

Figure 1. ETL Architecture Diagram

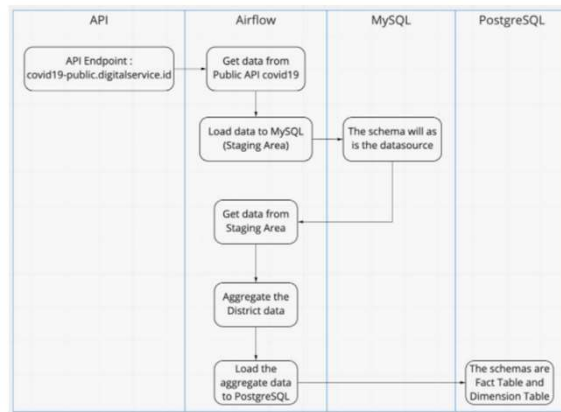


Figure 2. Integration Design Diagram



Figure 3. Docker Image

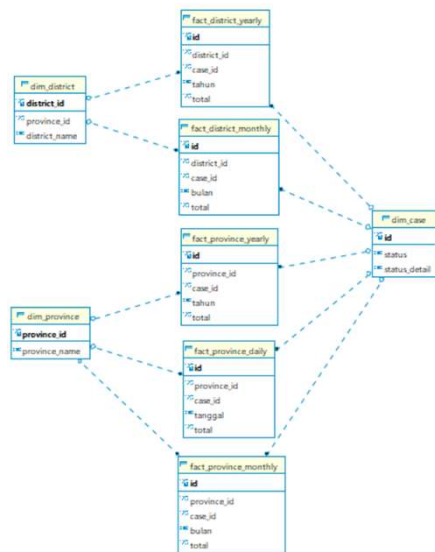


Figure 4. ERD Diagram

```

dag_id = 'final_project_schedule'
schedule_interval = '@ * * * * *'
start_date = days_ago(1)
catchup = False

@dag
def get_url():
    get_url = PythonOperator(
        task_id='get_url',
        python_callable=get_data,
        op_kwargs={'url_data': 'https://covid19-public.digitalservice.id/api/v1/rekapitulasi_v2/jabar/barian?level=kab'}
    )

    insert_to_staging_area = PythonOperator(
        task_id='insert_to_staging_area',
        python_callable=insert_into_table,
        op_kwargs={'eng_mysql': 'setting_mysql'}
    )

    migrate_data = PythonOperator(
        task_id='migrate_data',
        python_callable=migrate,
        op_kwargs={'eng_mysql': 'setting_mysql', 'eng_postgre': 'setting_postgre'}
    )

    create_table = PostgresOperator(
        task_id='create_table',
        postgres_conn_id='con_data_warehouse',
        sql='sql_file/create_table.sql'
    )

    temp_table = PythonOperator(
        task_id='temp_table',
        python_callable=temp_table_case,
        op_kwargs={'eng_postgre': 'setting_postgre'}
    )

    insert_table_dim = PostgresOperator(
        task_id='insert_table_dim',
        postgres_conn_id='con_data_warehouse',
        sql='sql_file/data_table_dim.sql'
    )

    insert_table_fact = PostgresOperator(

```

Figure 5. DAG Airflow

PostgresOperator PythonOperator

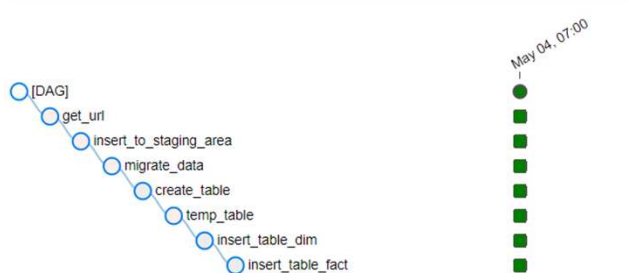


Figure 6. Tree View Result Airflow

	id	123	province_id	123	case_id	123	tahun	123	total
1	1		32		1		2020		35,463
2	2		32		2		2020		100,562
3	3		32		3		2020		3,125
4	4		32		4		2020		55,133
5	5		32		5		2020		162,890
6	6		32		6		2020		0
7	7		32		7		2020		15,039
8	8		32		8		2020		97,644
9	9		32		9		2020		106,694
10	10		32		10		2020		68,645
11	11		32		11		2020		1,163
12	12		32		1		2021		23,972
13	13		32		2		2021		107,756

Figure 7. One of the Fact Table

PROJECT EXPLANATION

The goals of this project is make tables in data warehouse that contain total of spread covid19 at Jawa Barat, the Data took API public. This project using docker as virtual machine for built airflow as scheduler, MySQL as staging area, and PostgreSQL as data warehouse. First step, built image at docker that contain airflow, MySQL, and PostgreSQL. Then, create DAG for know flow from API to fact table. In this project, the flow are getting url then save to staging area. After that, migrate (load) data from staging area to warehouse and make a dimensional table. Then, aggregate data and insert to fact table. The result can be seen in figure number 7.