NOVEMBER 28, 2023

Final Project
Data Engineer 15

Ricky Febrian

Ricky Febrian

https://www.linkedin.com/in/ricky-febrian-oce/

CASE

CASE

GOALS

Mengolah data dari PIKOBAR (Pusat Informasi & Koordinasi Covid-19 Jawa Barat) agar kemudian dapat di proses oleh tim DA atau DS

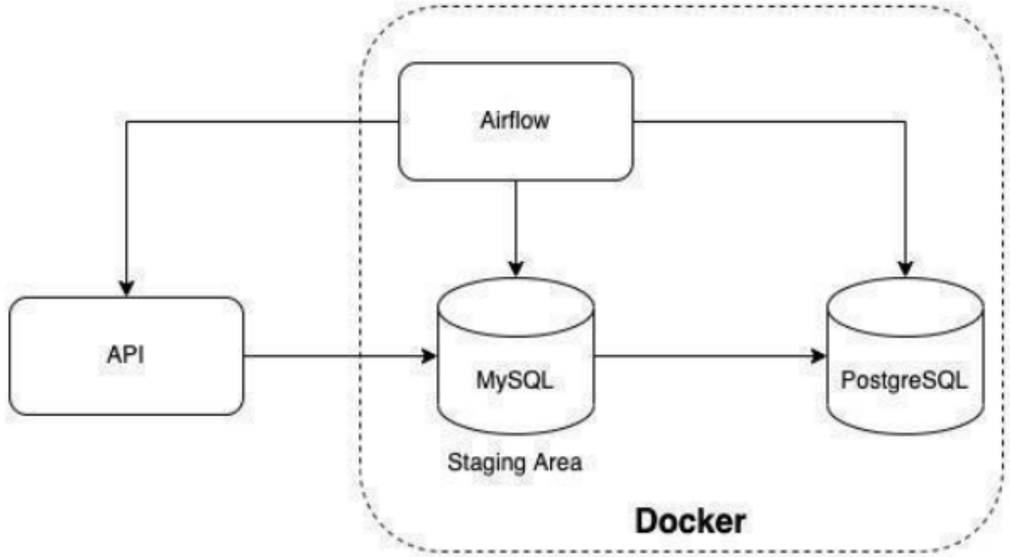
Contoh Data

Berikut contoh data PIKOBAR

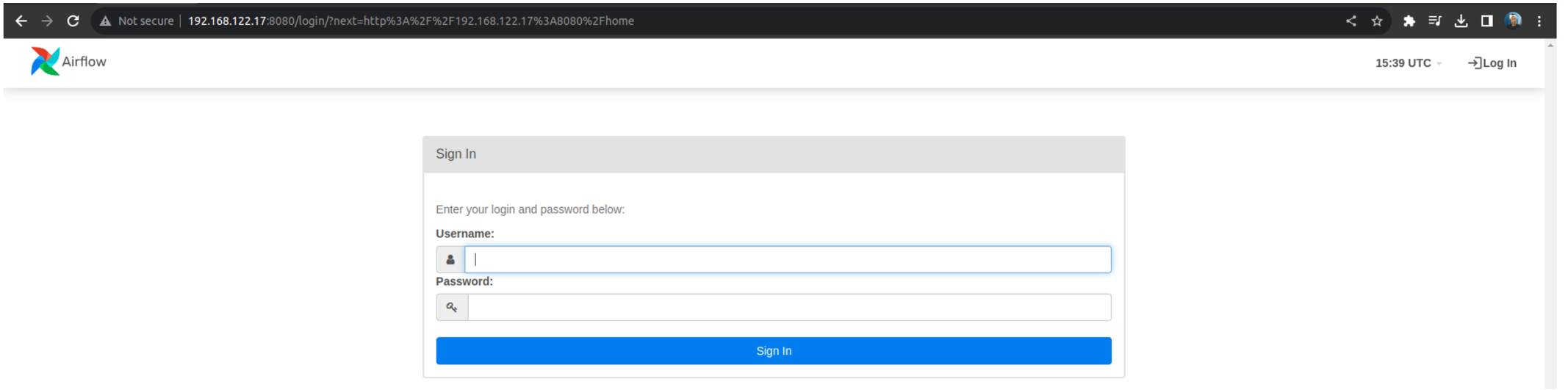
✨ Steps

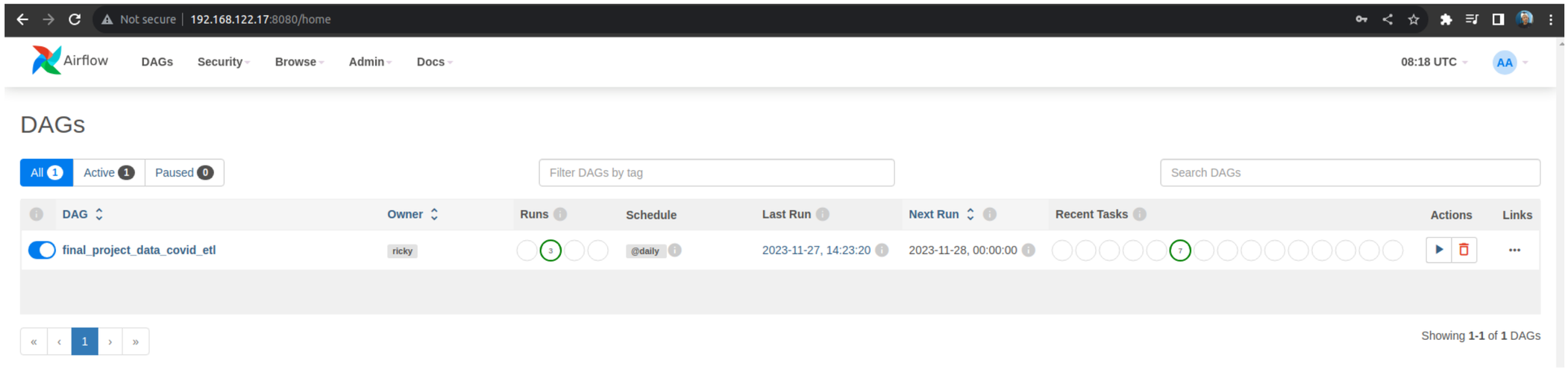
Rincian proses yg dilakukan pada airflow

1. Extract Data From Public API (PIKOBAR)
    a. Extract Data
2. Transform dan Load ke Staging Area
    a. Read Data From API
    b. Write Data to MySQL
3. Transform dan Load ke Production Area
    a. Insert Data to dim_province (postgresql)
    b. Insert Data to dim_district (postgresql)
    c. Insert Data to dim_case (postgresql)
    d. Insert Data to fact_district_daily (postgresql)
    e. Insert Data to fact_province_daily (postgresql)

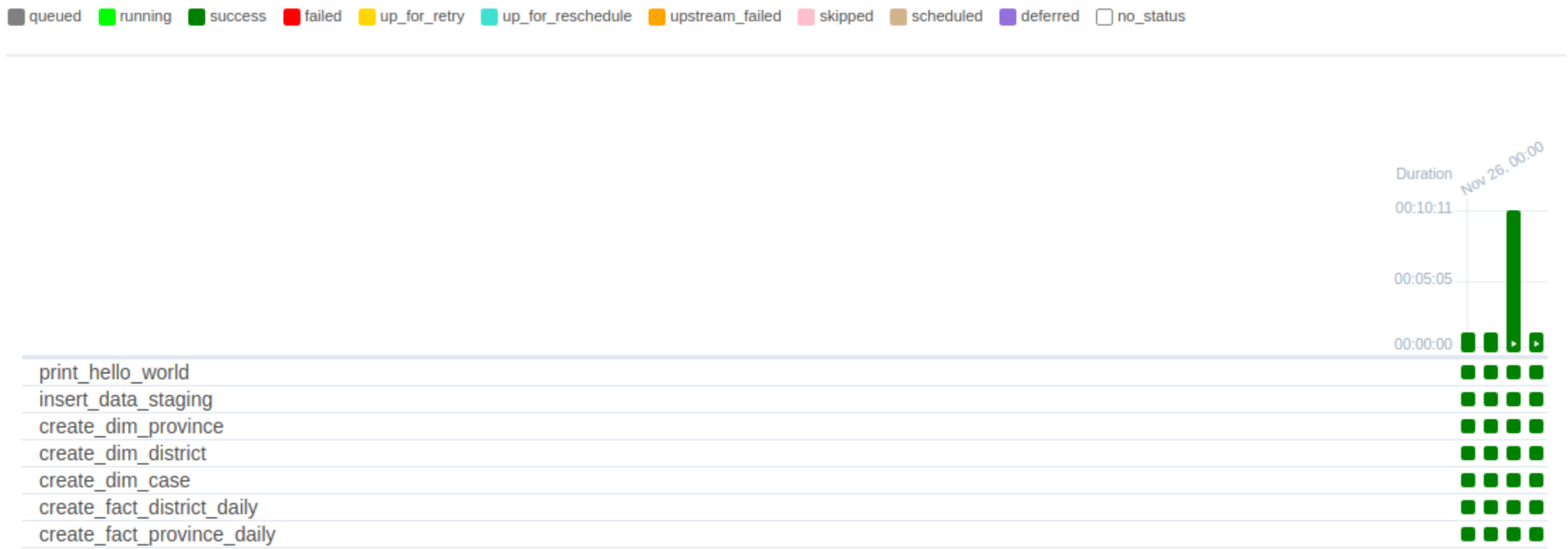


Airflow


Not secure | 192.168.122.17:8080/login/?next=http%3A%2F%2F192.168.122.17%3A8080%2Fhome
Airflow
15:39 UTC
Log In
Sign In
Enter your login and password below:
Username:
Password:
Sign In


Not secure | 192.168.122.17:8080/home
Airflow
DAGs
Security
Browse
Admin
Docs
08:18 UTC
AA
DAGs
All 1
Active 1
Paused 0
Filter DAGs by tag
Search DAGs
DAG
Owner
Runs
Schedule
Last Run
Next Run
Recent Tasks
Actions
Links
final_project_data_covid_etl
ricky
@daily
2023-11-27, 14:23:20
2023-11-28, 00:00:00
Showing 1-1 of 1 DAGs

Airflow



Global Variable

```
api_url = "http://103.150.197.96:5005/api/v1/rekapitulasi_v2/jabar/harian?level=kab"
staging_db_constring = "mysql://root:mysql@192.168.122.17:3307/finalproject"
dwh_constring = "postgresql://postgres:postgres@192.168.122.17:5435/dwh"
```

✨ Airflow

DigitalSkola

Extract Data

Staging Area

Production Area

Input

```python
op_print_hello_world = BashOperator(
    task_id='print_hello_world',
    bash_command='echo "HelloWorld!"'
)
```

Output

```
[2023-11-27, 14:23:21 UTC] {subprocess.py:74} INFO - Running command: ['bash', '-c', 'echo "HelloWorld!"']
[2023-11-27, 14:23:21 UTC] {subprocess.py:85} INFO - Output:
[2023-11-27, 14:23:21 UTC] {subprocess.py:92} INFO - HelloWorld!
[2023-11-27, 14:23:21 UTC] {subprocess.py:96} INFO - Command exited with return code 0
```

Airflow

DigitalSkola

Extract Data	Staging Area	Production Area
Input		
Output	Output - 1	Output - 2

```python
def get_data():
    r = requests.get(api_url)
    content = r.json()["data"]["content"]
    with open("dags/covid_data.json", "w") as f:
        json.dump(content, f)
    return content


def insert_data_staging():
    content = get_data()
    df = pd.DataFrame(content)
    df.to_sql("covid_jabar", staging_db_constring, if_exists="replace", index=False)
```

Extract Data &
Staging Area

Airflow

DigitalSkola

Extract Data

Staging Area

Production Area

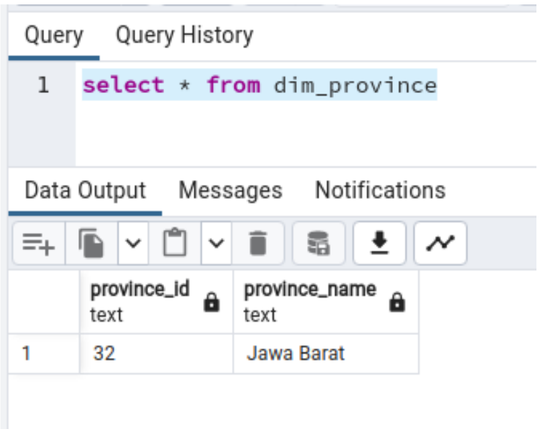
Input

```python
def create_dim_province():
    engine = create_engine(staging_db_constring)
    sql_query = pd.read_sql_query(
        '''select distinct kode_prov as province_id, nama_prov as province_name from covid_jabar''', engine)
    df = pd.DataFrame(sql_query, columns = ['province_id', 'province_name'])
    print(df.to_string(index=False))

    engine_postgres = create_engine(dwh_constring)
    df.to_sql('dim_province', engine_postgres, index=False, if_exists='replace')
```

Output

```
[2023-11-27, 07:13:21 UTC] {logging_mixin.py:115} INFO - province_id province_name
          32    Jawa Barat
          32    Jawa Barat
          32    Jawa Barat
          32    Jawa Barat
          32    Jawa Barat
          32    Jawa Barat
          32    Jawa Barat
          32    Jawa Barat
          32    Jawa Barat
          32    Jawa Barat
```



dim_province

Airflow

DigitalSkola

Extract Data

Staging Area

Production Area

Input

```python
def create_dim_district():
    engine = create_engine(staging_db_constring)
    sql_query = pd.read_sql_query(
        '''select kode_kab as district_id, kode_prov as province_id, nama_kab as district_name from covid_jabar''', engine)
    df = pd.DataFrame(sql_query, columns = ['district_id', 'province_id','district_name'])
    print(df.to_string(index=False))

    engine_postgres = create_engine(dwh_constring)
    df.to_sql('dim_district', engine_postgres, index=False, if_exists='replace')
```

Output

```
AIRFLOW_CTX_DAG_RUN_ID=scheduled__2023-11-26T00:00:00+00:00
[2023-11-27, 14:23:26 UTC] {logging_mixin.py:115} INFO - district_id province_id          district_name
       3204          32       Kabupaten Bandung
       3217          32 Kabupaten Bandung Barat
       3216          32        Kabupaten Bekasi
       3201          32         Kabupaten Bogor
       3207          32        Kabupaten Ciamis
       3203          32       Kabupaten Cianjur
       3209          32       Kabupaten Cirebon
       3205          32         Kabupaten Garut
       3212          32     Kabupaten Indramayu
```

Query | Query History

```
1  select * from dim_district
```

Data Output | Messages | Notifications

	district_id text	province_id text	district_name text
1	3204	32	Kabupaten Bandung
2	3217	32	Kabupaten Bandung Barat
3	3216	32	Kabupaten Bekasi
4	3201	32	Kabupaten Bogor
5	3207	32	Kabupaten Ciamis
6	3203	32	Kabupaten Cianjur

dim_district

Airflow

DigitalSkola

Extract Data

Staging Area

Production Area

Input

```python
def create_dim_case():
    engine = create_engine(staging_db_constring)
    sql_query = pd.read_sql_query(
        '''select closecontact_dikarantina, closecontact_discarded, closecontact_meninggal, confirmation_meninggal, confirmation_sembuh, probable_diisolasi, probab
    df = pd.DataFrame(sql_query, columns = ['closecontact_dikarantina', 'closecontact_discarded', 'closecontact_meninggal', 'confirmation_meninggal', 'confirmation
    print(df.to_string(index=False))

    df['id'] = range(1, len(df) + 1)

    print(df.to_string(index=False))

    df_melted = pd.melt(df, id_vars=["id"], value_vars=['closecontact_dikarantina', 'closecontact_discarded', 'closecontact_meninggal',
                                                        'confirmation_meninggal', 'confirmation_sembuh', 'probable_diisolasi', 'probable_discarded',
                                                        'probable_meninggal', 'suspect_diisolasi', 'suspect_discarded', 'suspect_meninggal'])


    df_melted[['id', 'status_name', 'status_detail']] = df_melted.apply(lambda row: pd.Series([row['id'], row['variable'].split('_')[0], row['variable'].split('_')

    df_melted['status'] = df_melted['status_name'] + '_' + df_melted['status_detail']

    df_melted = df_melted.drop(['variable', 'value'], axis=1)

    print(df_melted)

    engine_postgres = create_engine(dwh_constring)
    df_melted.to_sql('dim_case', engine_postgres, index=False, if_exists='replace')
```


dim_case

Airflow

DigitalSkola

Extract Data

Staging Area

Production Area

Output

```
AIRFLOW_CTX_DAG_RUN_ID=scheduled__2023-11-26T00:00:00+00:00
[2023-11-27, 14:23:27 UTC] {logging_mixin.py:115} INFO -  closecontact_dikarantina  closecontact_discarded  closecontact_meninggal  confirmation_meninggal  confirmation_sembuh  probable_diisolasi  probable_discarded  probable_meninggal  s
                     0                   274                       0                       0                       0                    0                  26                  31                2179                   0
                    72                   462                       0                       0                       0                    0                   7                   3                 773                   0
                   135                  1992                       0                       0                       0                    1                   0                  32                4001                1535                   0
                     0                     0                       0                       0                       0                    0                   0                 163                   0                   0                   0
                     3                  1292                       0                       0                       0                    0                   3                   0                2075                   0                   0
                     0                     0                       0                       0                       0                    0                   0                   0                   0                   0                   0
```

```
[2023-11-27, 14:23:27 UTC] {logging_mixin.py:115} INFO -        id   status_name status_detail                    status
0       1  closecontact   dikarantina  closecontact_dikarantina
1       2  closecontact   dikarantina  closecontact_dikarantina
2       3  closecontact   dikarantina  closecontact_dikarantina
3       4  closecontact   dikarantina  closecontact_dikarantina
4       5  closecontact   dikarantina  closecontact_dikarantina
...   ...           ...           ...                         ...
```

Query | Query History

```
1  select * from dim_case
```

Data Output | Messages | Notifications

	id bigint	status_name text	status_detail text	status text
1	1	closecontact	dikarantina	closecontact_dikarantina
2	2	closecontact	dikarantina	closecontact_dikarantina
3	3	closecontact	dikarantina	closecontact_dikarantina
4	4	closecontact	dikarantina	closecontact_dikarantina
5	5	closecontact	dikarantina	closecontact_dikarantina
6	6	closecontact	dikarantina	closecontact_dikarantina

dim_case

Airflow

DigitalSkola

Extract Data

Staging Area

Production Area

Input

```python
def create_fact_district_daily():
    engine = create_engine(staging_db_constring)
    sql_query = '''SELECT * FROM covid_jabar'''

    df = pd.read_sql_query(sql_query, engine)
    print(df.head())

    engine_postgres = create_engine(dwh_constring)
    df_pg = pd.read_sql_table('dim_case', engine_postgres)

    print(df_pg.head())

    df_piv = pd.DataFrame(df_pg)
    pivot_df = df_piv.pivot_table(index='status', aggfunc='size').reset_index(name='count')

    print(pivot_df)
```

Airflow

DigitalSkola

Extract Data

Staging Area

Production Area

Output

```
AIRFLOW_CTX_DAG_RUN_ID=scheduled__2023-11-26T00:00:00+00:00
[2023-11-27, 14:58:58 UTC] {logging_mixin.py:115} INFO -    CLOSECONTACT  CONFIRMATION  ...  suspect_meninggal     tanggal
0           274             0  ...                  0  2020-08-05
1           534             0  ...                  0  2020-08-05
2          2127             0  ...                  0  2020-08-05
3             0             0  ...                  0  2020-08-05
4          1295             0  ...                  0  2020-08-05

[5 rows x 20 columns]
[2023-11-27, 14:58:58 UTC] {logging_mixin.py:115} INFO -    id   status_name status_detail                    status
0   1  closecontact   dikarantina  closecontact_dikarantina
1   2  closecontact   dikarantina  closecontact_dikarantina
2   3  closecontact   dikarantina  closecontact_dikarantina
3   4  closecontact   dikarantina  closecontact_dikarantina
4   5  closecontact   dikarantina  closecontact_dikarantina
[2023-11-27, 14:58:58 UTC] {logging_mixin.py:115} INFO -                        status  count
0   closecontact_dikarantina    145
1     closecontact_discarded    145
2     closecontact_meninggal    145
3     confirmation_meninggal    145
4        confirmation_sembuh    145
5         probable_diisolasi    145
6         probable_discarded    145
7         probable_meninggal    145
8          suspect_diisolasi    145
9          suspect_discarded    145
10         suspect_meninggal    145
[2023-11-27, 14:58:58 UTC] {python.py:173} INFO - Done. Returned value was: None
```

✨ Airflow

DigitalSkola

Extract Data

Staging Area

Production Area

Input

```
def create_fact_province_daily():
    pass
```

Output

```
[2023-11-27, 14:58:59 UTC] {python.py:173} INFO - Done. Returned value was: None
```

Airflow

Extract Data

Staging Area

Production Area

```
with DAG(
    dag_id="final_project_data_covid_etl",
    default_args=default_args,
    start_date=datetime(2023, 11, 20),
    catchup=False,
    schedule_interval="@daily",
    ) as dag :

    op_print_hello_world = BashOperator(
        task_id='print_hello_world',
        bash_command='echo "HelloWorld!"'
    )
    insert_data_staging = PythonOperator(
        task_id="insert_data_staging",
        python_callable=insert_data_staging
    )
    create_dim_province = PythonOperator(
        task_id="create_dim_province",
        python_callable=create_dim_province
    )
    create_dim_district = PythonOperator(
        task_id="create_dim_district",
        python_callable=create_dim_district
    )
    create_dim_case = PythonOperator(
        task_id="create_dim_case",
        python_callable=create_dim_case
    )
    create_fact_district_daily = PythonOperator(
        task_id="create_fact_district_daily",
        python_callable=create_fact_district_daily
    )
    create_fact_province_daily = PythonOperator(
        task_id="create_fact_province_daily",
        python_callable=create_fact_province_daily
    )

op_print_hello_world >> insert_data_staging >> create_dim_province >> create_dim_district >> create_dim_case >> create_fact_district_daily >> create_fact_province
```

DAG

Extra

Kendala

1. Penulisan belum rapi, karena masih dalam development
2. Container Mysql tidak dapat diakses, sehingga perlu membuat database baru secara manual (exec docker container)
3. Terkadang error ketika menghapus file dag (reset semua dag), namun ketika di retry manual, dag berjalan lancar

Optimalisasi

1. Penggunaan Connection variable pada airflow dan Pembersihan file sampah
2. Update docker-compose
3. Kurangnya resource untuk airflow atau source code belum efisien
4. Update README.md github agar lebih mudah untuk dibaca dan dipahami sebagai portofolio

THANK YOU

DigitalSkola