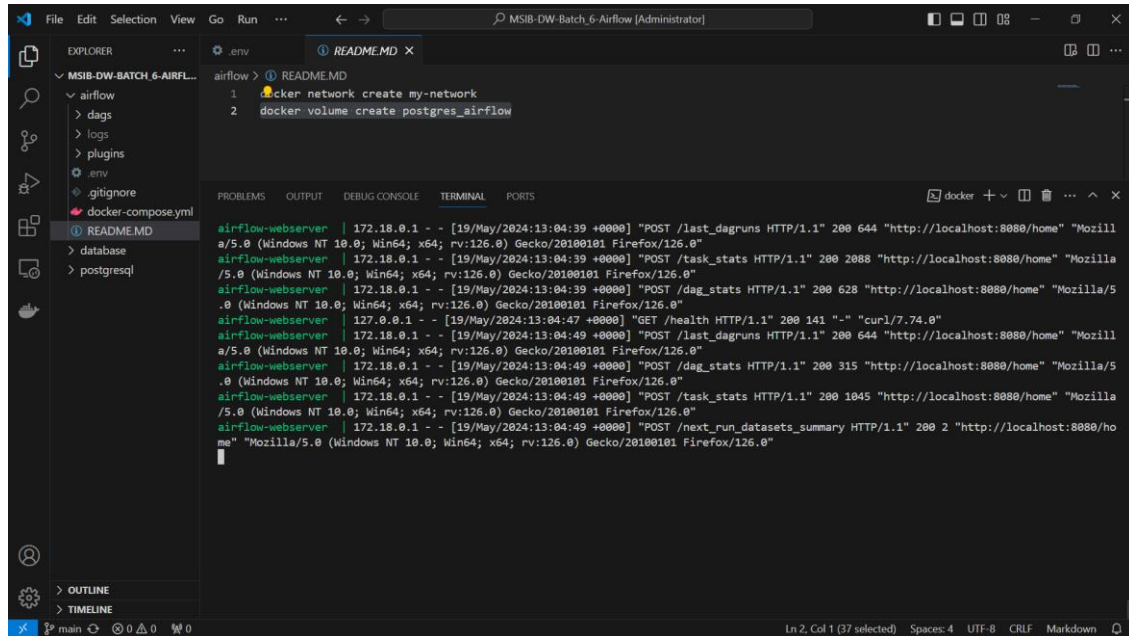


Airflow Connection & DAGs Running

1. (10 point) Jalankan docker apache airflow kemudian lampirkan bukti:
 - a. Screenshot terminal



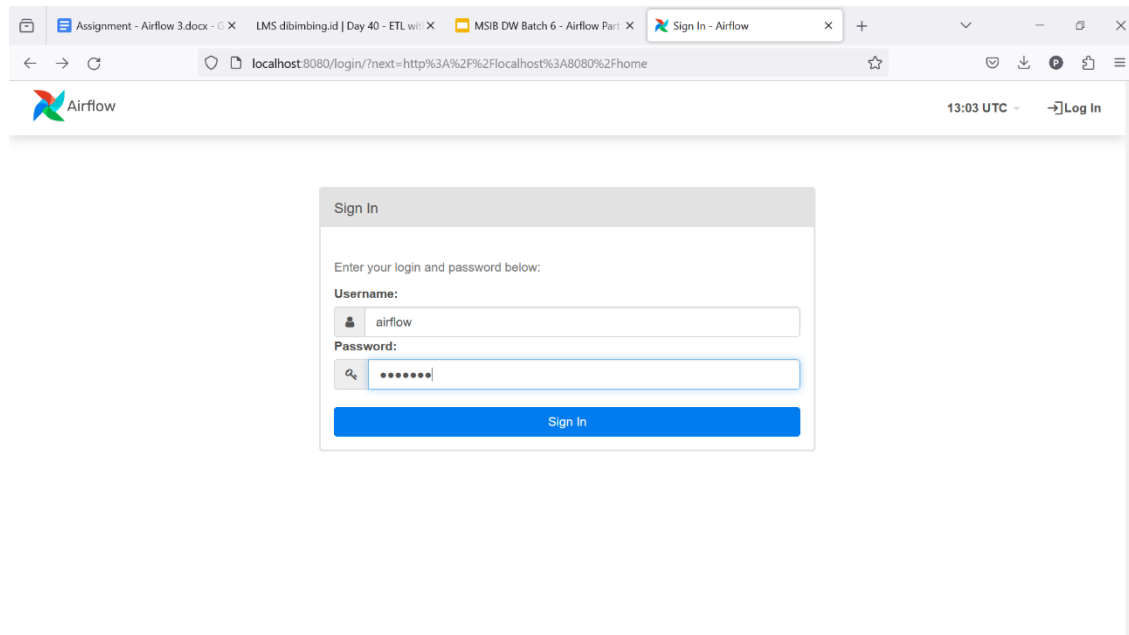
The screenshot shows a terminal window with the following content:

```
airflow > docker network create my-network
airflow > docker volume create postgres_airflow
```

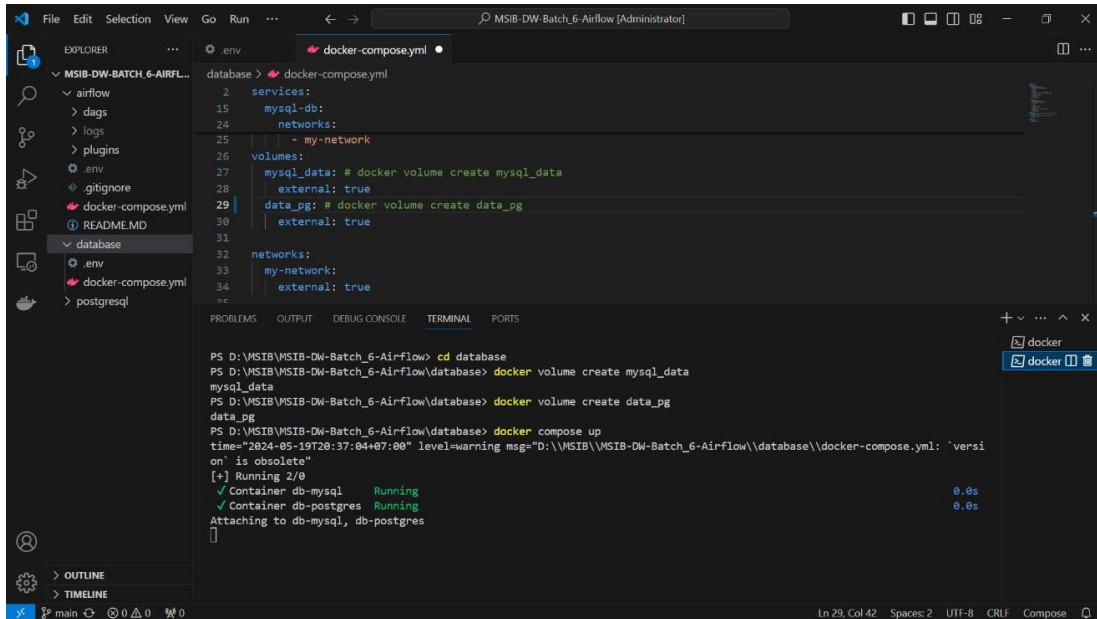
The terminal also displays logs for the Airflow webserver, showing HTTP requests and responses:

```
airflow-webserver | 172.18.0.1 - - [19/May/2024:13:04:39 +0000] "POST /last_dagruns HTTP/1.1" 200 644 "http://localhost:8080/home" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:126.0) Gecko/20100101 Firefox/126.0"
airflow-webserver | 172.18.0.1 - - [19/May/2024:13:04:39 +0000] "POST /task_stats HTTP/1.1" 200 2088 "http://localhost:8080/home" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:126.0) Gecko/20100101 Firefox/126.0"
airflow-webserver | 172.18.0.1 - - [19/May/2024:13:04:39 +0000] "POST /dag_stats HTTP/1.1" 200 628 "http://localhost:8080/home" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:126.0) Gecko/20100101 Firefox/126.0"
airflow-webserver | 127.0.0.1 - - [19/May/2024:13:04:47 +0000] "GET /health HTTP/1.1" 200 141 "-" "curl/7.74.0"
airflow-webserver | 172.18.0.1 - - [19/May/2024:13:04:49 +0000] "POST /last_dagruns HTTP/1.1" 200 644 "http://localhost:8080/home" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:126.0) Gecko/20100101 Firefox/126.0"
airflow-webserver | 172.18.0.1 - - [19/May/2024:13:04:49 +0000] "POST /dag_stats HTTP/1.1" 200 315 "http://localhost:8080/home" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:126.0) Gecko/20100101 Firefox/126.0"
airflow-webserver | 172.18.0.1 - - [19/May/2024:13:04:49 +0000] "POST /task_stats HTTP/1.1" 200 1045 "http://localhost:8080/home" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:126.0) Gecko/20100101 Firefox/126.0"
airflow-webserver | 172.18.0.1 - - [19/May/2024:13:04:49 +0000] "POST /next_run_datasets_summary HTTP/1.1" 200 2 "http://localhost:8080/home" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:126.0) Gecko/20100101 Firefox/126.0"
```

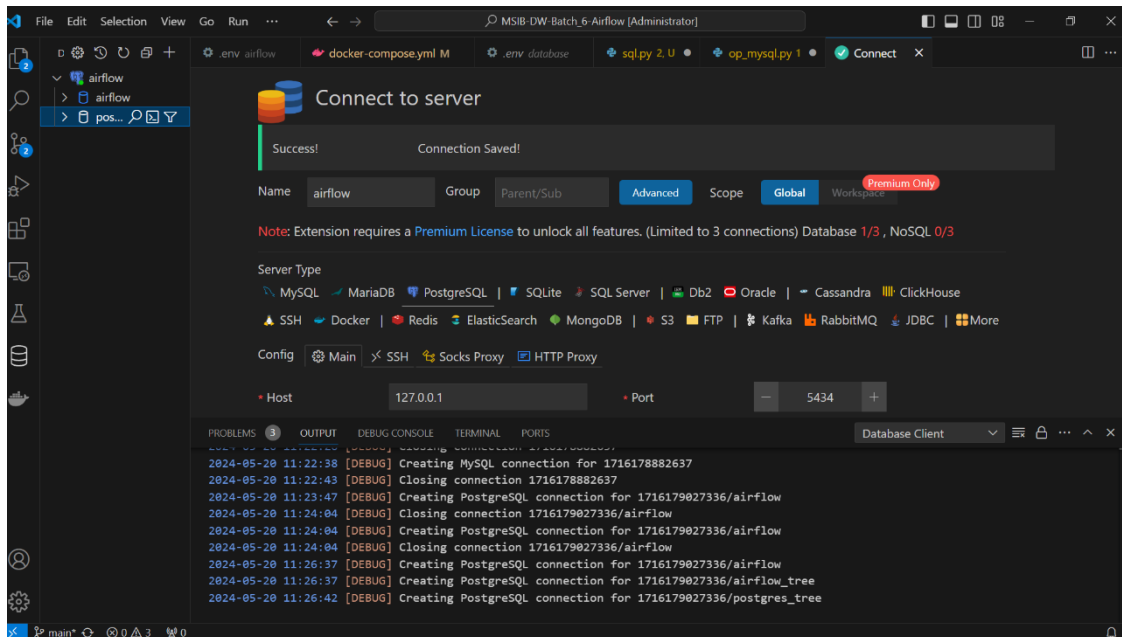
- b. Screenshot localhost:8080



2. (10 point) Jalankan docker database mysql dan postgresql kemudian lampirkan bukti:
- Screenshoot terminal



- Screenhoot Dbeaver / MySQL Connection Extension yang telah terhubung dengan postgresql dan mysql



3. (10 point) Buat connection MySQL dan PostgreSQL pada Apache Airflow kemudian
- a. Screenshoot UI hasil testing yang menunjukkan **connection successful**

The image displays two screenshots of the Apache Airflow web interface, both showing a 'Connection successfully tested' message at the top. The top screenshot shows the 'Edit Connection' form for a MySQL connection. The bottom screenshot shows the 'Add Connection' form for a PostgreSQL connection.

Top Screenshot: Edit Connection (MySQL)

- Connection Id: `mysql_conn`
- Connection Type: `MySQL`
- Description: (empty text area)
- Host: `db-mysql`
- Schema: (empty text area)
- Login: `root`
- Password: (masked with dots)
- Port: `3306`
- Extra: (empty text area)

Bottom Screenshot: Add Connection (PostgreSQL)

- Connection Id: `postgres_default`
- Connection Type: `Postgres`
- Description: (empty text area)
- Host: `db-postgres`
- Schema: `dash`
- Login: `postgres`
- Password: (masked with dots)
- Port: `5432`
- Extra: (empty text area)

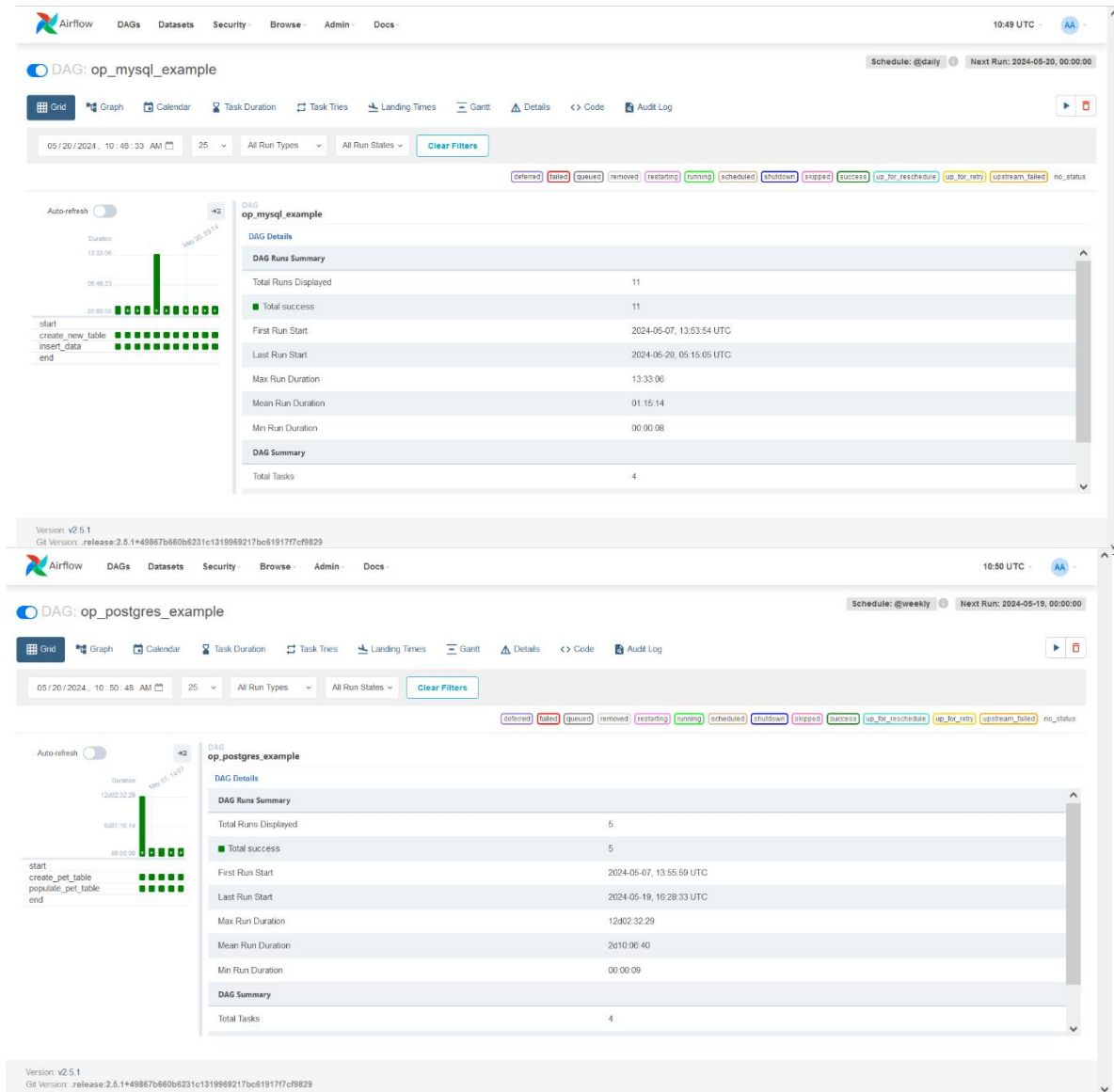
4. (10 point) Modifikasi DAGs `op_mysql` dan `op_postgresql` yang sudah ada dengan menambahkan EmptyOperator sehingga tampilan kedua DAG tersebut menjadi sebagai berikut:
- a. `op_mysql`



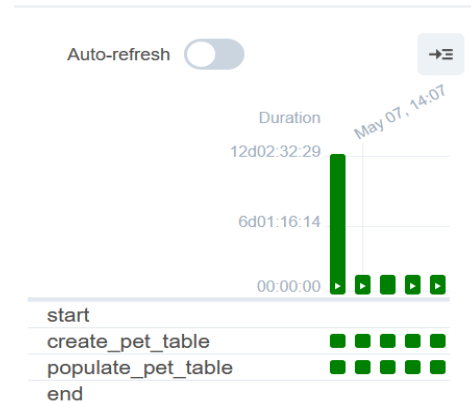
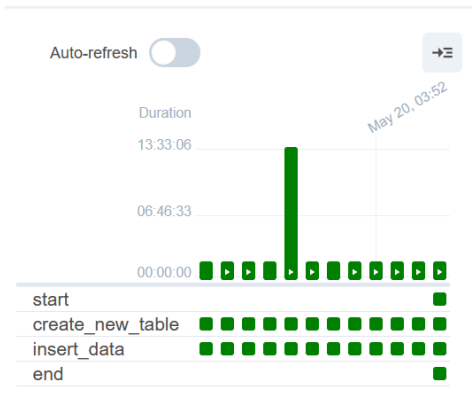
b. `op_postgresql`



5. (10 point) Jalankan DAGs `op_mysql` dan `op_postgresql` hingga `success` kemudian lampirkan hasil screenshotnya. Usahakan sampai pipeline berjalan `success` semua.



6. (10 point) Tampilkan hasil table yang dibuat oleh `op_mysql` dan `op_postgresql`



7. (30 point) Jelaskan apa yang dikerjakan oleh `op_mysql` dan `op_postgresql`? Apa perbedaan sebelum dan sesudah dijalankannya DAG? dan apa yang dapat di improve dari kedua DAG tersebut?

-`op_mysql`

Start =EmptyOperator, untuk menunjukan jika dimulai nya proses `op_mysql`.

Create_new_table, proses membuat table dengan Bernama "students" yang akan di simpan di MySQL.

Insert_data, dalam proses ini data di buat di dalam table yang bernama "students" di isi dengan nama "Alex Jones"

End EmptyOperator, di tahap ini merupakan proses akhir `op_mysql`.

-`Op_postgresql`

Start =EmptyOperator, untuk menunjukan jika dimulai nya proses `op_postgresql`

Create_pet_table, membuat tabel "pet" di dalam database PostgreSQL.

Populate_pet_table, proses dalam memasukkan data sampel ke dalam tabel "pet" menggunakan pernyataan SQL yang tersimpan dalam file `insert_pet.sql`. Data yang dimasukkan merupakan data dummy.

End EmptyOperator, di tahap ini merupakan proses akhir `op_postgresql`.

8. (10 point) Buat Kesimpulan mengenai Apache Airflow

Apache Airflow adalah alat yang digunakan untuk mengatur dan menjalankan tugas-tugas atau pekerjaan yang saling terhubung. Ini membantu kita membuat alur kerja yang kompleks, di mana satu tugas dapat bergantung pada tugas lainnya.

- Apache Airflow adalah platform open-source untuk otomatisasi, penjadwalan, dan pemantauan alur kerja (workflows) secara terprogram.
- Workflows yaitu Directed Acyclic Graph (DAG) dari tugas-tugas, memungkinkan alur kerja yang kompleks dan saling bergantung.
- Airflow dikembangkan pertama kali oleh Airbnb pada tahun 2014 dan saat ini dikelola oleh Apache Software Foundation.
- Airflow terdiri dari beberapa komponen utama:
 - Executor : Executor menjalankan tugas pada node pekerja yang didistribusikan di seluruh cluster.
 - Workers : Workers menjalankan tugas pada node pekerja yang didistribusikan di seluruh cluster.
 - Plugins : Plugins menyediakan fungsionalitas tambahan untuk Airflow, seperti operator dan sensor khusus.

Airflow benefits:

- Modular Architecture: Airflow memiliki arsitektur modular yang memungkinkan pengguna untuk mendefinisikan dan menggunakan kembali operator serta alur kerja. Modular architecture memudahkan pengembangan dan pengelolaan komponen-komponen dalam Airflow.
- Extensible: Arsitektur Airflow memungkinkan integrasi yang mudah dengan sistem eksternal. Airflow sangat dapat diperluas dan disesuaikan dengan kebutuhan pengguna.
- Scalable: Airflow dapat diskalakan secara horizontal untuk memenuhi tuntutan saluran pemrosesan data berskala besar. Kemampuan skalabilitas ini memungkinkan Airflow menangani beban kerja yang semakin meningkat.
- User-friendly UI: Airflow menyediakan antarmuka web yang ramah pengguna untuk memantau dan mengelola alur kerja, tugas, dan operator.
- Berbasis Python: Alur kerja Airflow didefinisikan menggunakan kode Python. Hal ini memudahkan pengembang untuk membuat dan menyesuaikan alur kerja sesuai kebutuhan.

Beberapa konsep Airflow:

- DAG (Directed Acyclic Graph): Ini adalah cara Airflow menggambarkan alur kerja, sebagai serangkaian tugas yang saling terhubung.
- Tugas (Task): Unit dasar dari alur kerja yang melakukan tindakan tertentu.
- Operator: Jenis tugas yang menentukan apa yang akan dilakukan.
- Sensor: Jenis tugas yang memantau kondisi tertentu dan memicu eksekusi alur kerja.

- Ketergantungan (Dependency): Hubungan antara tugas-tugas yang menentukan urutan eksekusi.

9. Kumpulkan hasil pengerjaan berupa file .pdf dan upload code hasil pengerjaan ke dalam github pribadi masing-masing.