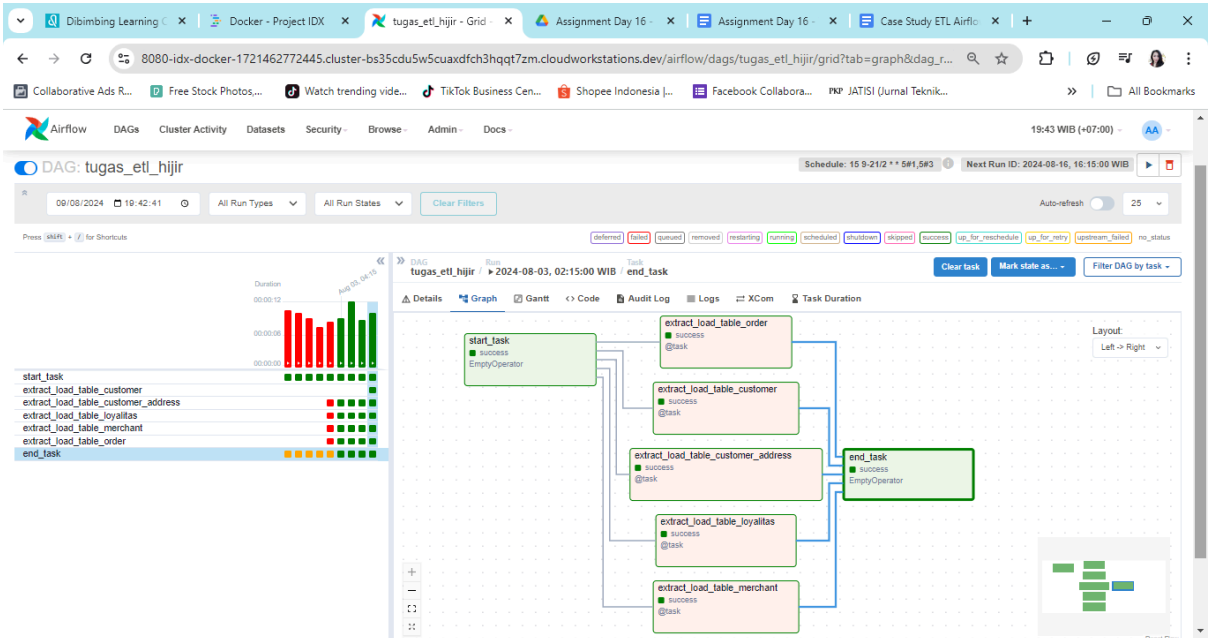
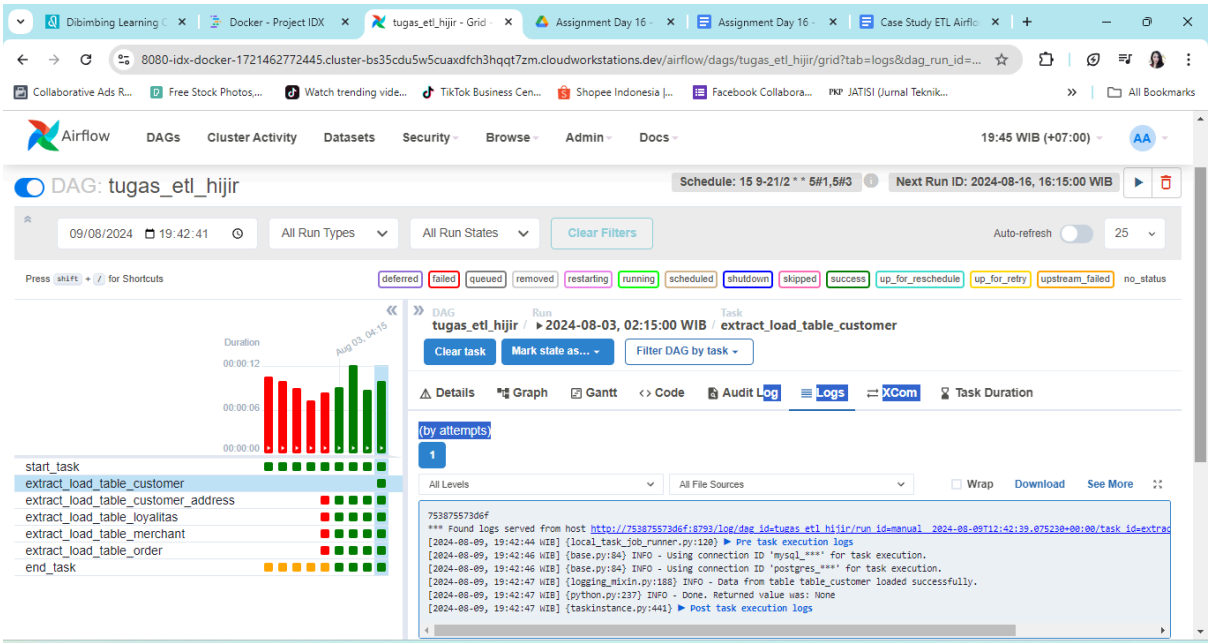
Testing DAGS
Hijir Della Wirasti

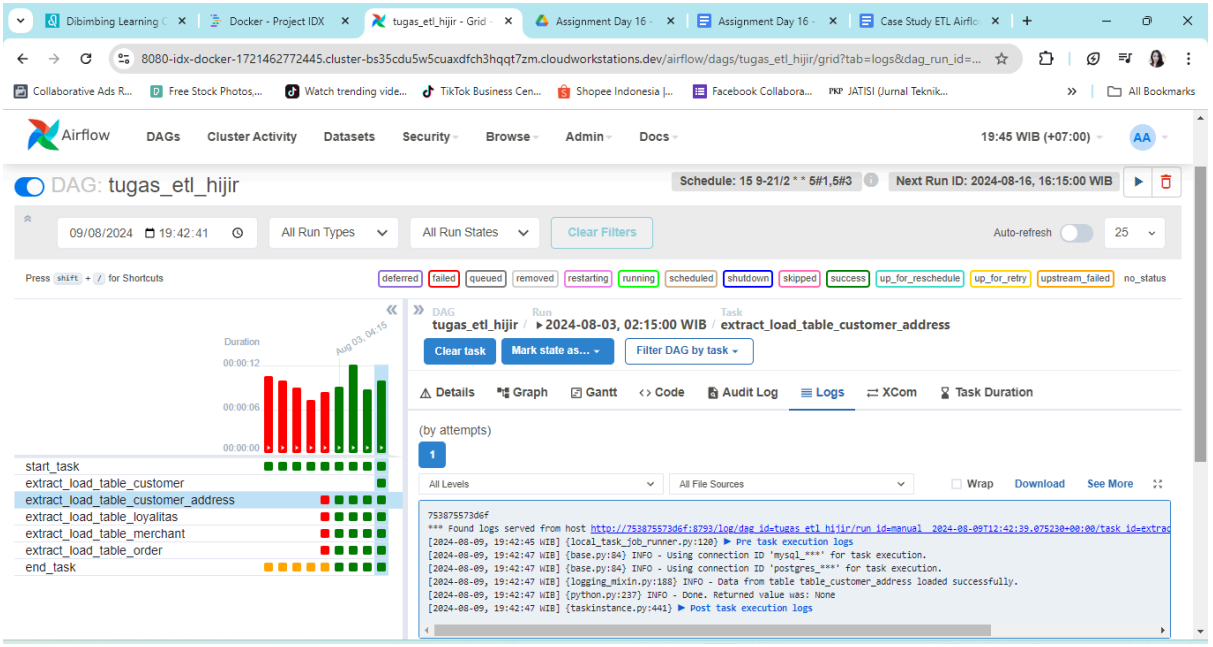
Current Graph



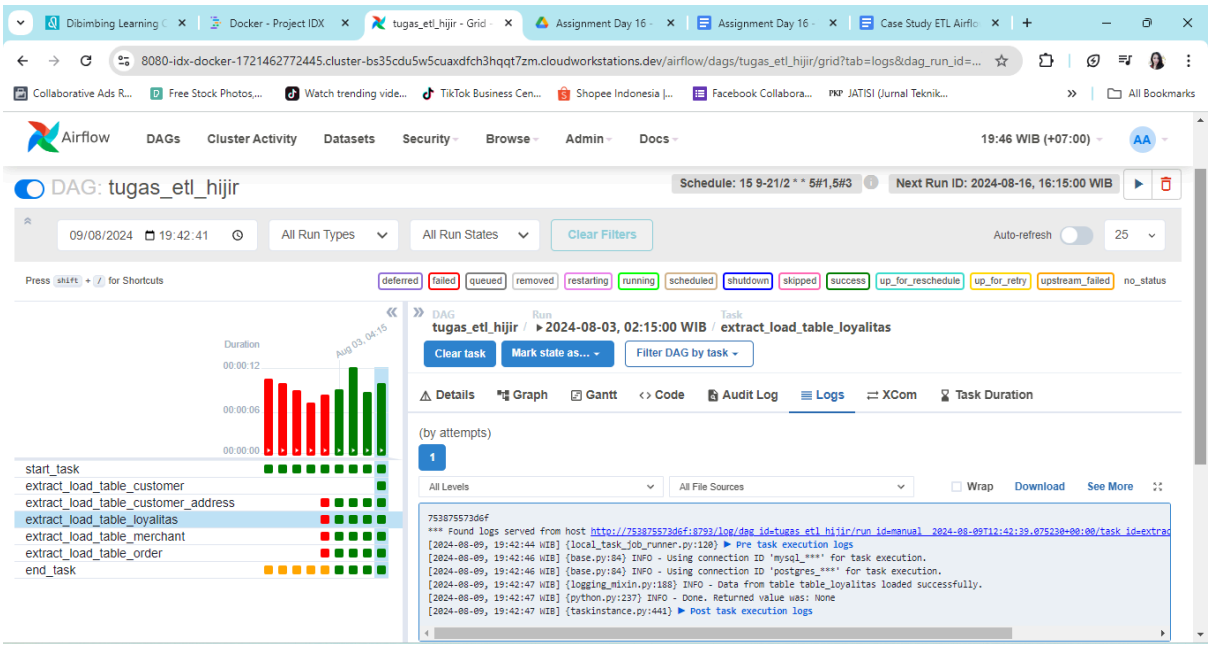
Extract 1 - table_customer



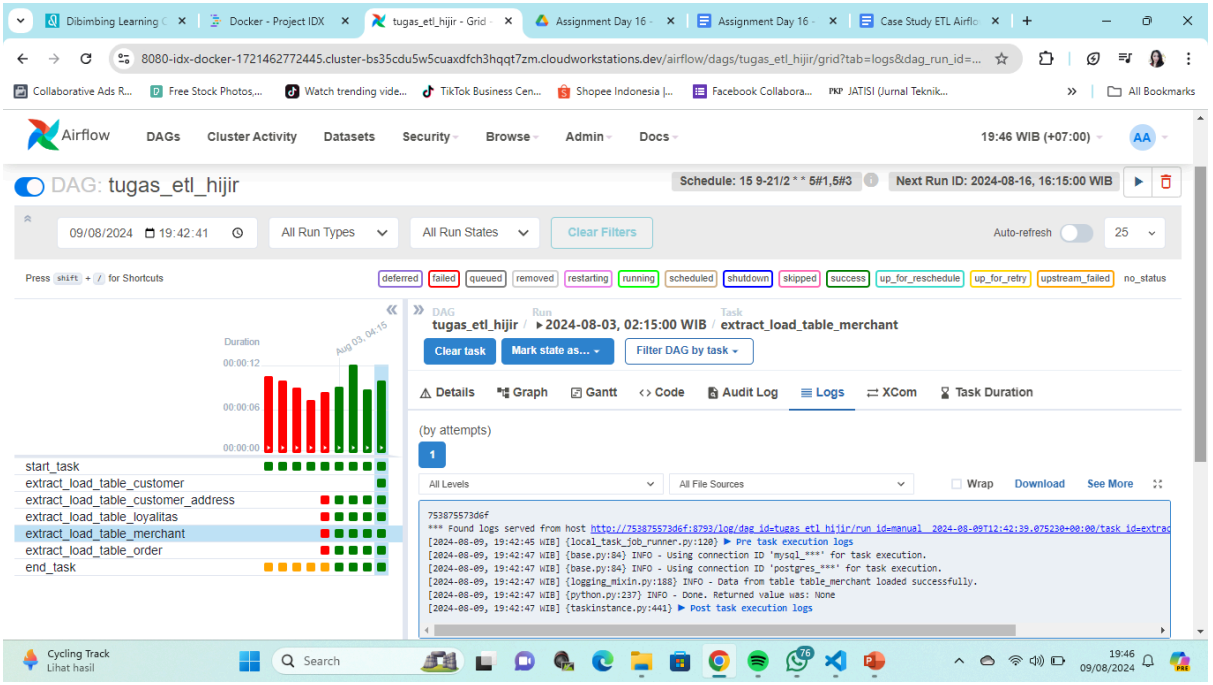
Extract 2 - table_customer_address



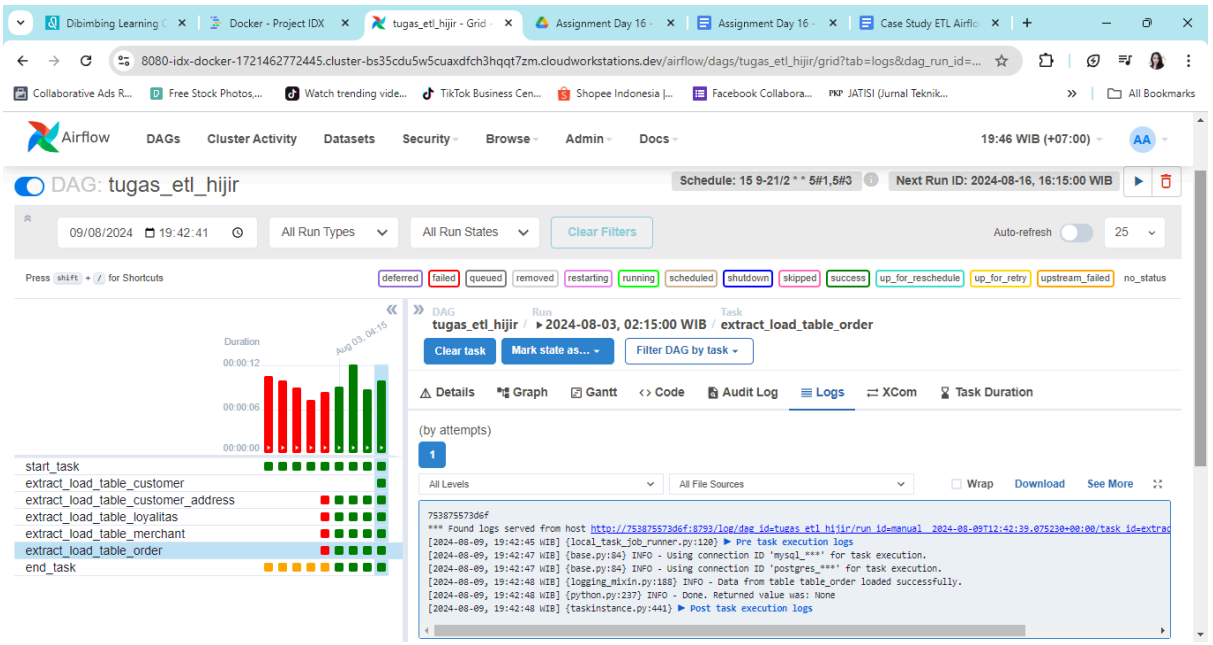
Extract 3 - table_loyalitas



Extract 4 table_merchant



Extract 5 table_order



CONFIG FILE

tugas_etl_hijir.py

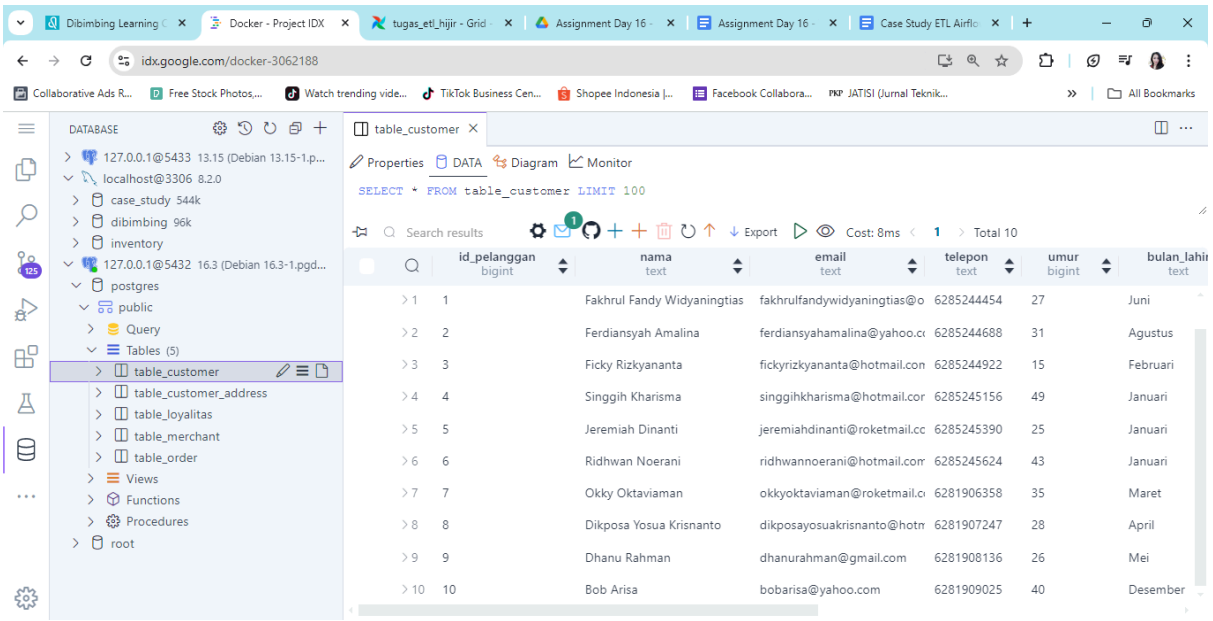
```
from airflow.decorators import dag, task
```

```
from airflow.operators.empty import EmptyOperator
from airflow.utils.dates import days_ago
from airflow.providers.mysql.hooks.mysql import MySqlHook
from airflow.providers.postgres.hooks.postgres import PostgresHook
import pandas as pd

# Daftar tabel yang akan diekstrak dari MySQL dan dimuat ke PostgreSQL
TABLES = ["table_customer", "table_customer_address",
"table_loyalitas", "table_merchant", "table_order"]

@dag(
    dag_id='tugas_etl_hijir',
    schedule_interval='15 9-21/2 * * 5#1,5#3',
    start_date=days_ago(1),
    catchup=False,
)
def tugas_etl_hijir():
    start_task = EmptyOperator(task_id="start_task")
    end_task = EmptyOperator(task_id="end_task")

    for table in TABLES:
        @task(task_id=f"extract_load_{table}")
        def extract_load(table):
            # Koneksi ke MySQL
            mysql_hook = MySqlHook(
                mysql_conn_id="mysql_dibimbing"
            ).get_sqlalchemy_engine()

            # Koneksi ke PostgreSQL
            postgres_hook = PostgresHook(
                postgres_conn_id="postgres_dibimbing"
            ).get_sqlalchemy_engine()

            with mysql_hook.connect() as mysql_conn,
postgres_hook.connect() as postgres_conn:
                # Cek apakah tabel ada di MySQL
                try:
                    # Ekstrak data dari MySQL
                    df = pd.read_sql_table(table, mysql_conn)
                    # Muat data ke PostgreSQL
                    df.to_sql(table, postgres_conn,
if_exists="replace", index=False)
```

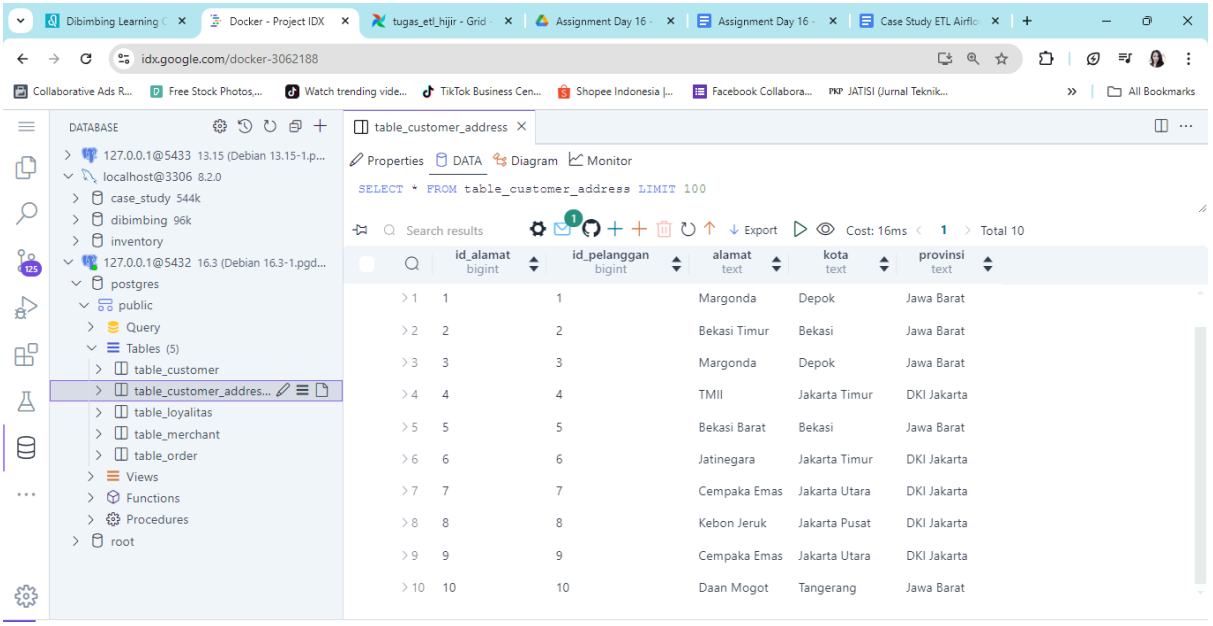
```
                    print(f"Data from table {table} loaded
successfully.")
                except ValueError as e:
                    print(f"Error: {e}. Table {table} not found in
MySQL.")

        # Hubungkan task mulai, proses ekstraksi dan pemuatan, serta
task selesai
        start_task >> extract_load(table) >> end_task

# Buat instance dari DAG
dag_instance = tugas_etl_hijir()
```

Hasil 5 Tabel di Postgresql

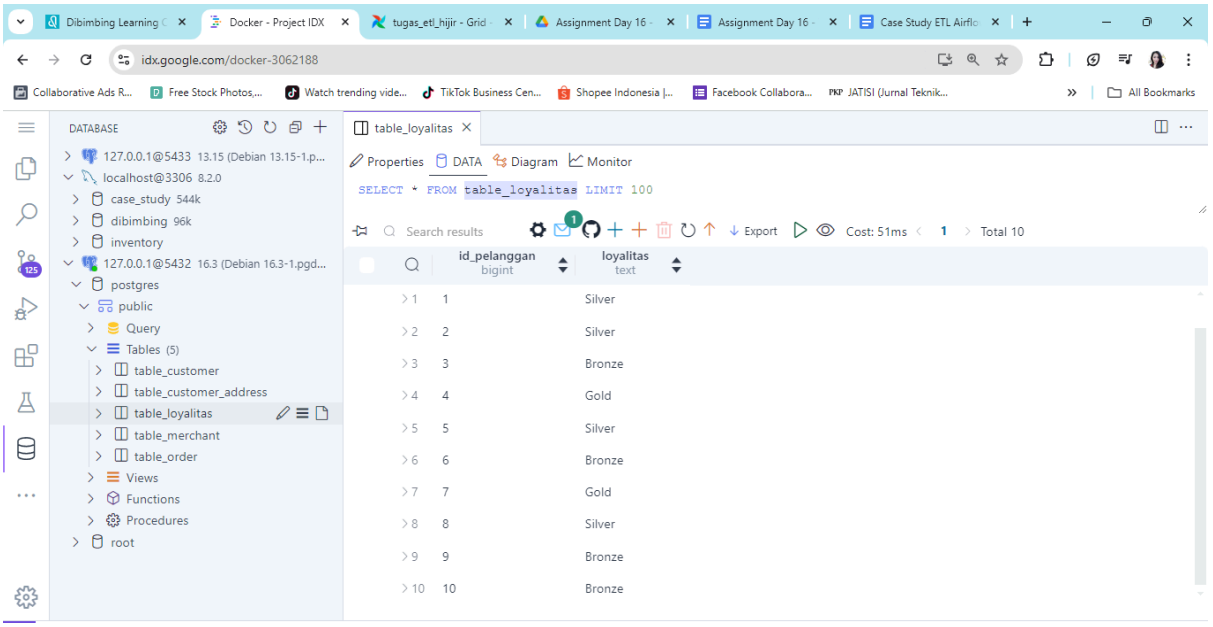
table_customer



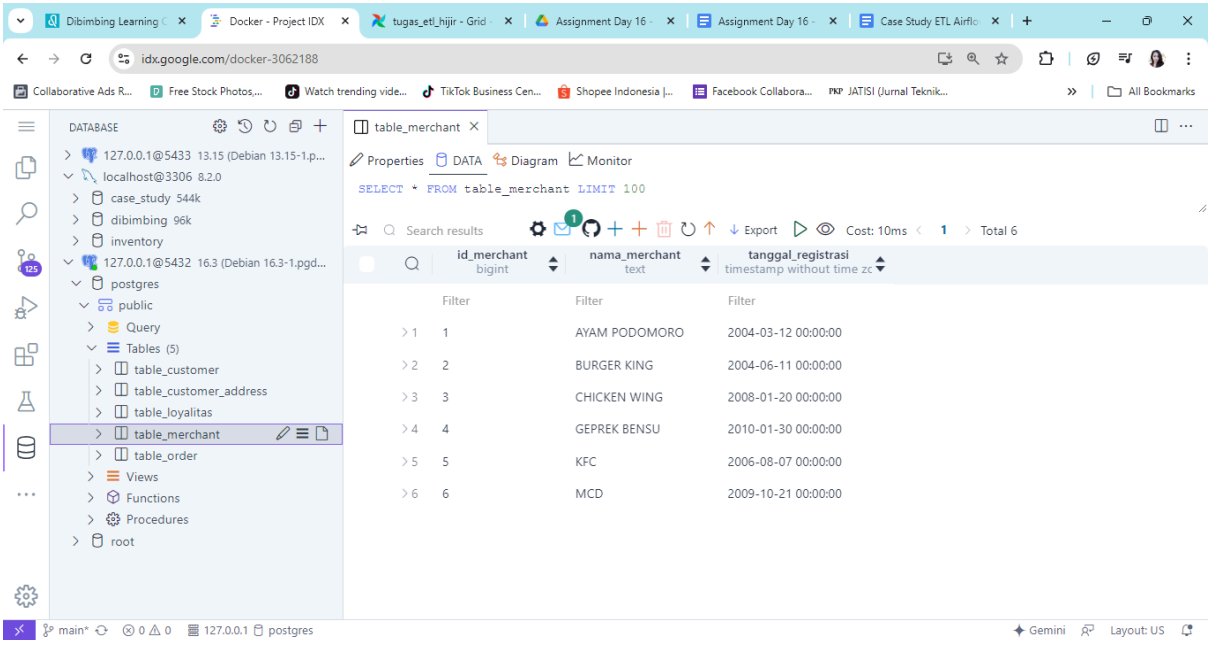
table_customer_address



table_loyalitas



table_merchant



table_order