

Building a Data Pipeline for Airline Flight Performance Analysis Using Batch Processing

Aditiya Budi Pratama

A horizontal white line with a small orange rectangular bar in the middle, located below the author's name.



Aditiya Budi Pratama

Education

*Universitas Darma Persada
S1 – Teknologi Informasi*

Working

Data Steward

Overview Project

- **ETL with Airflow**
Membuat pipeline menggunakan Airflow
- **Python**
Meningkatkan kualitas data untuk analisis dan pemodelan
- **Database**
Menyimpan dataset yang telah di transromasi
- **Apache Airflow**
Pipeline airflow per batch

Abstract geometric shapes on the left side of the slide, including a large black parallelogram, a light orange parallelogram, and a darker orange parallelogram, all slanted to the right.

Project Background



Membangun pipeline data batch yang mengotomatiskan proses extract, transform, dan load (ETL) untuk menganalisis kinerja maskapai penerbangan, khususnya dalam hal keterlambatan keberangkatan dan kedatangan. Pipeline ini menjadwalkan pemrosesan data menggunakan Apache Airflow, menyimpan data hasil transformasi di PostgreSQL, dan membuat visualisasi data hasil transformasi ke Metabase untuk dianalisis lebih lanjut.



- File Parquet sebagai *intermediate storage* hasil ekstraksi dan transformasi
- Tabel analitik di PostgreSQL berisi ringkasan rata-rata delay per maskapai
- UI Airflow yang menampilkan status pipeline dan log proses
- Otomatisasi batch processing setiap 2 menit yang dapat diskalakan

Mengapa penting ???


- **Efisiensi operasional maskapai** : Data ini memberi wawasan tentang maskapai mana yang paling sering mengalami keterlambatan.
- **Pengambilan keputusan berbasis data** : Informasi dari pipeline ini mendukung pengambilan keputusan strategis oleh operator bandara, maskapai, dan regulator.
- **Otomatisasi pipeline** : Dengan adanya Airflow, pipeline menjadi otomatis, terjadwal, dan mudah dipantau.
- **Skalabilitas** : Data akhir disimpan di Postgres yang memungkinkan analisis skala besar dan integrasi dengan Metabase.



Keuntungan untuk siapa ?

- **Maskapai Penerbangan** : Evaluasi performa operasional dan peningkatan layanan
- **Bandara** : Perencanaan jadwal yang lebih baik dan pengurangan antrian
- **Regulator (DOT)** : Monitoring akurasi pelaporan dan kepatuhan maskapai
- **Analisis Data/BI** : Dasar analitik keterlambatan dan korelasi antar variabel
- **Publik/Penumpang** : Transparansi informasi keterlambatan maskapai



A large black parallelogram on the left side of the slide, with two overlapping orange parallelograms in front of it, one slightly to the right and down from the other.

Problem Statement

Latar Belakang :

Keterlambatan penerbangan merupakan salah satu masalah operasional utama dalam industri penerbangan. Delay ini tidak hanya merugikan penumpang dari sisi waktu, tetapi juga menyebabkan biaya tambahan bagi maskapai dan bandara, serta menurunkan kepuasan pelanggan.

Permasalahan Khusus :

- Tidak adanya sistem otomatis untuk mengolah dan menganalisis data penerbangan harian dalam skala besar.
- Data mentah penerbangan (CSV) memiliki volume besar dan tidak siap untuk dianalisis langsung (butuh pre-processing).
- Belum ada ringkasan terstruktur (summary table) tentang keterlambatan per maskapai.
- Informasi delay tidak digunakan untuk memantau performa secara reguler oleh sistem backend otomatis.

Tujuan Project

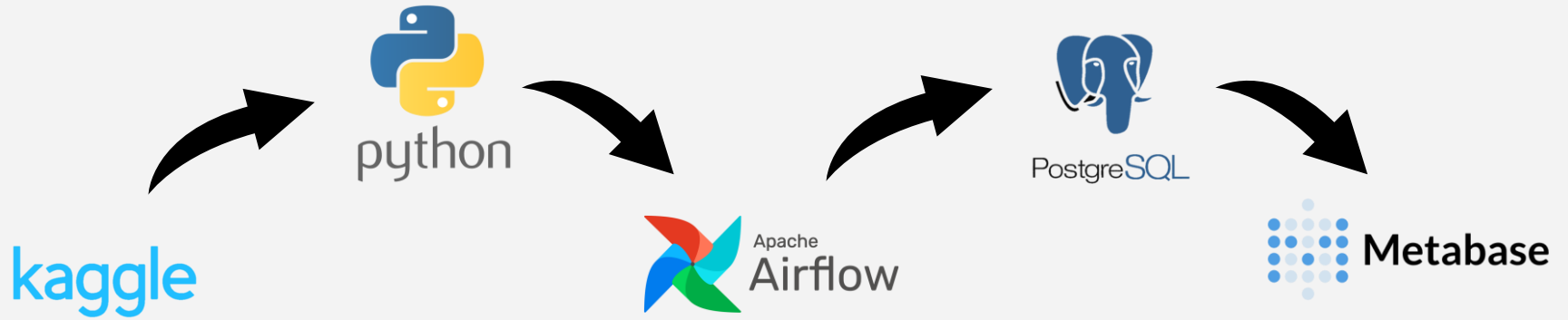
- Menjadwalkan pemrosesan data penerbangan setiap 2 menit secara otomatis melalui Airflow.
- Mengekstraksi data mentah dari file CSV.
- Melakukan transformasi data membersihkan kolom yang relevan (DepDelay, ArrDelay).
- Menghitung rata-rata delay keberangkatan dan kedatangan per maskapai (UniqueCarrier).
- Menyimpan hasil ringkasan ke dalam format efisien (Parquet) dan mengunggahnya ke PostgreSQL.
- Mendukung visualisasi di tools Metabase.

Metrik Kesuksesan

- Pipeline reliability
- Data availability
- Average delay metrics
- Coverage rate
- Query readiness

A large black parallelogram is positioned on the left side of the slide. Overlapping its bottom edge are two orange parallelograms, one in a lighter shade and one in a darker shade, creating a layered effect.

Data Platform Understanding



Abstract geometric shapes on the left side of the slide: a large black parallelogram, a medium yellow parallelogram, and a smaller orange parallelogram, all slanted to the right.

Data Understanding

Data Source

kaggle

[hflights.csv](#)

Struktur Data

Year	int64
Month	int64
DayofMonth	int64
DayOfWeek	int64
DepTime	float64
ArrTime	float64
UniqueCarrier	object
FlightNum	int64
TailNum	object
ActualElapsedTime	float64
AirTime	float64
ArrDelay	float64
DepDelay	float64
Origin	object
Dest	object
Distance	int64
TaxiIn	float64
TaxiOut	float64
Cancelled	int64
CancellationCode	object
Diverted	int64

Tinjauan Kualitas Data

Case	Keterangan
Volume Besar	Cocok untuk data mining dan analitik skala besar.
Representatif	Mencakup ratusan bandara dan maskapai di seluruh AS.
Kaya fitur	Banyak kolom penyebab delay, cancel, diverted, dll.

Masalah Kualitas Data

Masalah	Keterangan
Missing Values	Kolom DepDelay dan ArrDelay sering kosong jika penerbangan dibatalkan.
Duplikasi	Kemungkinan ada entri duplikat untuk penerbangan yang sama (perlu dedup).
Null Delay Reason	Kolom delay penyebab (*_Delay) banyak yang null walaupun ada delay.
Zero/Negative Delay	Perlu validasi karena kadang delay < 0 bisa terjadi karena koreksi waktu.
Inconsistency	Nama maskapai atau bandara mungkin berbeda format atau casing.

On the left side of the slide, there are three overlapping geometric shapes: a large black parallelogram at the top, a medium-sized light orange parallelogram in the middle, and a smaller dark orange parallelogram at the bottom, all slanted to the right.

Transformation & Consideration

Latensi :

- Batch 2 menit

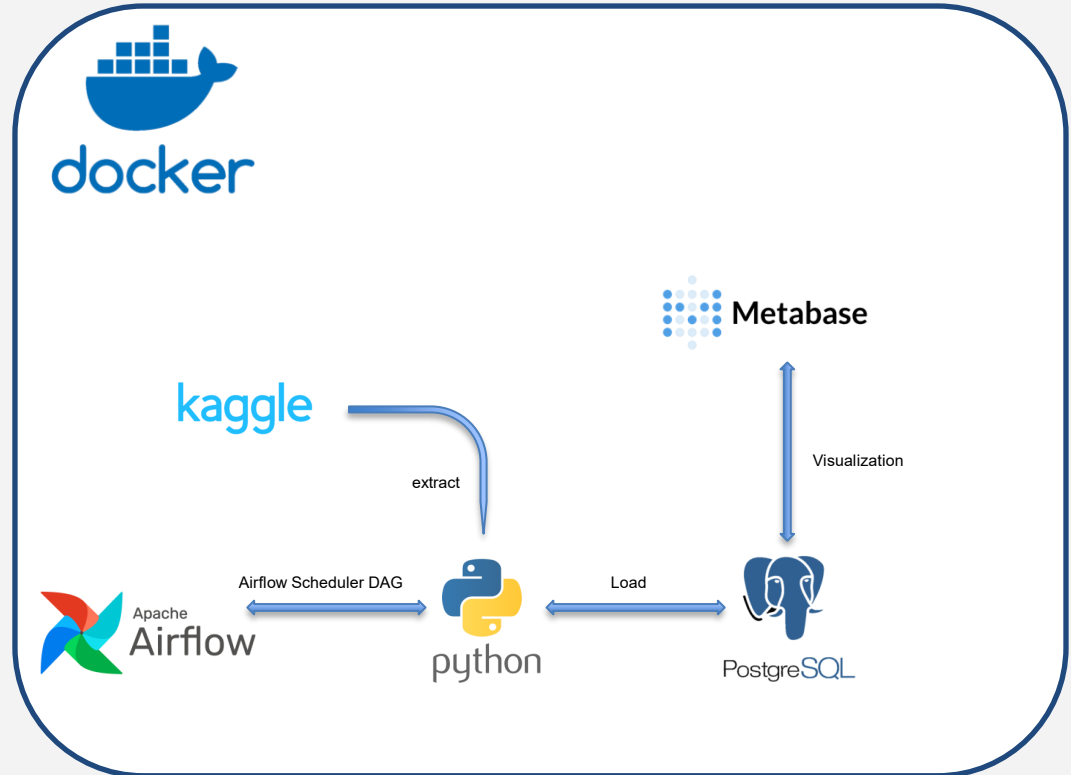
Ukuran Data :

- 227496 baris data CSV

Alat Pemrosesan :

- Python
- Apache Airflow
- PostgreSQL
- Metabase

Diagram Arsitektur



A large black parallelogram on the left side of the slide, with two overlapping orange parallelograms positioned below it, creating a modern, abstract geometric design.

Data Modeling (Business)

Hasil Tranformasi Data :

flights_summary		
A-Z	UniqueCarrier	text
123	AVG_DEP_DELAY	float8
123	AVG_ARR_DELAY	float8

flights_timeofday		
A-Z	UniqueCarrier	text
123	DepDelay	float8
123	ArrDelay	float8
123	DepTime	float8
A-Z	TimeOfDay	text

A large black parallelogram on the left side of the slide, with two overlapping orange parallelograms positioned below it, creating a modern, abstract geometric design.

Conclusion & Recommendation

Scripts DAG :

```
dags > airflow_pipeline.py > ...
1 import os
2 import sys
3 from airflow import DAG
4 from airflow.operators.python import PythonOperator
5 from airflow.utils.task_group import TaskGroup
6 from datetime import datetime, timedelta
7
8 # Pastikan /opt/airflow sudah ada di sys.path sebelum import scripts
9 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
10 if BASE_DIR not in sys.path:
11     sys.path.insert(0, BASE_DIR)
12
13 from scripts.extract_data import extract_etl
14 from scripts.transform_data import transform_etl
15 from scripts.load_data import load_etl
16
17 default_args = {
18     'owner': 'airflow',
19     'start_date': datetime(2024, 1, 1),
20     'retries': 1,
21     'retry_delay': timedelta(minutes=1),
22 }
23
24 with DAG(
25     'airflights_etl_pipeline',
26     default_args=default_args,
27     description='ETL pipeline',
28     schedule_interval='*/2 * * * *', # setiap 2 menit
29     catchup=False
30 ):
```

```
dags > airflow_pipeline.py > ...
33 with TaskGroup('extract_task', tooltip='Extract Batch') as extract_task:
34     # prev_task = None
35     for i in range(1, 6): # contoh 5 batch
36         extract_data = PythonOperator(
37             task_id=f'extract_{i}',
38             python_callable=extract_etl,
39             op_kwargs={
40                 'input_file': '/opt/airflow/data/hflights.csv',
41                 'output_file': f'/opt/airflow/data/output/extracted_batch{i}.csv',
42                 'batch_number': i,
43                 'batch_size': 5000
44             }
45         )
46         # if prev_task:
47         #     prev_task >> extract_data
48         # prev_task = extract_data
49
50 transform_data = PythonOperator(
51     task_id='transform',
52     python_callable=transform_etl,
53     op_kwargs={
54         'input_file': [
55             '/opt/airflow/data/output/extracted_batch1.csv',
56             '/opt/airflow/data/output/extracted_batch2.csv',
57             '/opt/airflow/data/output/extracted_batch3.csv',
58             '/opt/airflow/data/output/extracted_batch4.csv',
59             '/opt/airflow/data/output/extracted_batch5.csv',
60         ],
61         'output_file': '/opt/airflow/data/output/transformed.parquet',
```

Scripts Extract :



```
scripts > extract_data.py > ...
1 import pandas as pd
2
3
4 def extract_etl(
5     input_file,
6     output_file,
7     batch_number,
8     batch_size=5000
9 ):
10     skip_rows = 1 + (batch_number - 1) * batch_size
11     df = pd.read_csv(
12         input_file,
13         skiprows=range(1, skip_rows),
14         nrows=batch_size
15     )
16     df.to_csv(output_file, index=False)
17     print(f"Extracted batch {batch_number} to {output_file}")
18
```

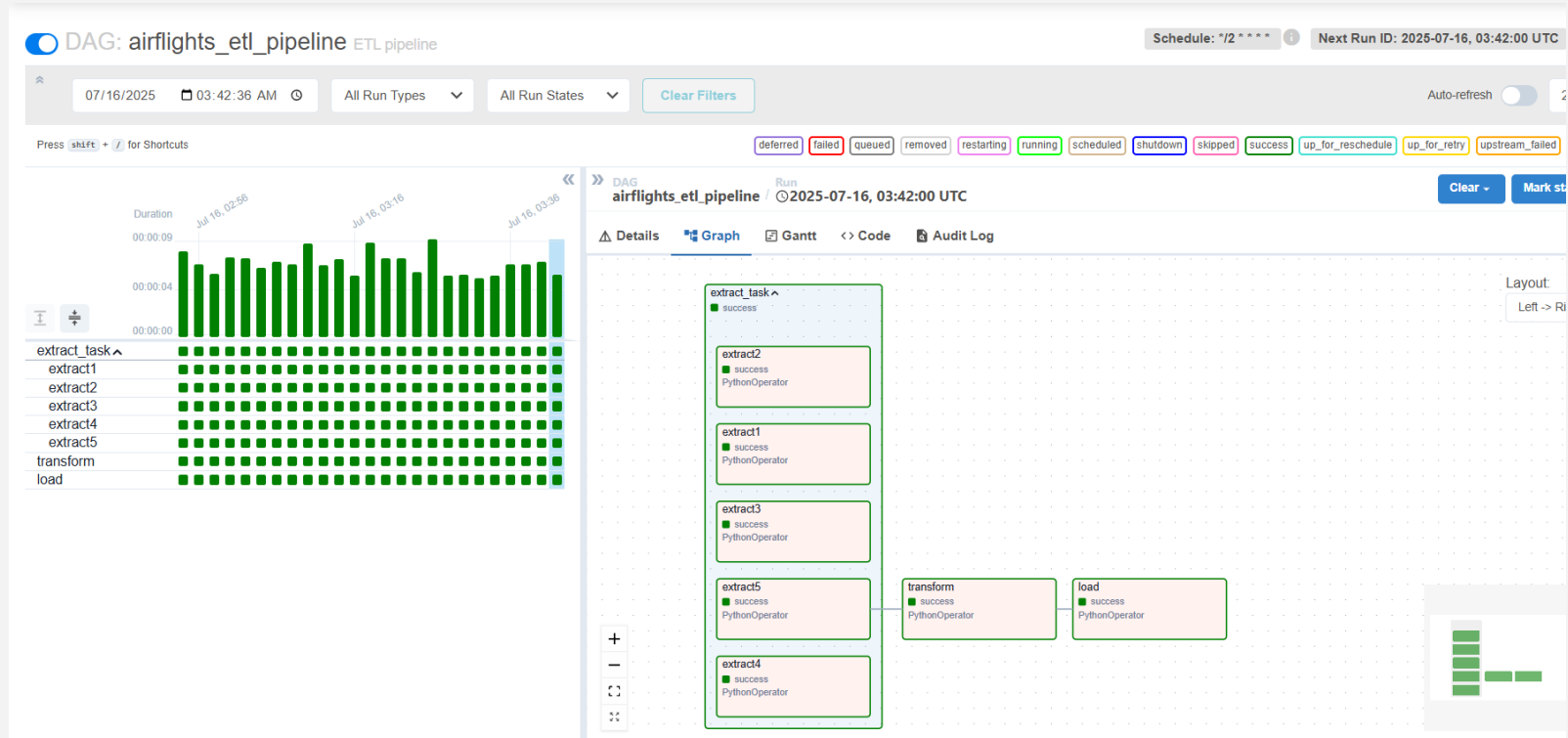
Scripts Transform :

```
scripts > transform_data.py > ...
1 import pandas as pd
2
3
4 def transform_etl(input_file, output_file_summary, output_file_full):
5
6     # gabungkan semua file CSV
7     df_list = []
8     for file in input_file:
9         df = pd.read_csv(file)
10         df_list.append(df)
11     combined_df = pd.concat(df_list, ignore_index=True)
12     print(f'total rows after combining: {len(combined_df)}')
13
14     # pastikan kolom yang relevan ada
15     required_cols = ['UniqueCarrier', 'DepDelay', 'ArrDelay']
16     missing_cols = [col for col in required_cols if col not in df.columns]
17
18     if missing_cols:
19         raise KeyError(f"Kolom berikut tidak ditemukan di CSV: {missing_cols}")
20
21     # hanya kolom yang relevan
22     df = combined_df[required_cols + (['DepTime'] if 'DepTime' in combined_df
23                                     .columns else 'Tidak ada kolom DepTime')]
24
25     # bersihkan data: drop baris dengan NA di kolom penting
26     before = len(df)
27     df = df.dropna(subset=['DepDelay', 'ArrDelay'])
28     after = len(df)
29     print(f"Dropped {before - after} rows with missing DepDelay or ArrDelay.")
30
```

Scripts Load :

```
scripts > load_data.py > ...
5 def load_etl(input_file,
6
7     df = pd.read_parquet(input_file)
8     df2 = pd.read_parquet(input_file_day)
9
10
11     # Buat koneksi SQLAlchemy
12     connection_uri = (
13         f"postgresql://{db_user}:{db_pass}@{db_host}:{db_port}/{db_name}"
14     )
15     engine = create_engine(connection_uri)
16
17     # Load ke PostgreSQL
18     with engine.begin() as connection:
19         df.to_sql(table_name,
20                 connection,
21                 if_exists='replace',
22                 index=False)
23
24     print(f"Data loaded to PostgreSQL table {table_name}")
25
26
27     with engine.begin() as connection:
28         df2.to_sql(
29             table_name_full,
30             connection,
31             if_exists='replace',
32             index=False)
33
34     print(f"Data loaded to PostgreSQL table {table_name_full}")
35
```

Tampilan Proses ETL Airflow



Keterlambatan Penerbangan

Flights Summary, Max of Avg Dep Delay, Grouped by UniqueCarrier



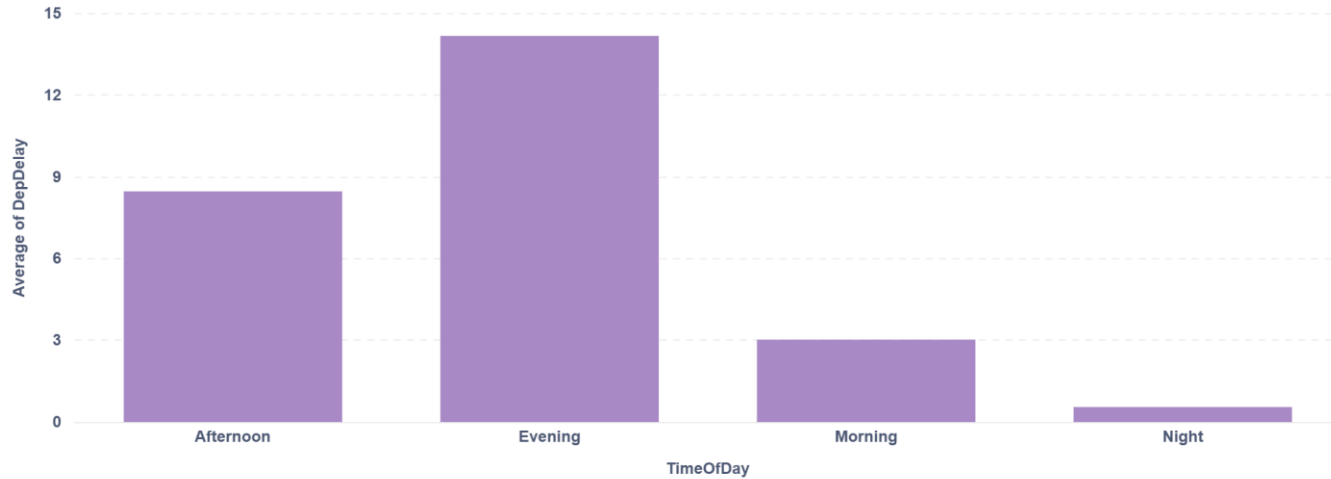
Keterlambatan Kedatangan Penerbangan

Flights Summary, Max of Avg Arr Delay, Grouped by UniqueCarrier - Duplicate



Keterlambatan Waktu Penerbangan

Flights Timeofday, Average of DepDelay, Grouped by TimeOfDay



Kemampuan Platform

- Menjalankan ETL setiap 2 menit secara otomatis menggunakan **Apache Airflow** dalam container **Docker**.
- Mengelola dan menyimpan data secara terstruktur di dalam **PostgreSQL** sebagai data warehouse
- Melakukan transformasi data yang relevan untuk analisis seperti mendapatkan rata-rata keterlambatan penerbangan setiap maskapai, mendapatkan rata-rata keterlambatan tiba di bandara tujuan dan mendapatkan waktu penerbangan paling banyak keterlambatan
- Menyajikan data secara visual menggunakan **Metabase** untuk membuat stakeholder memahami insight secara tepat.

Keterbatasan Platform

- Transformasi menggunakan Python script, ini membatasi performa pada volume data besar
- Scheduler bisa overload kalau DAG terlalu berat atau banyak task paralel > CPU/memori.
- Visualisasi hanya sebaik performa database → kalau query lambat, dashboard juga lambat.
- Volume mapping kadang error jika folder Windows tidak di-share atau permission salah.

A large, stylized graphic on the left side of the slide. It consists of a blue outline of a person's head and shoulders. Inside the head is a large orange circle with a smaller orange circle in the center. Inside the shoulders is a large orange circle with a smaller orange circle in the center.

**Terima
Kasih.**