

Final Project DE

This final project is about Dockerize ETL Pipeline using ETL tools Airflow that extract Public API data from PIKOBAR, then load into MySQL (Staging Area) and finally aggregate the data and save into PostgreSQL.

A. ETL Architecture Diagram and Integration Design Diagram

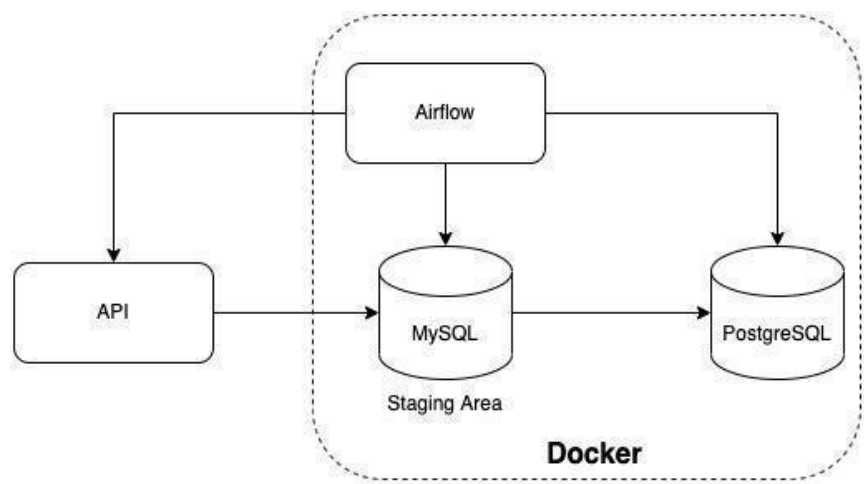


Figure 1. ETL Architecture Diagram

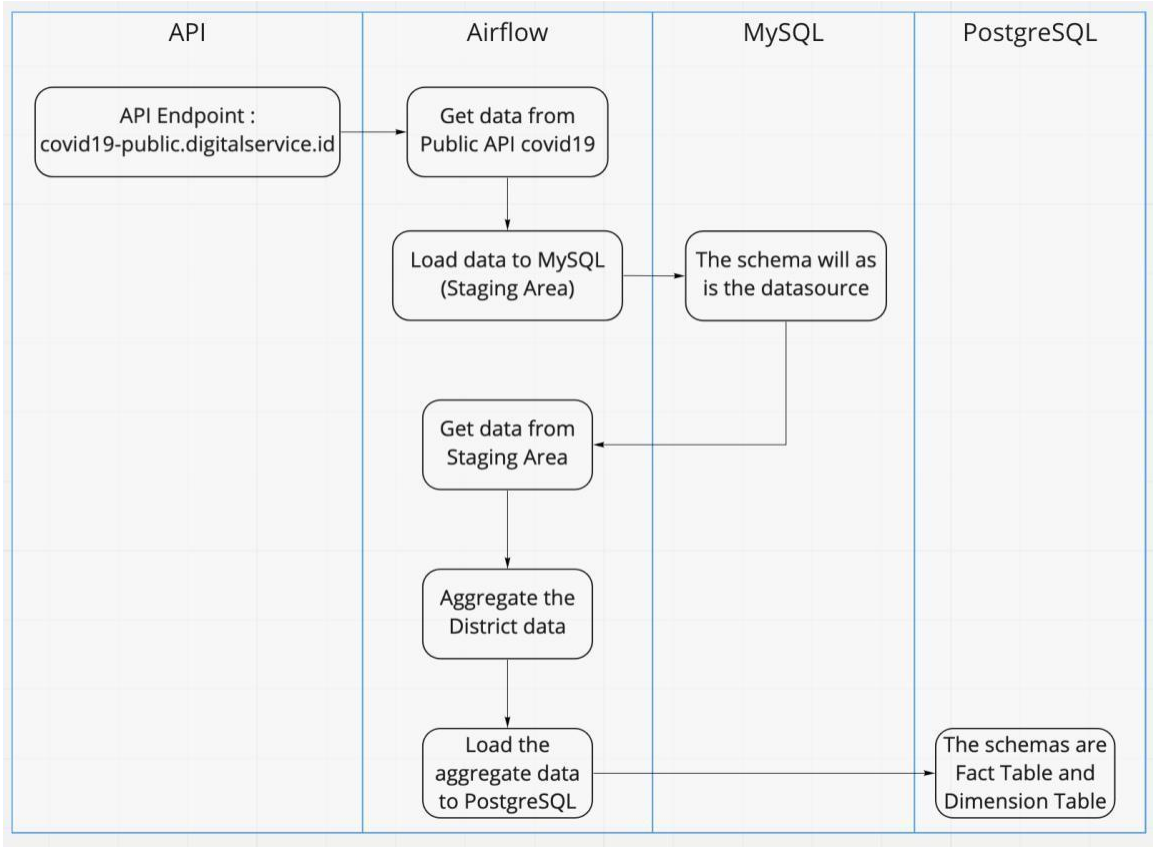


Figure 2. Integration Design Diagram

B. API Specifications

API Endpoint	http://103.150.197.96:5005
Path	/api/v1/rekapitulasi_v2/jabar/harian
Content-Type	application/json
Request Parameter	level

Request method	GET
----------------	-----

C. API Request Example

```
curl -X GET "http://103.150.197.96:5005/api/v1/rekapitulasi_v2/jabar/harian?level=kab"  
-H "accept: application/json"
```

D. API Response Example

```
{  
  "status_code":  
200, "data": {  
  "metadata": {  
    "last_update": null  
  },  
  "content": [  
    {  
      "tanggal": "2020-08-05",  
      "kode_prov": "32",  
      "nama_prov": "Jawa Barat",  
      "kode_kab": "3204",  
      "nama_kab": "Kabupaten  
Bandung", "SUSPECT": 2210,  
      "CLOSECONTACT": 274,  
      "PROBABLE": 26,  
      "suspect_diisolasi": 31,  
      "suspect_discarded": 2179,  
      "closecontact_dikarantina": 0,  
      "closecontact_discarded": 274,  
      "probable_diisolasi": 0,  
      "probable_discarded": 0,  
      "CONFIRMATION": 0,  
      "confirmation_sembuh": 0,  
      "confirmation_meninggal": 0,  
      "suspect_meninggal": 0,  
      "closecontact_meninggal": 0,  
      "probable_meninggal": 26  
    }  
  ]  
}
```

E. Project Steps

Config tools:

- 1. Create Docker (MySQL, Airflow and PostgreSQL) in your local computer
- 2. Create Database in MySQL and PostgreSQL

Get Data:

- 3. Create Connection from Airflow to MySQL and PostgreSQL
- 4. Create DDL in MySQL
- 5. Get data from Public API covid19 and load data to MySQL

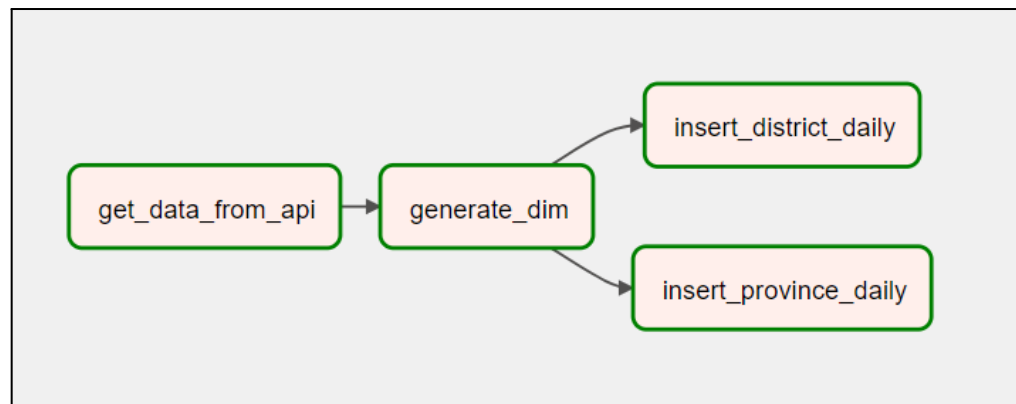
Transform and Load:

- 6. Create DDL in PostgreSQL for Fact table and Dimension table
- 7. Create load data to Dimension table
- 8. Create script for aggregate Province Daily save to Province Daily Table
- 9. Create script for aggregate District Daily save to District Daily Table

Schedule

- 10. Create DAG with schedule daily basis with task:
 - a. get data from api
 - b. generate dimension
 - c. province daily
 - d. district daily

Flow dag:



Output Report

11. Show data with tools dashboard as a report (optional but point plus)

F. Table Specification

Dimension table

1. Province table
 - a. province_id
 - b. province_name
2. District table
 - a. district_id
 - b. province_id
 - c. district_name
3. Case table
 - a. Id
 - b. Status name (suspect, closecontact, probable, confirmation)
 - c. Status detail

Fact table

1. Province Daily Table
 - a. Id (auto generate)
 - b. province_id
 - c. case_id
 - d. date
 - e. total
2. District Daily Table
 - a. Id (auto generate)
 - b. district_id
 - c. case_id
 - d. date
 - e. total

Link notion:

<https://imminent-locust-045.notion.site/Session-33-Final-Project-32a1dad656ce4db199b409624962f651>

Catatan untuk sesi presentasi:

1. Gunakan PPT
2. Jelaskan flow project secara singkat
3. Jelaskan tools yang digunakan secara singkat
4. Tampilkan output dari:
 - a. data di mysql - data from api
 - b. status scheduler airflow (success/failed)

- c. data di postgres - dimension table dan fact table
- 5. Tampilkan dashboard/report (jika ada)
- 6. List issue yang ditemukan (jika ada)