Automated Data Pipeline for Sentiment Analysis Using Apache Airflow

SIB A-INFOTECH

Team Member

Muhamad Fahrurrozi

Muhammad Mahda

Nareswari Huwaida N

Muhsyi Fadhilah Hirani

CHAPTER 1

Project Overview

Project Overview

PRODUCT → AUTOMATED DATA PIPELINE

TEAM → DATA PIPELINE ENGINEER

DATA PIPELINE ENGINEER → AUTOMATED DATA PIPELINE & SENTIMENT ANALYSIS SYSTEM

BUSSINES PROCESS

MANUAL

AMBIL DATA DARI FINNHUB

DISIMPAN DI MONGODB

DI ANALISIS SENTIMEN

HASILNYA DISIMPAN DI POSTGRES

AUTOMATED DATA PIPELINE

Apache Airflow

DATASET TRANSFER FINNHUB KE ATLAS

SENTIMENT ANALYSIS + TRANSFER ATLAS KE POSTGRES

AUTOMATED DATA PIPELINE MENGGUNAKAN AIRFLOW

WORKFLOW

1. LOAD DATA TO MONGODB
2. MACHINE LEARNING TO SENTIMENT ANALYSIS
3. RESULTH DATA TO POSTGRES
4. AUTOMATION AND SCHEDULER AIRFLOW

PROBLEM

Proses manual dalam pengumpulan dan penyimpanan data saat ini tidak hanya kurang efisien tetapi juga memakan waktu yang lama, menyebabkan keterlambatan dalam analisis dan respon terhadap informasi pasar yang cepat berubah.

GOALS

Menghemat waktu dan Meminimalisir human error karena proses pengumpulan data analisis data dilakukan secara otomatis

OBJECTIVE

Membangun sistem otomatis yang dapat mengumpulkan, menganalisis sentimen, dan menyimpan data berita untuk memberikan wawasan pasar

BUSINESS METRICS

- Ketepatan waktu
- Responsivitas Data

CHAPTER 2

Implementation

Step-by-Step Process

Finnhub Data Extraction

Menggunakan Finnhub API untuk mengambil data berita

Data Loading

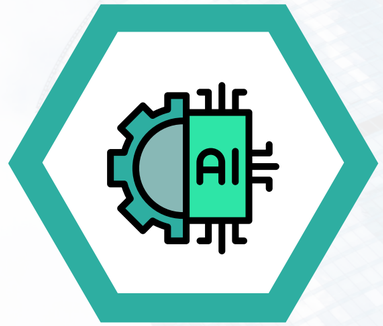
Menyimpan data mentah ke dalam MongoDB

Sentiment Analysis

Menganalisis sentimen dari data tersebut

Result Storage

Menyimpan hasil analisis sentimen ke dalam PostgreSQL



Automation

Menjadwalkan tugas dengan Apache Airflow

Technologies Used

1. Docker Compose
2. Python Scripts
3. Apache Airflow
4. MongoDB & PostgreSQL

KONFIGURASI DOCKER COMPOSE

Postgres

Menggunakan postgres:12, pengaturan pengguna dan database

airflow-init

Inisialisasi DB dan pembuatan pengguna admin

airflow-webserver & airflow-scheduler

Instal requirements dan menjalankan layanan terkait

Volumes & Networks

Direktori mounting dan pengaturan jaringan

CODE SNIPPETS

Highlight key parts of the code

```yaml
&airflow-common
  image: apache/airflow:2.8.4-python3.10
  environment:
    - AIRFLOW__CORE__EXECUTOR=LocalExecutor
    - AIRFLOW__CORE__SQL_ALCHEMY_CONN=postgresql+psycopg2://airflow:airflow@postgres:5432/airflow
    - AIRFLOW__CORE__FERNET_KEY=FB0o_zt4e3Ziq3LdUUO7F2Z95cvFFx16hU8jTeR1ASM=
    - AIRFLOW__CORE__LOAD_EXAMPLES=False
    - AIRFLOW__CORE__LOGGING_LEVEL=INFO
  env_file:
    - /.env
  user: "${AIRFLOW_UID:-50000}:0"
  volumes:
    - ./dags:/opt/airflow/dags
    - ./source:/opt/airflow/source
    - ./output:/opt/airflow/output
    - ./logs:/opt/airflow/logs
    - ./plugins:/opt/airflow/plugins
    - ./requirements.txt:/requirements.txt
```

```yaml
services:
  postgres:
    image: postgres:12
    environment:
      - POSTGRES_USER=airflow
      - POSTGRES_PASSWORD=airflow
      - POSTGRES_DB=airflow
    volumes:
      - postgres-db-volume:/var/lib/postgresql/data
    healthcheck:
      test: [ "CMD", "pg_isready", "-U", "airflow" ]
      interval: 5s
      retries: 5
    ports:
      - "5432:5432"
    networks:
      - airflow_network
```

```yaml
airflow-init:
  <<: *airflow-common
  container_name: airflow_init
  entrypoint: /bin/bash
  command:
    - -c
    - >
      airflow db init &&
      airflow users create --role Admin
                           --username airflow
                           --password airflow
                           --email airflow@airflow.com
                           --firstname airflow
                           --lastname airflow
  restart: on-failure
  depends_on:
    postgres:
      condition: service_healthy
  networks:
    - airflow_network
```

CODE SNIPPETS

```yaml
airflow-webserver:
    <<: *airflow-common
    command: >
      bash -c "python -m pip install --upgrade pip && pip install --no-cache-dir -r /requirements.txt && airflow webserver"
    ports:
      - 8080:8080
    container_name: airflow_webserver
    depends_on:
      airflow-init:
        condition: service_completed_successfully
    restart: always
    networks:
      - airflow_network

  airflow-scheduler:
    <<: *airflow-common
    command: >
      bash -c "python -m pip install --upgrade pip && pip install --no-cache-dir -r /requirements.txt && airflow scheduler"
    container_name: airflow_scheduler
    depends_on:
      airflow-init:
        condition: service_completed_successfully
    restart: always
    networks:
      - airflow_network
```

Highlight key parts of the code

Ringkasan Proses

Mengumpulkan dan Menyimpan Data ke MongoDB

Koneksi ke MongoDB

Scraping Data Berita

Memuat Data ke MongoDB

Analisis Sentimen dan load ke PostgreSQL

Inisialisasi Koneksi → Analisis Sentimen → Mengonversi ke DataFrame → Load DF ke PostgresSQL

Code Snippets

```python
def get_mongo_client(mongo_uri):

    try:
        client = pymongo.MongoClient(mongo_uri)
        client.admin.command('ping')

        print("Successfully connected to MongoDB!")
        return client

    except pymongo.errors.ConnectionFailure as e:
        print(f"Connection failed: {e}")

        return None
```

Koneksi MongoDB

Pengambilan Data

```python
def scrape_news():
    api_key = os.getenv("FINNHUB_API_KEY")
    finnhub_client = finnhub.Client(api_key=api_key)
    news = finnhub_client.general_news('general', min_id=0)
    return news
```

Memuat Data ke MongoDB

```python
def extract_load():
    news = finnhub_loader.scrape_news()
    collection = mongodb_loader.load('news', 'finnhub_news')
    collection.insert_many(news)
    print("Successfully load news to MongoDB")
```

Code Snippets

Analisis Sentimen

```python
class SentimentAnalysis:
    def execute(self):
        analysis = TextBlob(self.text)
        if analysis.sentiment.polarity > 0:
            data = {'text': self.text, 'sentiment': 'positive'}
        elif analysis.sentiment.polarity == 0:
            data = {'text': self.text, 'sentiment': 'neutral'}
        else:
            data = {'text': self.text, 'sentiment': 'negative'}
        return data

```

Inisialisasi Koneksi MongoDB dan Pengambilan Data

```python
def run_analysis():
    mongo_uri = os.getenv("MONGO_URI")
    db = get_mongo_client(mongo_uri)
    news_collection = db['news']['finnhub_news']

    news = [x for x in news_collection.find()]
```

Memuat Data ke PostgreSQL

```python
def load(data, table_name):
    conn_string = 'postgresql://airflow:airflow@postgres:5432/data_warehouse'
    engine = create_engine(conn_string)
    data.to_sql(table_name, con=engine, if_exists='append', index=False)
    print("Successfully loaded to Postgres")
```

CHAPTER 3

Results

Summary, Impact and Metrics

Summary

1 Tujuan dari pengerjaan project final

Impact

2 Ketepatan waktu,Akurasi Data and responsivitas data

Metrics

3 Ketepatan Waktu Eksekusi terjadwal and Tingkat kesalahan data

Summary

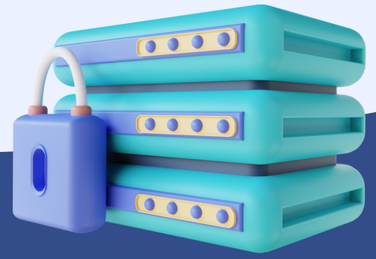
bertujuan untuk mengotomatisasi proses pengumpulan dan analisis data sentiment dari Finnhub dengan menggunakan pipeline Airflow, Pipeline tersebut mengintegrasikan beberapa komponen penting.

Finnhub

pengambilan data

PostgreSQL

penyimpanan data akhir



MongoDB

penyimpanan data sementara

TextBlob

analisis sentiment

Impact

MANUAL

AMBIL DATA DARI FINNHUB

DISIMPAN DI MONGODB

DI ANALISIS SENTIMEN

HASILNYA DISIMPAN DI POSTGRES

penggunaan waktu sekitar 20-40 menit untuk analysis secara manual oleh individu

OTOMATIS

DATASET TRANSFER FINNHUB KE ATLAS

SENTIMENT ANALYSIS + TRANSFER ATLAS KE POSTGRES

AUTOMATED DATA PIPELINE MENGGUNAKAN AIRFLOW

penggunaan waktu sekitar 10-30 detik untuk menganalysis sentiment, dengan penjadwalan airflow

Impact

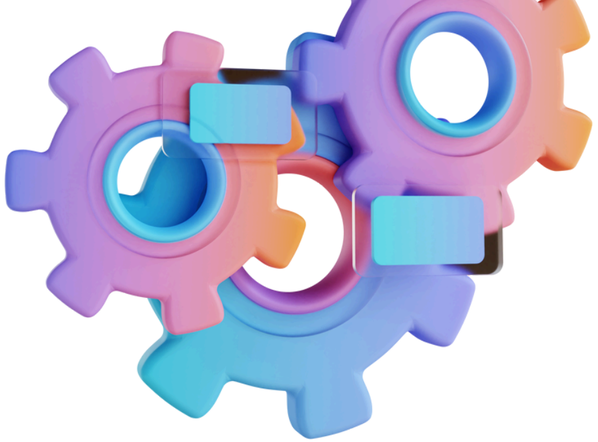
- Time : 20 menit menjadi 20 detik
- Man : dari 1 orang menjadi diatur oleh sistem

60x lebih cepat

Time

98%

Man

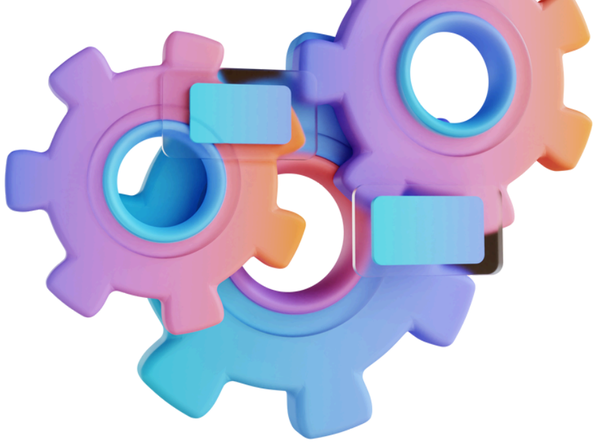


Impact

Efisiensi Waktu

Akurasi Data

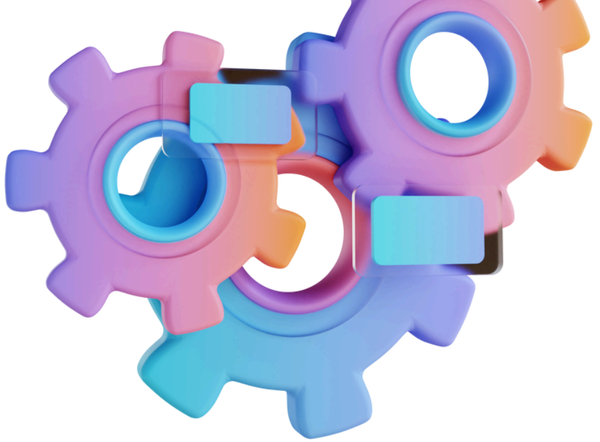
Responsivitas



Metrics

Implementasi project menghasilkan metrics bussines, dimana pada project ini mencapai tujuan yaitu responsivitas data dan juga ketepatan waktu

Metrics

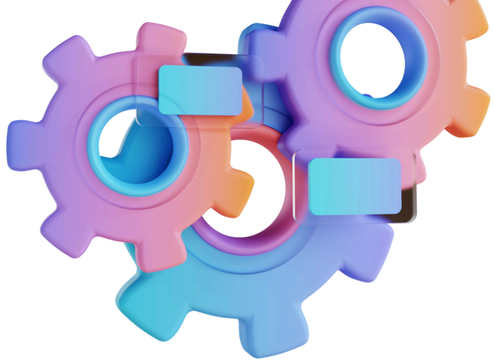


Ketepatan Waktu Eksekusi Terjadwal (Scheduled Execution Time Accuracy)

- Secara manual : Dari 30 eksekusi yang dijadwalkan dalam satu bulan, hanya 20 yang dijalankan tepat waktu (65.5%).
- Secara otomatis : Dari 30 eksekusi yang dijadwalkan dalam setiap hari, dijalankan tepat waktu pada jam 00.00.

Keterlambatan dalam Pipeline Data (Latency in Data Pipeline)

- Secara manual : Rata-rata waktu pemrosesan adalah 20 - 30 menit.
- Secara otomatis : Rata-rata waktu pemrosesan berkurang menjadi 20 - 30 detik



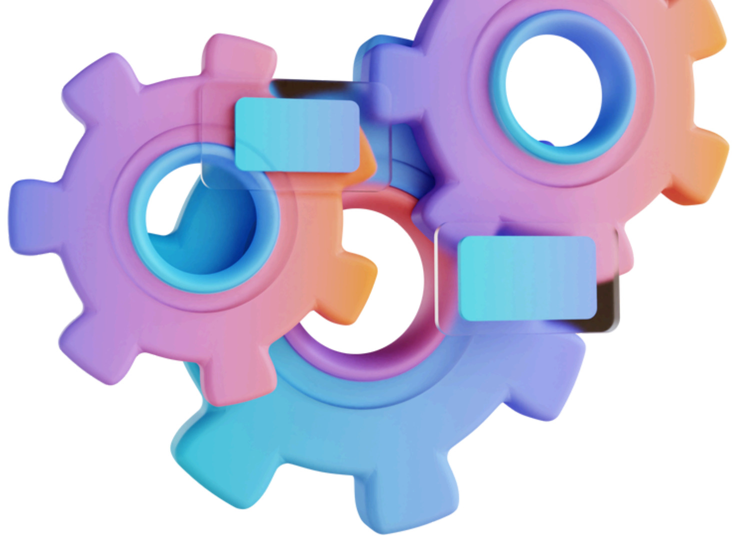
Result for Project

Connect in mongoDB

```
mahda@LAPTOP-6QU2AGRG:/mnt/c/Users/HP/STUPEN/Airflow_Data_Pipeline_for_Sentiment_Analysis/plugins$ python3 finnhub_mongodb_loader.py
/home/mahda/.local/lib/python3.8/site-packages/pymongo/network.py:36: UserWarning: Failed to use the installed version of PyOpenSSL. Fal
ling back to stdlib ssl, disabling OCSP support. This is likely caused by incompatible versions of PyOpenSSL < 23.2.0 and cryptography >
= 42.0.0. Try updating PyOpenSSL >= 23.2.0 to enable OCSP.
  from pymongo import _csot, helpers, message, ssl_support
Pinged your deployment. You successfully connected to MongoDB!
Successfully load news to MongoDB
```

Connected in Postgres

```
mahda@LAPTOP-6QU2AGRG:/mnt/c/Users/HP/STUPEN/Airflow_Data_Pipeline_for_Sentiment_Analysis/plugins$ python3 sentiment_analysis_loader.py
/home/mahda/.local/lib/python3.8/site-packages/pymongo/network.py:36: UserWarning: Failed to use the installed version of PyOpenSSL. Fal
ling back to stdlib ssl, disabling OCSP support. This is likely caused by incompatible versions of PyOpenSSL < 23.2.0 and cryptography >
= 42.0.0. Try updating PyOpenSSL >= 23.2.0 to enable OCSP.
  from pymongo import _csot, helpers, message, ssl_support
Pinged your deployment. You successfully connected to MongoDB!
Summary 'Vanguard data casts a shadow on the economy' successfully analyzed
Summary 'The Federal Reserve should be able to cut interest rates with inflation coming down and the labor market cooling, but the timin
g of any move is uncertain, Federal Reserve governor Lisa Cook said Tuesday.' successfully analyzed
Berhasil dimuat ke PostgreSQL
Successfully loaded to Postgres
```



Result for Project

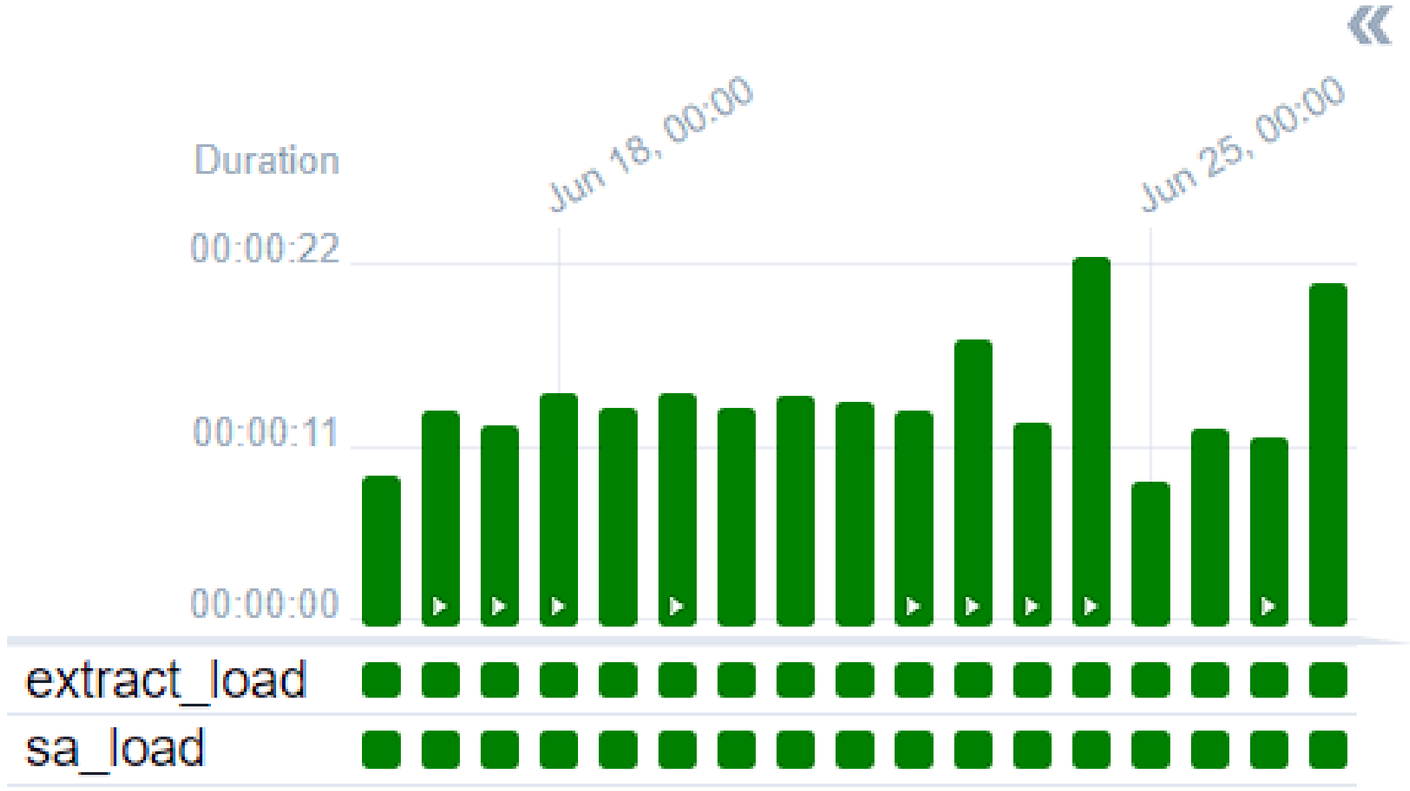
Connect in data_warehouse

```
data_warehouse-# \c data_warehouse
You are now connected to database "data_warehouse" as user "airflow".
data_warehouse-# \d
              List of relations
 Schema |          Name            | Type  |  Owner
--------+--------------------------+-------+---------
 public | sentiment_news_analysis | table | airflow
(1 row)
```

displays data in the data_warehouse table

```
                                                                    text
                                                  | sentiment
------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------+-----------
 Each team gets about $10 million for taking part in the European Championships — but none of that goes directly to the players.
                                                  | positive
 Have a clear road map for your estate so your heirs don't drown in paperwork and probate after your death
                                                  | positive
 Roaring Kitty weighed in on the event with fresh cryptic tweets.
                                                  | positive
 Oil futures edged higher Tuesday, on track to notch back-to-back gains.
                                                  | positive
```

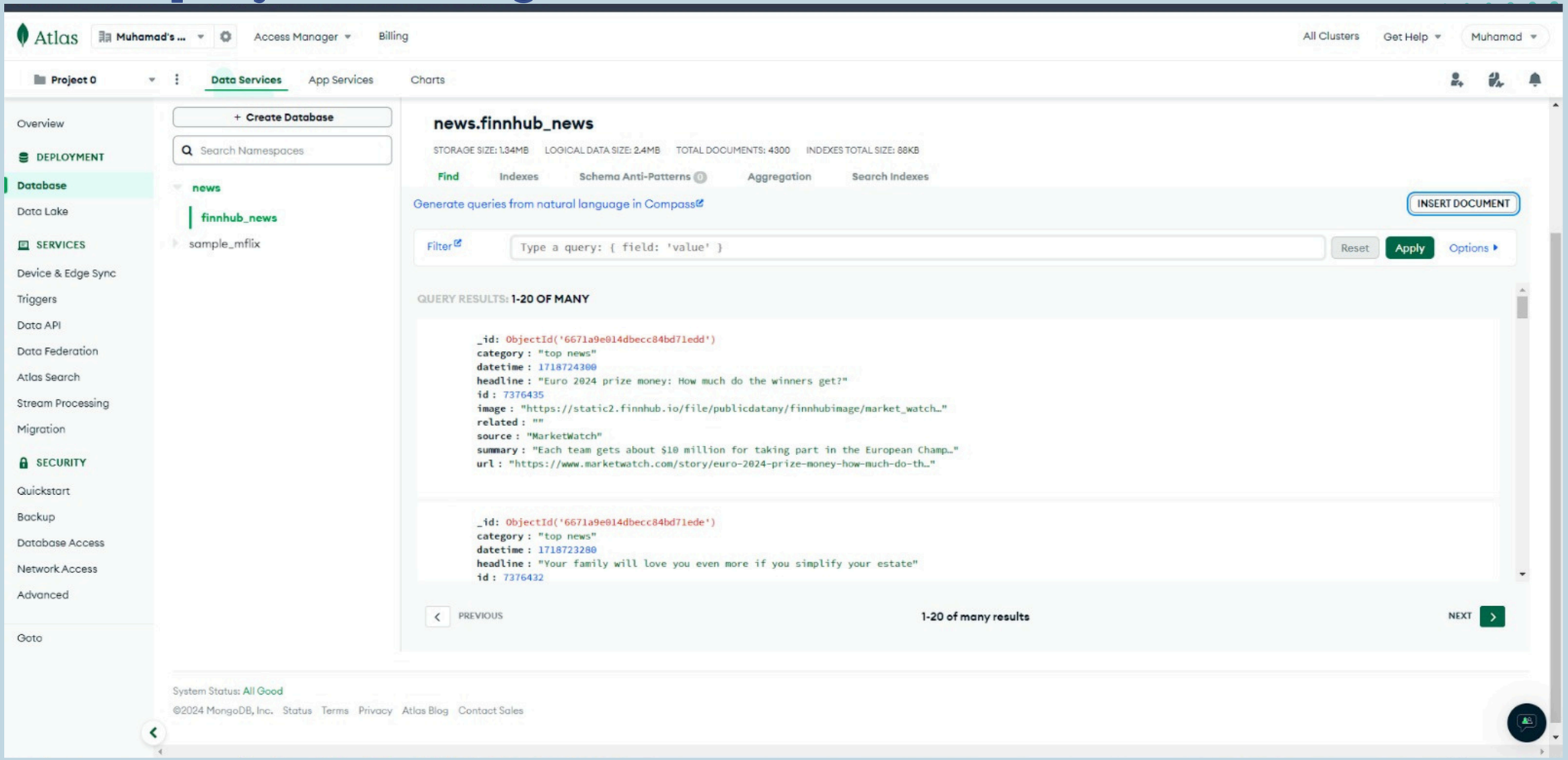
Result in otomatisasi



implementation Airflow

Result for Project

displays in MongoDB atlas



CHAPTER 4

Conclusion

Implementasi pipeline otomatis telah berhasil mencapai tujuan yang ditetapkan, yaitu menghemat waktu, meminimalisir kesalahan manusia, meningkatkan ketepatan waktu, dan meningkatkan responsivitas data. Dengan hasil ini, perusahaan dapat memperoleh insight yang lebih cepat dan akurat, memungkinkan pengambilan keputusan yang lebih baik dan lebih responsif terhadap perubahan pasar.

CHAPTER 5

Discussion

INFOTECH

Thank You

LET'S DISCUSS!