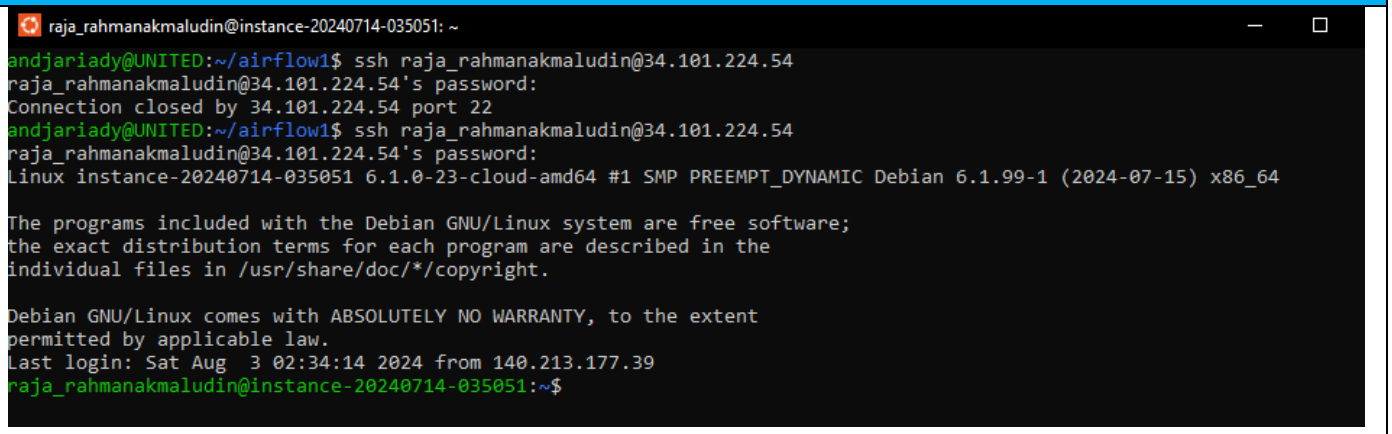
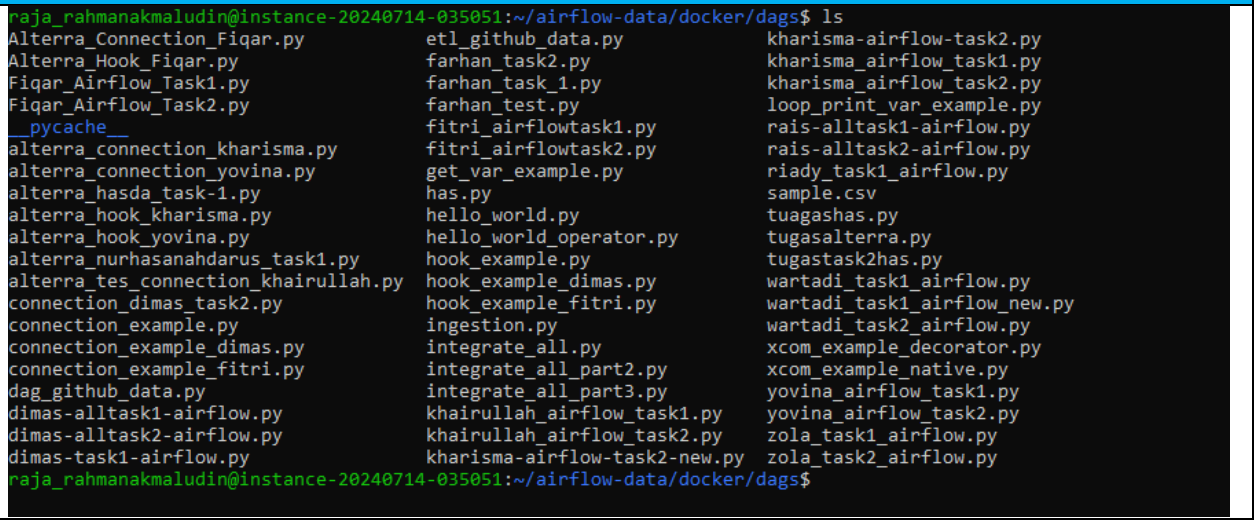
RIADY ANDJAR SAPUTRA

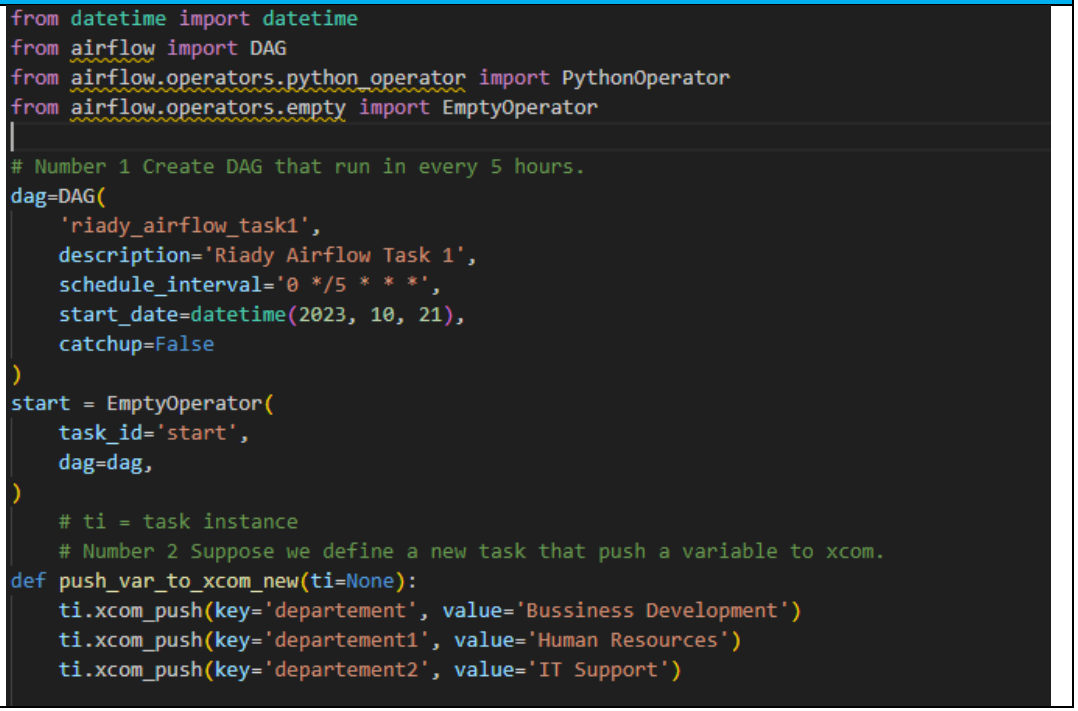
DE – 4

AIRFLOW TASK 1

No	Deskripsi	Perintah / Output
1.1	Pertama kita coba sambungakan computer local dengan ke alamat IP 34.101.224.54 dan memasukkan password untuk login sebagai **raja_rahmanakmaludi**.	

No	Deskripsi	Perintah / Output
1.2	Ketikkan perintah Linux **sudo nano riady_task1_airflow.py** untuk membuat dan mengedit isi file tersebut, kemudian file tersebut akan tersimpan ke dalam computer yang sudah kita sambungkan tadi. Tepatnya file tersebut disimpan di **raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker/dags**	

```
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker/dags$ ls
Alterra_Connection_Fiqar.py           etl_github_data.py              kharisma-airflow-task2.py
Alterra_Hook_Fiqar.py                 farhan_task2.py                 kharisma_airflow_task1.py
Fiqar_Airflow_Task1.py                farhan_task_1.py                kharisma_airflow_task2.py
Fiqar_Airflow_Task2.py                farhan_test.py                  loop_print_var_example.py
__pycache__                           fitri_airflowtask1.py           rais-alltask1-airflow.py
alterra_connection_kharisma.py        fitri_airflowtask2.py           rais-alltask2-airflow.py
alterra_connection_yovina.py          get_var_example.py              riady_task1_airflow.py
alterra_hasda_task-1.py               has.py                          sample.csv
alterra_hook_kharisma.py              hello_world.py                  tuagashas.py
alterra_hook_yovina.py                hello_world_operator.py         tugasalterra.py
alterra_nurhasanahdarus_task1.py      hook_example.py                 tugastask2has.py
alterra_tes_connection_khairullah.py  hook_example_dimas.py           wartadi_task1_airflow.py
connection_dimas_task2.py             hook_example_fitri.py           wartadi_task1_airflow_new.py
connection_example.py                 ingestion.py                    wartadi_task2_airflow.py
connection_example_dimas.py           integrate_all.py                xcom_example_decorator.py
connection_example_fitri.py           integrate_all_part2.py          xcom_example_native.py
dag_github_data.py                    integrate_all_part3.py          yovina_airflow_task1.py
dimas-alltask1-airflow.py             khairullah_airflow_task1.py     yovina_airflow_task2.py
dimas-alltask2-airflow.py             khairullah_airflow_task2.py     zola_task1_airflow.py
dimas-task1-airflow.py                kharisma-airflow-task2-new.py   zola_task2_airflow.py
raja_rahmanakmaludin@instance-20240714-035051:~/airflow-data/docker/dags$
```

No	Deskripsi	Perintah / Output
2.1	Ini adalah query yang akan membuat DAG berjalan setiap 5 jam, lalu menyimpan beberapa nilai ke XCom, kemudian nilai-nilai tersebut diambil dari XCom. Query ini menjadi trigger untuk task program yang akan berjalan pada Airflow. Terdapat 3 task dalam query ini: 1. Start 2. Push_var_to_xcom_new 3. Pull_multiple_values_new	

```python
from datetime import datetime
from airflow import DAG
from airflow.operators.python_operator import PythonOperator
from airflow.operators.empty import EmptyOperator

# Number 1 Create DAG that run in every 5 hours.
dag=DAG(
    'riady_airflow_task1',
    description='Riady Airflow Task 1',
    schedule_interval='0 */5 * * *',
    start_date=datetime(2023, 10, 21),
    catchup=False
)
start = EmptyOperator(
    task_id='start',
    dag=dag,
)
    # ti = task instance
    # Number 2 Suppose we define a new task that push a variable to xcom.
def push_var_to_xcom_new(ti=None):
    ti.xcom_push(key='departement', value='Bussiness Development')
    ti.xcom_push(key='departement1', value='Human Resources')
    ti.xcom_push(key='departement2', value='IT Support')
```

No	Deskripsi	Perintah / Output
2.2	Setelah itu kita buka http://34.101.224.54:8080/ untuk masuk ke local airflow. File yang berisi perintah tadi sudah masuk ke dalam Airflow tersebut dan sukses belum ada error yang terjadi.