geografis (*binning*). Setiap transaksi dipetakan ke dalam grid tersebut untuk menghitung kepadatan permintaan.

```
.. Building Area Dimension...
Area Dimension Sample:
```

	pickup_lat_bin	pickup_lon_bin	orders	cancel_rate	avg_gmv
0	-0.54	113.689995	94	0.12766	81166.127660
1	-1.89	111.899994	122	0.147541	75644.540984
2	1.77	112.729996	95	0.147368	87825.694737
3	-3.02	112.879997	122	0.106557	75425.049180
4	-2.51	112.119995	93	0.139785	75916.290323

- Hasil: Tabel `area_dimension` yang memetakan koordinat wilayah dengan jumlah permintaan (*demand*).

## C. Hasil Analisis Data

Berdasarkan simulasi data yang telah diproses, berikut adalah hasil analisis utama yang diperoleh:

### 1. Ringkasan Kinerja Bisnis (KPI)

Sistem menghasilkan tabel ringkasan yang mencakup metrik vital perusahaan dari total dataset yang diolah:

```
... Calculating KPIs...
```

	total_orders	completed	cancelled	cancel_rate	gmv_total_idr	trip_minutes_p90
0	30000000	26400726	3599274	0.119976	2.400204e+12	54.483333

- **Total Volume Order**

Menampilkan jumlah total transaksi dalam sistem (simulasi 30 juta baris).

- **Status Pesanan**

Rasio antara pesanan yang sukses diselesaikan dibandingkan dengan pesanan yang dibatalkan (*Cancel Rate*).

- **Total GMV**

Akumulasi nilai transaksi kotor yang beredar dalam platform.

## 2. Deteksi Kecurangan (Fraud Detection)

Dengan memanfaatkan tabel `driver_dimension`, sistem melakukan pengurutan data untuk mengidentifikasi mitra dengan perilaku anomali.

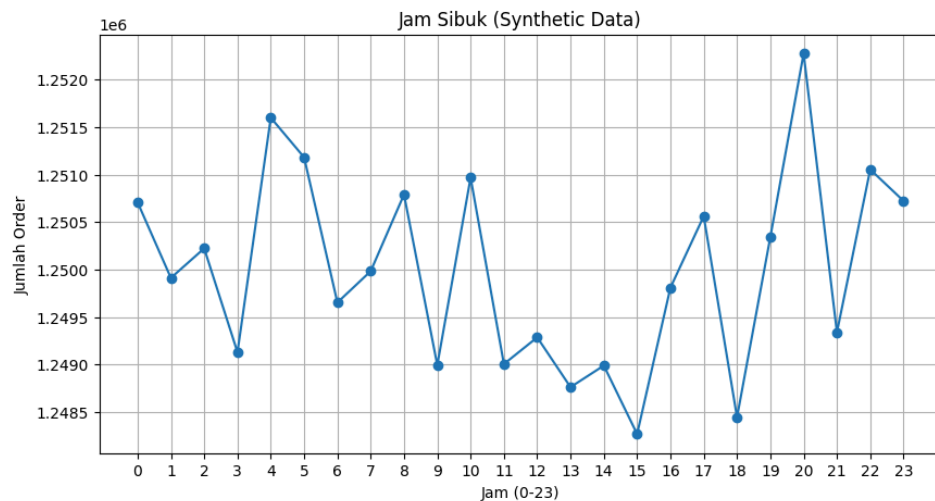
- **Analisis**

Sistem berhasil mendeteksi "Top Risky Drivers", yaitu mitra yang memiliki tingkat pembatalan (*cancel rate*) ekstrem (mendekati 100%) atau pola aktivitas yang tidak wajar. Data ini menjadi input krusial bagi tim Fraud untuk tindakan penangguhan akun (*suspend*).

## 3. Analisis Waktu & Lokasi (Spatiotemporal)

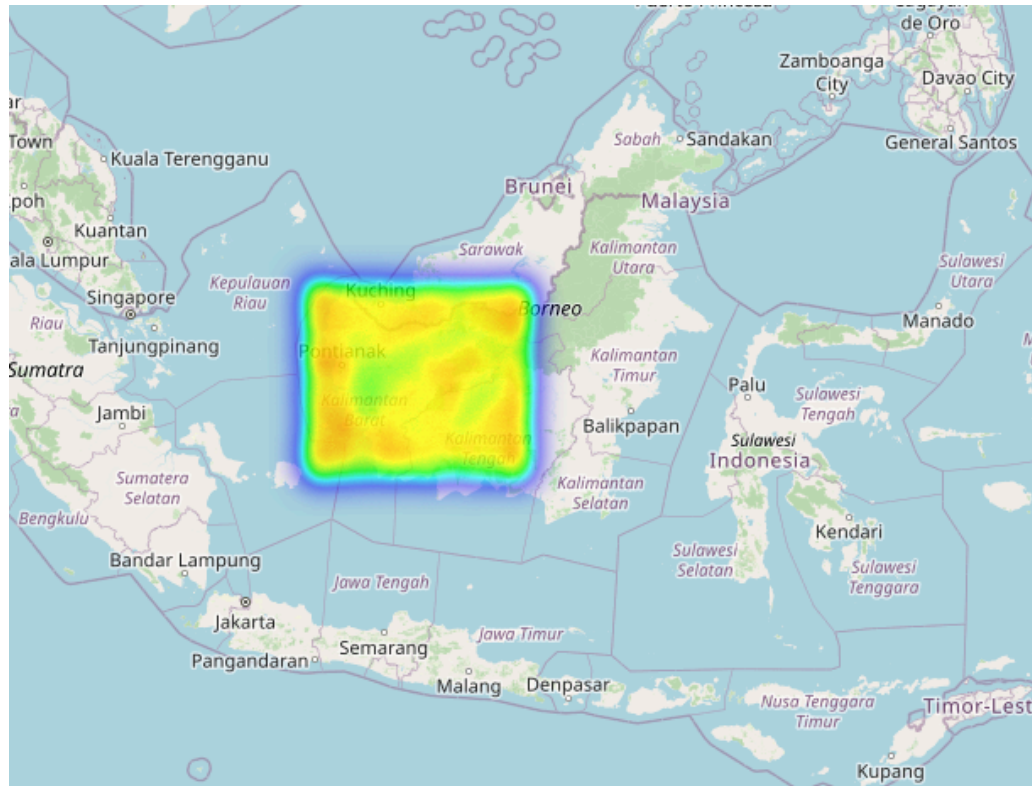
Visualisasi data menghasilkan wawasan mengenai perilaku pasar:

- **Tren Jam Sibuk**



Grafik garis menunjukkan kurva beban pesanan sepanjang hari (00:00 - 23:00), memungkinkan perusahaan mengidentifikasi jam puncak (*peak hours*) untuk

penerapan harga dinamis (*dynamic pricing*).



- **Peta Panas (Heatmap)**

Visualisasi geospasial menyoroti area-area dengan densitas pesanan tertinggi (warna merah pekat), yang berguna untuk strategi alokasi mitra pengemudi.