

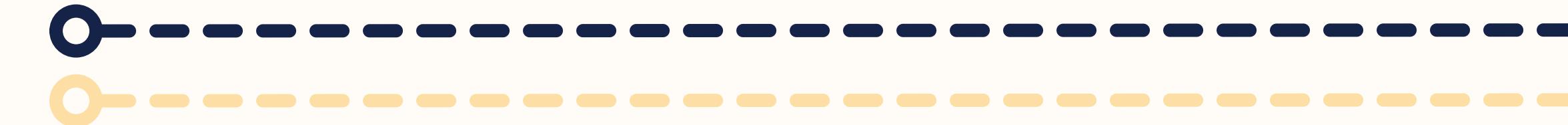
DATA ENGINEER UNTUK ENTRY LEVEL

----- **DigitalSkola** -----

FINAL PROJECT

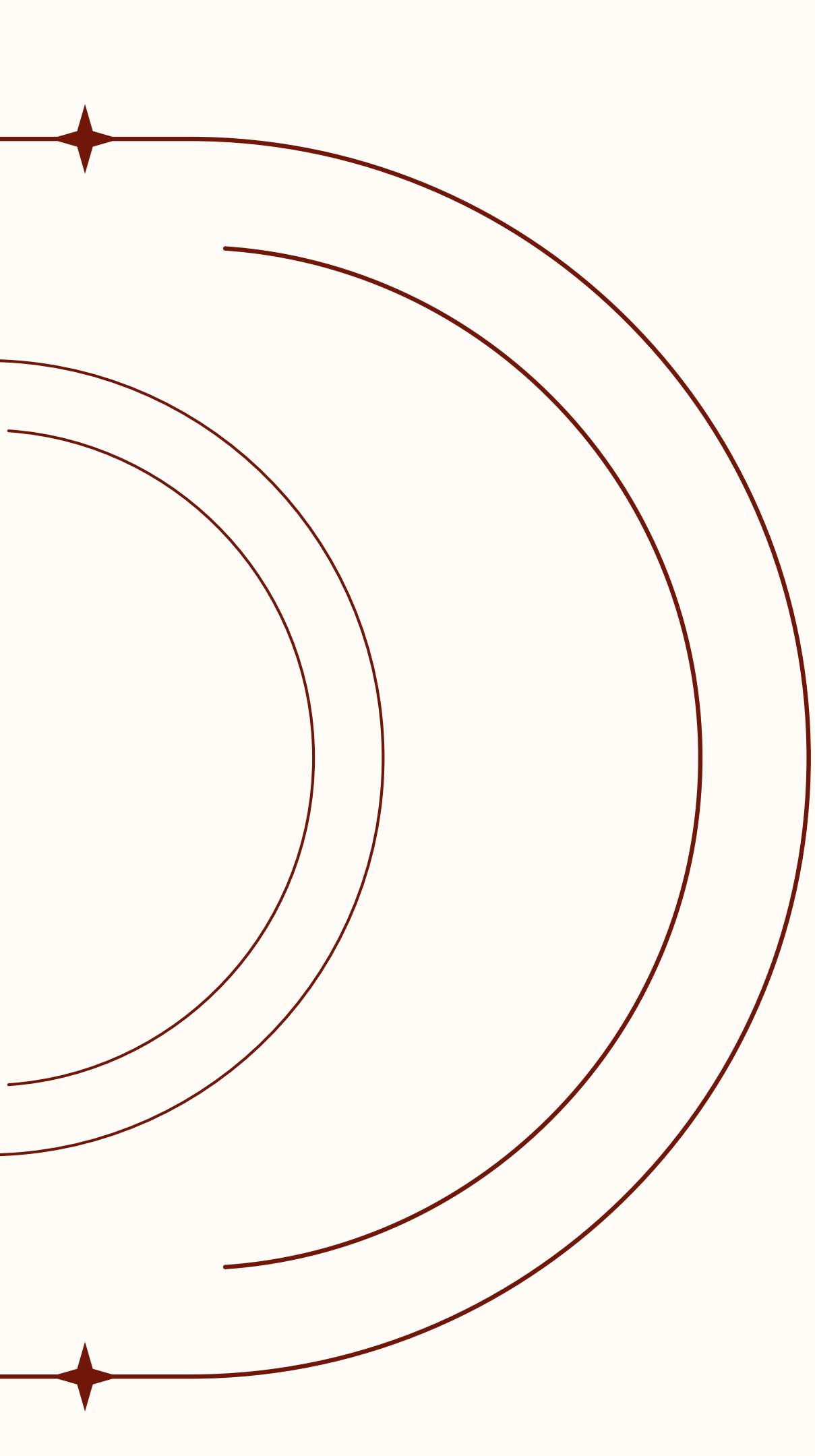
BY 8DREAM (KELOMPOK 1)

Data Realm Engineers And Maestros



- 01 Afroh Fauziah
- 02 Andi Rosilala
- 03 Althaf Nawadir
Taqiyyah
- 04 Andrew Fortino
Mahardika Suadnya

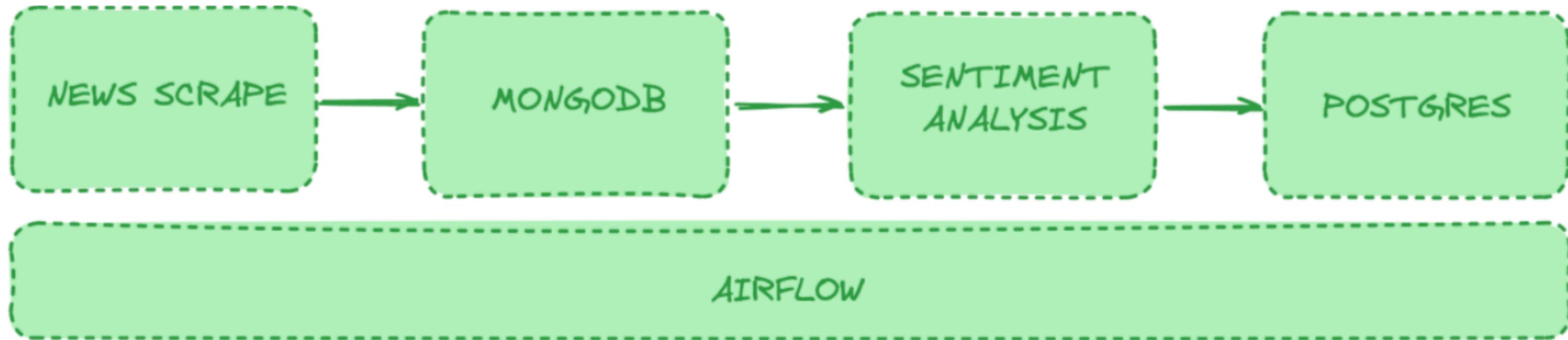




TUJUAN PROJEK

-
- Membangun pipeline otomatis untuk mengakuisisi data *news* dari API Finnhub.
 - Melakukan analisis sentimen terhadap data *news* yang telah dikumpulkan.
 - Menyimpan hasil analisis data *news* dalam database.

~ “CASE STUDY” ~

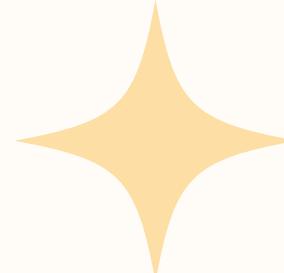


Membuat Batch Data Pipeline

Dimulai dari schedule menggunakan airflow dengan data source New Scrape lalu di load ke MongoDB Atlas. Dengan Sentiment Analysis kemudian di load ke Postgres hingga menjadi data warehouse.



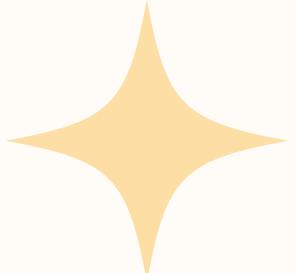
~ TOOLS ~



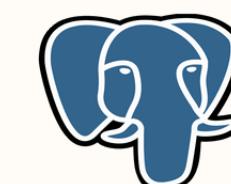
Finnhub



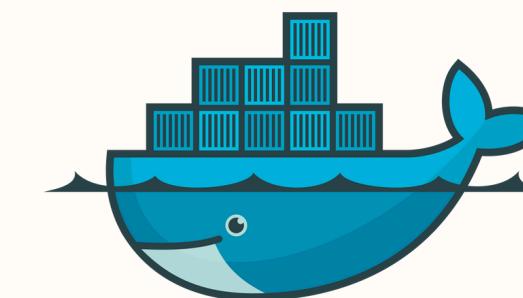
Apache
Airflow



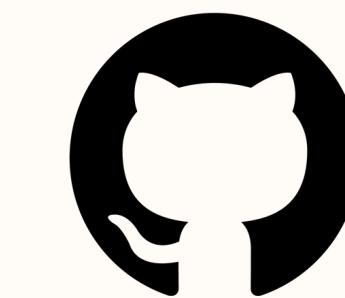
mongoDB®



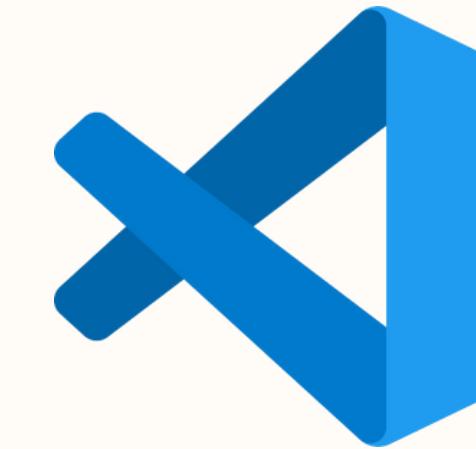
PostgreSQL



docker



GitHub





~ MONGODB ~

1. Sign Up / Login ke Finnhub '<https://filearnnnhub.io/docs/api/symbol-search>' dan juga MongoDB Atlas '<https://cloud.mongodb.com>'
2. Lalu masuk ke Network Access dan buat Ip Address baru.
3. Setelah itu buka database dan masuk ke collections learn mongo untuk create database. Juga dapat membuat database access user baru.

Add IP Access List Entry

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more.](#)

Access List Entry:

Comment:

This entry is temporary and will be deleted in

IP Address	Comment	Status	Actions
0.0.0.0/0 <small>(includes your current IP address)</small>	new_movies	Active	<input type="button" value="EDIT"/> <input type="button" value="DELETE"/>



Network Access

Create Database

Database name ?

news

Collection name ?

finnhub_news

Additional Preferences

Select



Database

The screenshot shows the MongoDB Compass interface. At the top, it says "news.finnhub_news". Below that, it displays storage information: "STORAGE SIZE: 4KB", "LOGICAL DATA SIZE: 0B", "TOTAL DOCUMENTS: 0", and "INDEXES TOTAL SIZE: 4KB". Underneath, there are tabs for "Find", "Indexes", "Schema Anti-Patterns", "Aggregation", and "Search Indexes". A blue link "Generate queries from natural language in Compass" is visible. At the bottom right is a button labeled "INSERT DOCUMENT".



Database Access

Password Authentication

temp_user

.....

Autogenerate Secure Password Copy

Database User Privileges

Configure role based access control by assigning privileges. A user will gain access to all actions wi
must choose at least one role or privilege. [Learn](#)

Built-in Role

Select one [built-in role](#) for this user.

Read and write to any database



Restrict Access to Specific Clusters/Federated Database Instances/Stream Processing Instances

Enable to specify the resources this user can access. By default, all resources in this project are accessible.

Grant Access To

The screenshot shows the "Grant Access To" section. It includes a search bar "Search user resources..." and a "Done" button. Under "Select all current Clusters", the option "learn-mongo" is checked. There is also an unchecked option "Select all current Federated Database Instances".



Temporary User

This user is temporary and will be deleted after your specified duration of 6 hours, 1 day, or 1 week.

Temporary User duration

6 hours

The screenshot shows the user details section. It lists the user "temp_user" with a SCRAM password, created "6 DAYS ago". The authentication method is "SCRAM". The role is "readWriteAnyDatabase@admin". At the bottom, it says "1 Cluster, 0 Federated Database Instances, 0 Stream Processing Instances". There are "EDIT" and "DELETE" buttons.



~ DOCKER-COMPOSE.YML ~



Membuat file docker-compose.yml untuk menjalankan docker dimana terdapat services postgres 12, airflow init, webserver, & scheduler.

```
# docker-compose.yml
 1  version: '3'
 2  x-airflow-common:
 3    &airflow-common
 4    image: apache/airflow:2.8.4-python3.10
 5    environment:
 6      - AIRFLOW__CORE__EXECUTOR=LocalExecutor
 7      - AIRFLOW__CORE__SQLALCHEMY_CONN=postgresql+psycopg2://airflow:airflow@postgres:5432/airflow
 8      - AIRFLOW__CORE__FERNET_KEY=FB0o_zt4e3Ziq3LdUU07F2Z95cvFFx16hU8jTeR1ASM=
 9      - AIRFLOW__CORE__LOAD_EXAMPLES=False
10      - AIRFLOW__CORE__LOGGING_LEVEL=INFO
11    volumes:
12      - ./dags:/opt/airflow/dags
13      - ./source:/opt/airflow/source
14      - ./output:/opt/airflow/output
15      - ./logs:/opt/airflow/logs
16      - ./plugins:/opt/airflow/plugins
17      - ./requirements.txt:/requirements.txt
18
19  services:
20    postgres:
21      image: postgres:12
22      environment:
23        - POSTGRES_USER=airflow
24        - POSTGRES_PASSWORD=airflow
25        - POSTGRES_DB=airflow
26        - POSTGRES_PORT=5432
27      volumes:
28        - postgres-db-volume:/var/lib/postgresql/data
```



```
29    healthcheck:
30      test: ["CMD", "pg_isready", "-U", "airflow"]
31      interval: 5s
32      retries: 5
33    ports:
34      - "5432:5432"
35
36    airflow-init:
37      << : *airflow-common
38      container_name: airflow_init
39      entrypoint: /bin/bash
40      command:
41        - -c
42        - airflow db init &&
43          airflow users create
44            --role Admin
45            --username airflow
46            --password airflow
47            --email airflow@airflow.com
48            --firstname airflow
49            --lastname airflow
50      restart: on-failure
51      depends_on:
52        postgres:
53          condition: service_healthy
```



```
55    airflow-webserver:
56      << : *airflow-common
57      command: >
58        bash -c "python3 -m pip install --upgrade pip &&
59          pip install --no-cache-dir -r /requirements.txt &&
60          airflow webserver"
61    ports:
62      - 8081:8080
63    container_name: airflow_webserver
64    depends_on:
65      airflow-init:
66        condition: service_completedSuccessfully
67    restart: always
68
69    airflow-scheduler:
70      << : *airflow-common
71      command: >
72        bash -c "python3 -m pip install --upgrade pip &&
73          pip install --no-cache-dir -r /requirements.txt &&
74          airflow scheduler"
75    container_name: airflow_scheduler
76    depends_on:
77      airflow-init:
78        condition: serviceCompletedSuccessfully
79    restart: always
80
81    volumes:
82      postgres-db-volume:
```

~ DOCKER-COMPOSE.YML ~



Untuk menjalankannya gunakan 'docker compose up' dan tunggu sampai semua berhasil running.



```
PS D:\SIB6\hands on\FINALPROJECT> docker-compose up --build
time="2024-06-24T13:05:50+07:00" level=warning msg="D:\\SIB6\\hands on\\FINALPROJECT\\docker-compose.yml: `version` is obsolete"
[+] Running 4/4
✓ airflow-scheduler Pulled
✓ airflow-webserver Pulled
✓ postgres Pulled
✓ airflow-init Pulled
[+] Running 2/6
✓ Network finalproject_default          Created
✓ Volume "finalproject_postgres-db-volume" Created
- Container finalproject-postgres-1      Created
- Container airflow_init                 Created
- Container airflow_webserver            Created
- Container airflow_scheduler            Created
Attaching to airflow_init, airflow_scheduler, airflow_webserver, postgres-1
postgres-1 | The files belonging to this database system will be owned by user "postgres".
```

Cek statusnya pada docker desktop / dengan 'docker ps' ataupun localhost webserver yang sesuai.



```
PS D:\SIB6\hands on\FINALPROJECT> docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
50ceee6f51fc        apache/airflow:2.8.4-python3.10   "/usr/bin/dumb-init ..."   9 minutes ago       Up 7 minutes
          0.0.0.0:8081->8080/tcp    airflow_webserver
d6e92dcabb5d        apache/airflow:2.8.4-python3.10   "/usr/bin/dumb-init ..."   11 minutes ago      Up 10 minutes
          8080/tcp                   airflow_scheduler
c0dd7a127ead        apache/airflow:2.8.4-python3.10   "/bin/bash -c 'airfl..."   11 minutes ago      Up 15 seconds
          8080/tcp                   airflow_init
54d19f78cf8d        postgres:12                      "docker-entrypoint.s..."  11 minutes ago      Up 11 minutes (healthy)
          0.0.0.0:5432->5432/tcp   finalproject-postgres-1
```

“REQUIREMENTS”

```
requirements.txt
1  pandas
2  pymongo
3  Flask-SQLAlchemy
4  textblob
5  finnhub-python
```

Tampilan saat docker berhasil running akan otomatis membuat volumes sebagaimana query.

FINALPROJECT

- > dags
- > env
- > logs
- > output
- > plugins
- > source
- 🐳 docker-compose.yml
- ≡ requirements.txt

```
plugins > ᐃ mongodb_loader.py > ...
1  import pymongo
2
3  def get_mongo_client(mongo_uri):
4      """Establish connection to the MongoDB."""
5      try:
6          client = pymongo.MongoClient(mongo_uri)
7          # Send a ping to confirm a successful connection
8          try:
9              client.admin.command('ping')
10             print("Pinged your deployment. You successfully connected to MongoDB!")
11
12             return client
13         except Exception as e:
14             print(e)
15     except pymongo.errors.ConnectionFailure as e:
16         print(f"Connection failed: {e}")
17         return None
18
19 def load(database, collection):
20     mongo_uri = "mongodb+srv://temp_user:temp_user123@learn-mongo.sv4y2nl.mongodb.net/?retryWrites=true&w=majority&appName=learn-mongo"
21     if not mongo_uri:
22         print("MONGO_URI not set in environment variables")
23
24     mongo_client = get_mongo_client(mongo_uri)
25
26     # Ingest data into MongoDB
27     db = mongo_client[database]
28     col = db[collection]
29
30     return col
```

Di dalam querinya terdapat koneksi, mengakses, dsb.

Dengan membuat file `mongodb_loader.py` dalam folder `plugins` ini akan berfungsi untuk mengelola proses memasukkan (load) data ke dalam database MongoDB.

“MONGODB LOADER”

FINNHUB LOADER

File finnhub_loader.py ini memiliki fungsi untuk mengambil berita umum dari Finnhub, sebuah layanan API yang menyediakan data pasar keuangan.

Dengan membuat objek finnhub client menggunakan api key, memanggil general news dan panggilan data return news

```
plugins > finnhub_loader.py > ...
1 import finnhub
2
3 def scrape_news():
4     finnhub_client = finnhub.Client(api_key="c9e0e9e01919403hqa0open9e9e01919403hqag")
5
6     news = finnhub_client.general_news('general', min_id=0)
7
8     return news
```

POSTGRES LOADER

File postgres_loader.py memiliki fungsi untuk memuat data ke dalam tabel PostgreSQL. Dengan mengimport pustaka yang dibutuhkan, memuat data ke dalam tabel PostgreSQL, Variabel & String Koneksi, Membuat Objek Koneksi juga pesan jika sukses dijalankan.

```
plugins > postgres_loader.py > ...
1 from sqlalchemy import create_engine
2
3
4 def load(data, table_name):
5     user = 'airflow'
6     passwd = 'airflow'
7     hostname = 'postgres'
8     database = 'data_warehouse'
9
10    conn_string = f'postgresql://{{user}}:{{passwd}}@{{hostname}}:5432/{{database}}'
11
12    db = create_engine(conn_string)
13    conn = db.connect()
14
15    data.to_sql(table_name, con=conn, if_exists='append',
16                index=False)
17
18    print("Successfully loaded to postgres")
```

SENTIMENT ANALYSIS

File `sentiment_analysis.py` memiliki fungsi untuk melakukan analisis sentimen terhadap teks yang diberikan menggunakan pustaka `TextBlob`. Terdapat class `SentimentAnalysis`, metode `execute` untuk melakukan analisis sentimen pada teks yang diberikan, membuat objek `TextBlob` juga set `sentiment`. File ini nantinya akan mempengaruhi sentimen analysis loader untuk menjalankannya.

```
plugins > sentiment_analysis.py > ...
1 import re
2 from textblob import TextBlob
3
4
5 class SentimentAnalysis:
6
7     def __init__(self, text):
8         self.text = text
9
10    # only for english language
11    def execute(self):
12        # create TextBlob object of passed tweet text
13        analysis = TextBlob(self.text)
14
15        # set sentiment
16        if analysis.sentiment.polarity > 0:
17            data = {'text': self.text, 'sentiment': 'positive'}
18        elif analysis.sentiment.polarity == 0:
19            data = {'text': self.text, 'sentiment': 'neutral'}
20        else:
21            data = {'text': self.text, 'sentiment': 'negative'}
22
23        return data
24
25
26 if __name__ == "__main__":
27     # calling main function
28     SentimentAnalysis('hard to learn NLTK').execute()
```

```
plugins > finnhub_mongodb_loader.py > ...
1 import finnhub_loader
2 import mongodb_loader
3
4
5 def extract_load():
6     news = finnhub_loader.scrape_news()
7
8     collection = mongodb_loader.load('news', 'finnhub_news')
9     collection.insert_many(news)
10
11    print("Successfully load news to MongoDB")
12
13
14 if __name__ == "__main__":
15     extract_load()
```

File finnhub_mongodb_loader ini digunakan untuk mengekstraksi berita dari sumber data tertentu menggunakan modul finnhub_loader dan kemudian memuat data tersebut ke dalam koleksi MongoDB menggunakan modul mongodb_loader.

FINNHUB MONGODB LOADER

```
(andrew) D:\Downloads\FinalProject_DE_DigitalSkola>python D:\Downloads\FinalProject_DE_DigitalSkola\plugins\finnhub_mongodb_loader.py
Pinged your deployment. You successfully connected to MongoDB!
Successfully load news to MongoDB
```

Setelah query dibuat maka selanjutnya jalankan python sampai berhasil dan cek status query berhasil masuk tidaknya pada MongoDB Atlas.

The screenshot shows the MongoDB Compass interface for the `news.finnhub_news` collection. The collection has a storage size of 112KB, logical data size of 112.22KB, 200 total documents, and index sizes totaling 36KB. The interface includes tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. A search bar allows generating queries from natural language. A query results section shows the first 20 documents of many, with one document highlighted:

```
_id: ObjectId('667bf70f85b1d0180ce7a04f')
category : "top news"
datetime : 1719399060
headline : "Oil prices bounce ahead of official data on U.S. inventories"
id : 7378424
image : "https://static2.finnhub.io/file/publicdatany/finnhubimage/market_watch..."
related : ""
source : "MarketWatch"
summary : "Oil futures rose Wednesday, finding support amid continued jitters ove..."
url : "https://www.marketwatch.com/story/oil-prices-bounce-ahead-of-official-..."
```

SENTIMENT ANALYSIS LOADER

```
plugins > sentiment_analysis_loader.py > ...
1  import mongodb_loader
2  import pandas as pd
3  from sentiment_analysis import SentimentAnalysis
4  import postgres_loader
5
6  def run():
7      db = mongodb_loader.get_data("news", "finnhub_news")
8
9      news = [x for x in db.finnhub_news.find()]
10
11     output = []
12     for news_summary in news:
13         output.append(SentimentAnalysis(text=news_summary["summary"]).execute())
14
15         print(f"Summary {news_summary['summary']} successfully analized")
16
17     sentiment_output = pd.DataFrame(output)
18
19     postgres_loader.load(sentiment_output, "sentiment_news_analysis")
20
21     print("Successfully loaded to Postgres")
22
23 if __name__ == "__main__":
24     run()
```



Skrip Python untuk mengekstraksi database news dari MongoDB, melakukan analisis sentimen pada data tersebut, dan kemudian memuat hasil analisis ke dalam database PostgreSQL.

Cek hasil python dari `sentiment_analysis_loader.py` sampai berhasil keakses.

```
(andrew) D:\Downloads\FinalProject_DE_DigitalSkola>python D:\Downloads\FinalProject_DE_DigitalSkola\plugins\sentiment_analysis_loader.py
Pinged your deployment. You successfully connected to MongoDB!
Summary Oil futures rose Wednesday, finding support amid continued jitters over the potential for a broader Middle East conflict as trader
successfully analized
```

HASIL ANALISIS SENTIMENT

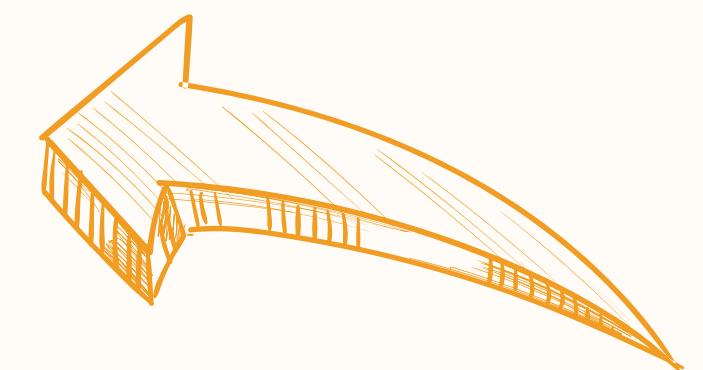
Summary successfully analized

		text sentiment
0	Oil futures rose Wednesday, finding support am...	negative
1	Societe Generale's U.S. equity team says inves...	positive
2	A senior research economist at the Federal Res...	negative
3	"We have no debt and live comfortably on \$150,...	positive
4	Citadel founder Ken Griffin received a rare pu...	positive
..
195	Former Treasury Secretary Larry Summers on Mon...	neutral
196	Are rising jobless claims a sign of emerging w...	negative
197	"Recent election surprises in Mexico, India, a...	neutral
198	The brewer of Corona and Modelo is among the b...	positive
199		neutral

[200 rows x 2 columns]



LOAD HASIL ANALISIS SENTIMENT RE DATA WAREHOUSE



Tampilan pada docker desktop
untuk melihat data warehouse.

finalproject_de_digitalskola-postgres-1

postgres:12
324ac8819086 ⚡
5432:5432 ⚡

Logs Inspect Bind mounts **Exec** Files Stats

```
# psql -Uairflow
psql (12.19 (Debian 12.19-1.pgdg120+1))
Type "help" for help.

airflow=# create database data_warehouse
airflow# \l
```

List of databases						
Name	Owner	Encoding	Collate	Ctype	Access privileges	
airflow	airflow	UTF8	en_US.utf8	en_US.utf8		
data_warehouse	airflow	UTF8	en_US.utf8	en_US.utf8		
postgres	airflow	UTF8	en_US.utf8	en_US.utf8		
template0	airflow	UTF8	en_US.utf8	en_US.utf8	=c/airflow	+
					airflow=CTc/airflow	
template1	airflow	UTF8	en_US.utf8	en_US.utf8	=c/airflow	+
					airflow=CTc/airflow	

(5 rows)

```
dags > dag_sentiment_analysis.py > ...
1  from airflow import DAG
2  from airflow.operators.python import PythonOperator
3  from datetime import datetime, timedelta
4
5  import finnhub_mongodb_loader
6  import sentiment_analysis_loader
7
8
9
10 default_args = {
11     'owner': 'de-team',
12     'depends_on_past': False,
13     'start_date': datetime(2023, 9, 21),
14     'wait_for_downstream': False,
15     'retries': 1,
16     'retry_delay': timedelta(minutes=5),
17     'sla': timedelta(days=1),
18 }
19
20 schedule_interval = '0 0 * * *'
21
22 with DAG(
23     'dag_sentiment_analysis',
24     default_args=default_args,
25     schedule_interval=schedule_interval,
26     catchup=False,
27     tags=['machine-learning']
28 ) as dag:
29     extract_load = PythonOperator(
30         task_id=f'extract_load',
31         python_callable=finnhub_mongodb_loader.extract_load
32     )
33
34     sa_load = PythonOperator(
35         task_id=f'sa_load',
36         python_callable=sentiment_analysis_loader.run
37     )
38
39     extract_load >> sa_load
```

SENTIMENT ANALYSIS FOR AIRFLOW SCHEDULE

Script python ini berguna untuk membuat dan mengelola Directed Acyclic Graph (DAG) menggunakan Apache Airflow yang berfungsi untuk mengotomatisasi proses ekstraksi data dari database news, melakukan analisis sentimen, dan pemuatan data ke dalam MongoDB dan PostgreSQL.

HASIL SCHEDULED AIRFLOW

A screenshot of the Airflow web interface. At the top, there's a navigation bar with tabs like Airflow, DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs. The main area shows a DAG named "dag_sentiment_analysis" with a schedule of "0 0 * * *" and the next run ID of "2024-06-25, 00:00:00". Below this, there are tabs for Grid, Graph, Calendar, Task Duration, Task Tries, Landing Times, Gantt, Details, Code, and Audit Log. The Grid tab is selected. The main content area displays the DAG summary, which includes the total number of tasks (2), PythonOperators (2), and details such as Dag id: "dag_sentiment_analysis", Description: "null", Fileloc: "/opt/airflow/dags/dag_sentiment_analysis.py", Has import errors: "false", and Has task concurrency limits: "false".

DAG: dag_sentiment_analysis

Schedule: 0 0 * * * | Next Run ID: 2024-06-25, 00:00:00

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details Code Audit Log

26/06/2024 13:16:56 25 All Run Types All Run States Clear Filters

deferred failed queued removed restarting running scheduled skipped success up_for_reschedule up_for_retry upstream_failed no_status

extract_load sa_load

DAG Summary

Total Tasks	2
PythonOperators	2

DAG Details

Dag id	dag_sentiment_analysis
Description	null
Fileloc	/opt/airflow/dags/dag_sentiment_analysis.py
Has import errors	false
Has task concurrency limits	false

Tampilan dags saat berhasil memanggil dan menjalankan airflow.



DigitalSkola

**TERIMA
KASIH**

ATAS PERHATIANNYA

BY 8DREAM (KELOMPOK 1)