Eksplorasi Pasar Mobil Bekas di Platform Carmudi.co.id dengan Web Scraping dan Data Analytics

ARIFIN SATRIA AJINUSA

Arifin S Ajinusa

Education

Informatika, Universitas Sebelas Maret

Working

Data Analyst at FinAccel

Project Background

PROJECT BACKGROUND

- **Deskripsi:** Proyek ini membangun data pipeline untuk menganalisis penjualan mobil bekas dari Carmudi.co.id, dengan tujuan memberikan insight yang akurat tentang tren pasar. Data dikumpulkan melalui web scraping, diproses menggunakan Apache Airflow sebagai orkestrator pipeline. Kemudian, data divisualisasikan di Metabase dalam bentuk dashboard interaktif dan insightful.

PROJECT BACKGROUND

- **Tujuan:**
  - Menyediakan insight tentang tren harga dan penjualan mobil bekas.
  - Mempermudah analisis berdasarkan merek, lokasi, tahun produksi, dan fitur lainnya.
  - Mengotomatiskan pengambilan dan pembaruan data dengan Apache Airflow.
  - Membuat dashboard yang mudah dipahami untuk memantau dinamika pasar.

- **Hasil yang Diharapkan:** Insightful Dashboard Pasar Mobil Bekas

Problem Statement

Tantangan dalam pengembangan project ini di antaranya:

1. Membuat environment dengan docker untuk dapat mengimplementasikan Airflow, Metabase, Pyspark, Postgres dan Mysql
2. Melakukan scraping data dari Carmudi.co.id
3. Menyimpan data hasil scraping ke dalam datalake dalam hal ini menggunakan Postgres
4. Membuat Datamart dengan melakukan transformasi data dari Datalake Postgres ke Mysql
5. Membuat visualisasi data dari Datamart yang telah dibuat

ibimbing

Data Source	Data Lake	Data Mart	Visualization
Web Scraping: carmudi INDONESIA	PostgreSQL	MySQL	Metabase

Code Overview: Airflow

docker-compose-airflow.yml

```yaml
services:
    scheduler:
        image: ajinusa-de8-dibimbing/airflow
        container_name: ${AIRFLOW_SCHEDULER_CONTAINER_NAME}
        hostname: ${AIRFLOW_SCHEDULER_CONTAINER_NAME}
        command: scheduler
        restart: always
        environment:
            - AIRFLOW__CORE__SQL_ALCHEMY_CONN=postgresql+psycopg2:/
            - AIRFLOW__CORE__EXECUTOR=LocalExecutor
            - POSTGRES_USER=${POSTGRES_USER}
            - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
            - POSTGRES_DB=${POSTGRES_DB}
            - POSTGRES_DW_DB=${POSTGRES_DW_DB}
            - POSTGRES_CONTAINER_NAME=${POSTGRES_CONTAINER_NAME}
            - POSTGRES_PORT=${POSTGRES_PORT}
            - SPARK_MASTER_HOST_NAME=${SPARK_MASTER_HOST_NAME}
            - SPARK_MASTER_PORT=${SPARK_MASTER_PORT}
            - AIRFLOW__CORE__SQL_ALCHEMY_CONN=mysql+mysqldb://root:
            # - AIRFLOW__CORE__SQL_ALCHEMY_CONN=mysql+mysqldb://roo
        volumes:
            - ../dags:/opt/airflow/dags
```

docker > docker-compose-airflow.yml

```yaml
services:
    webserver:
        image: ajinusa-de8-dibimbing/airflow
        container_name: ${AIRFLOW_WEBSERVER_CONTAINER_
        hostname: ${AIRFLOW_WEBSERVER_CONTAINER_NAME}
        entrypoint: /scripts/entrypoint.sh
        restart: always
        depends_on:
            - scheduler
        environment:
            - AIRFLOW__CORE__SQL_ALCHEMY_CONN=postgres
            - AIRFLOW__CORE__EXECUTOR=LocalExecutor
            - POSTGRES_USER=${POSTGRES_USER}
            - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
            - POSTGRES_DB=${POSTGRES_DB}
            - POSTGRES_DW_DB=${POSTGRES_DW_DB}
            - POSTGRES_CONTAINER_NAME=${POSTGRES_CONTA
            - POSTGRES_PORT=${POSTGRES_PORT}
            - SPARK_MASTER_HOST_NAME=${SPARK_MASTER_HO
            - SPARK_MASTER_PORT=${SPARK_MASTER_PORT}
            # - AIRFLOW__CORE__SQL_ALCHEMY_CONN=mysql+
            - AIRFLOW__CORE__SQL_ALCHEMY_CONN=mysql+my
```

Code Overview: Mysql

docker > docker-compose-mysql.yml

```yaml
version: '3.8'

services:
  mysql:
    image: mysql:latest
    container_name: ajinusa-mysql-container
    environment:
      MYSQL_ROOT_PASSWORD: ajinusa
      MYSQL_DATABASE: de8_final_project
    ports:
      - "3306:3306"
    networks:
      - ajinusa-de8-network
    restart: always
    volumes:
      - mysql-data:/var/lib/mysql   # Untuk menyimpan data MySQL di volume
      - ./my.cnf:/etc/mysql/my.cnf  # Menambahkan file konfigurasi custom

volumes:
  mysql-data:

# networks:
```

Code Overview: Postgres

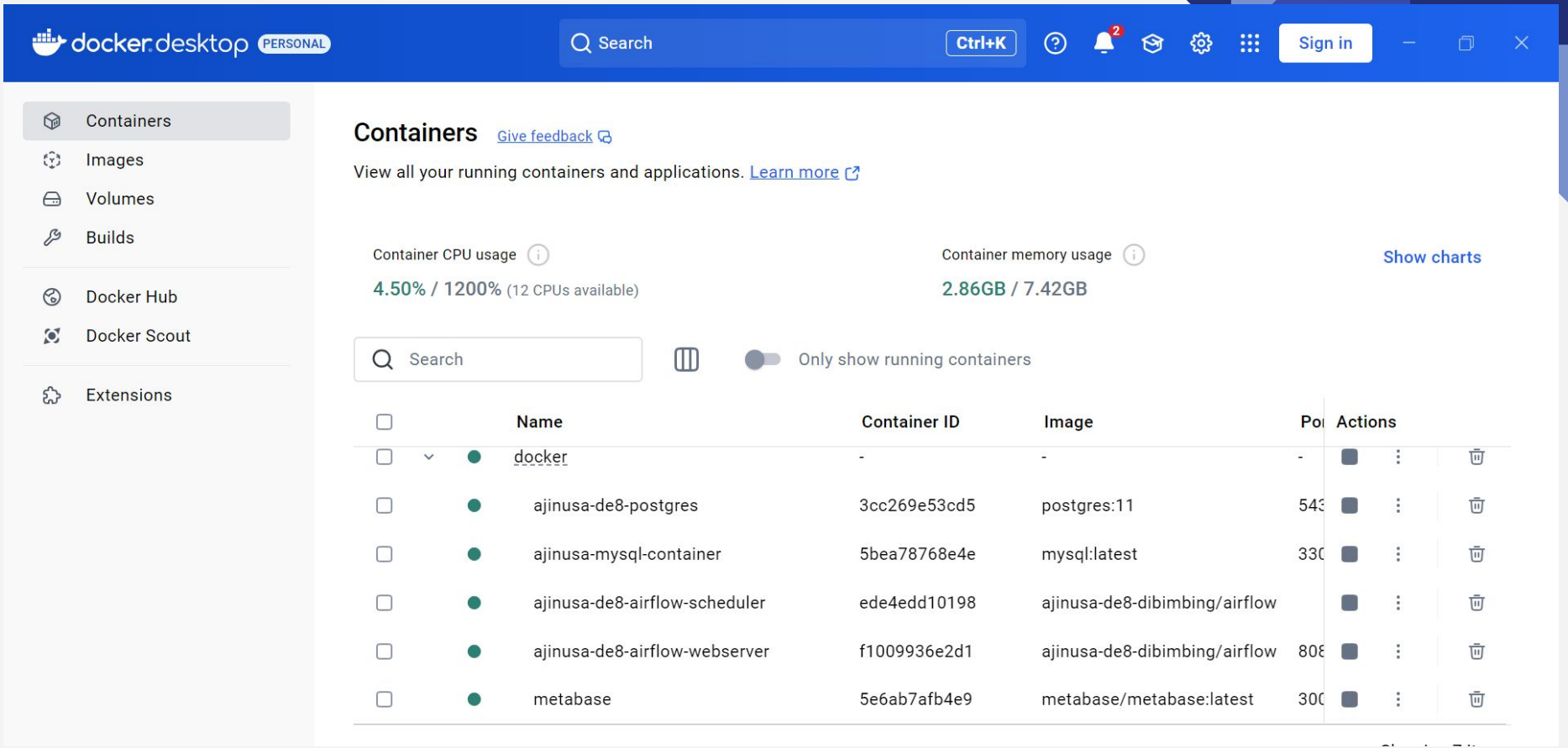
docker > docker-compose-postgres.yml

```yaml
version: "3.8"

services:
    ajinusa-de8-postgres:
        image: postgres:11
        container_name: ${POSTGRES_CONTAINER_NAME}
        restart: unless-stopped
        hostname: ${POSTGRES_CONTAINER_NAME}
        networks:
            - ajinusa-de8-network
        environment:
            - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
            - POSTGRES_USER=${POSTGRES_USER}
            - POSTGRES_DB=${POSTGRES_DB}
            - PGDATA=/var/lib/postgresql/data/pgdata
        volumes:
            - ../sql:/sql
            - ../data:/data
        ports:
            - ${POSTGRES_PORT}:5432
```

Code Overview: Metabase

docker > docker-compose-metabase.yml

```yaml
version: '3.8'

services:
  metabase:
    image: metabase/metabase:latest
    container_name: metabase
    ports:
      - "3000:3000"  # Port untuk akses Metabase
    environment:
      - MB_DB_TYPE=mysql
      - MB_DB_DBNAME=de8_final_project  # Nama database yang digunakan di MySQL
      - MB_DB_HOST=ajinusa-mysql-container  # Nama container MySQL
      - MB_DB_USER=root  # Username untuk MySQL
      - MB_DB_PASS=ajinusa  # Password untuk MySQL
      - MB_DB_PORT=3306  # Port MySQL
    restart: always
    networks:
      - ajinusa-de8-network

```

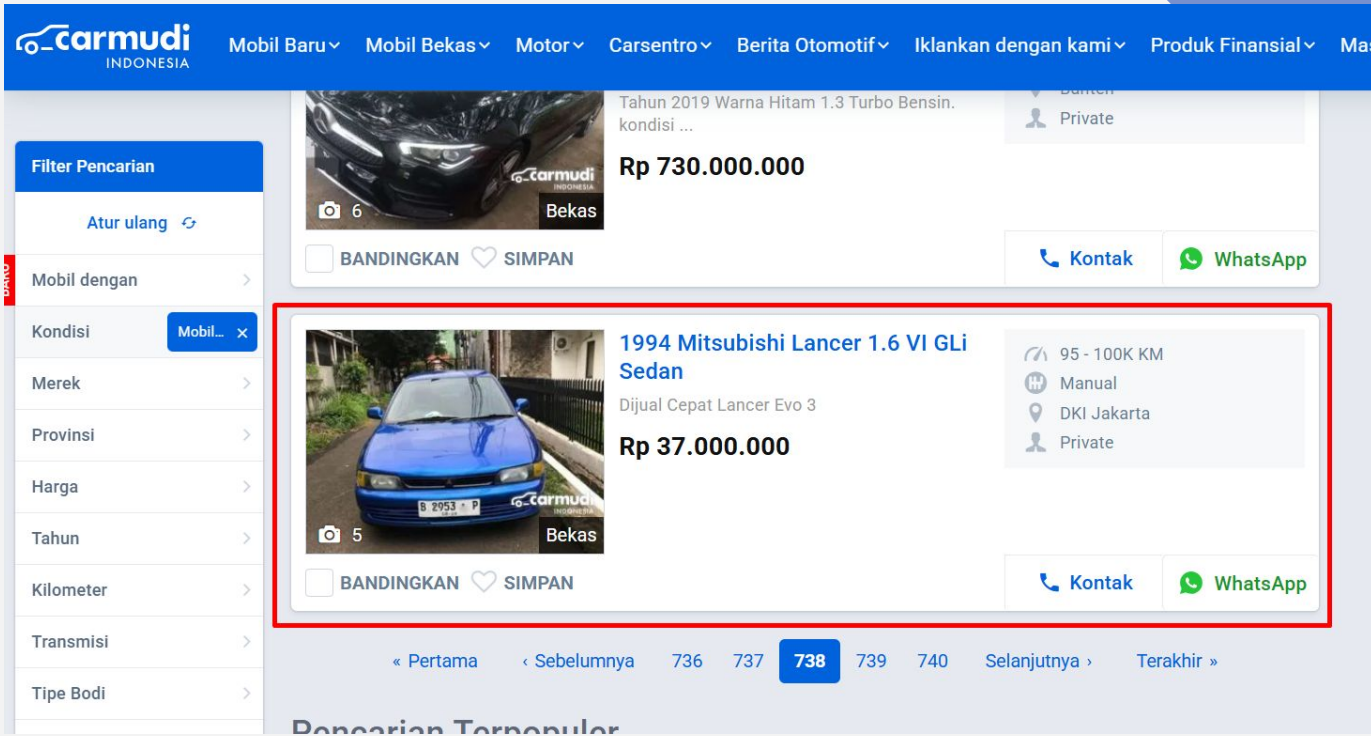

docker desktop PERSONAL
Search
Ctrl+K
Sign in
Containers
Images
Volumes
Builds
Docker Hub
Docker Scout
Extensions
Containers
Give feedback
View all your running containers and applications. Learn more
Container CPU usage
4.50% / 1200% (12 CPUs available)
Container memory usage
2.86GB / 7.42GB
Show charts
Search
Only show running containers
Name
Container ID
Image
Actions
docker
ajinusa-de8-postgres
3cc269e53cd5
postgres:11
ajinusa-mysql-container
5bea78768e4e
mysql:latest
ajinusa-de8-airflow-scheduler
ede4edd10198
ajinusa-de8-dibimbing/airflow
ajinusa-de8-airflow-webserver
f1009936e2d1
ajinusa-de8-dibimbing/airflow
metabase
5e6ab7afb4e9
metabase/metabase:latest

Data Understanding

- **Sumber Data:** Carmudi.co.id
- **Jenis data:** Data Mobil Bekas
- **Jumlah data:** 18,000
- **Jumlah Kolom :** 12

Data yang diambil di antaranya:

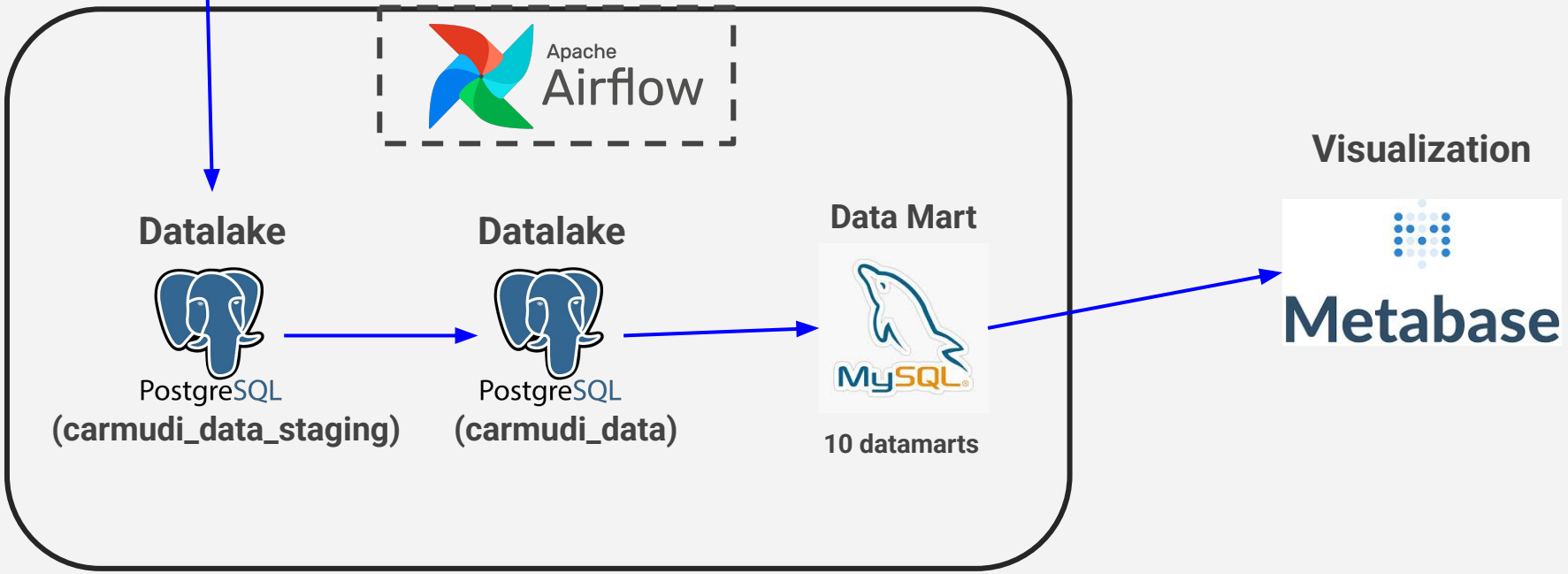
1. Judul
2. Tahun
3. Merek
4. Nama Mobil
5. Harga
6. Kilometer
7. Transmisi
8. Lokasi
9. Tipe Penjual
10. WA Number
11. Link Iklan
12. *snapshot_dt*

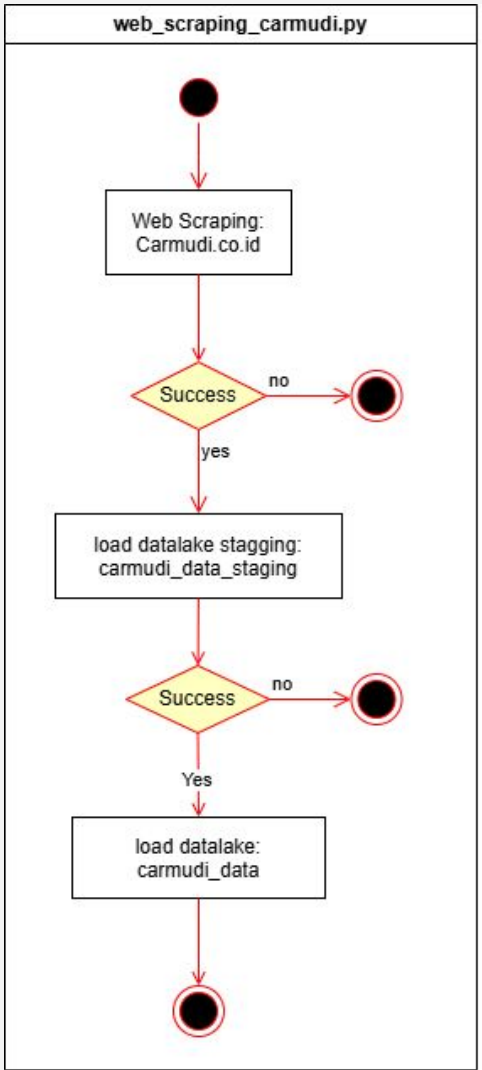


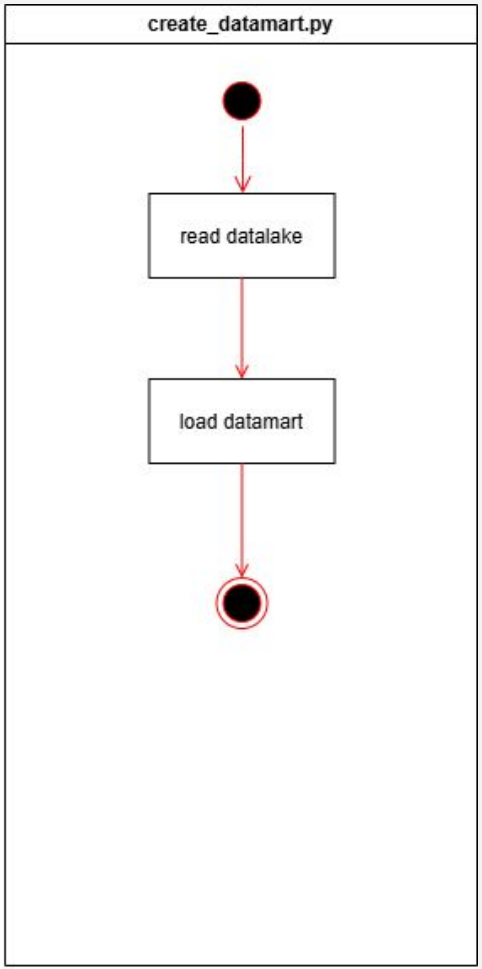
Web Scraping:

carmudi INDONESIA

Architecture Pipeline




web_scraping_carmudi.py
Web Scraping:
Carmudi.co.id
Success
no
yes
load datalake stagging:
carmudi_data_staging
Success
no
Yes
load datalake:
carmudi_data


create_datamart.py
read datalake
load datamart

DAGs

DAGs : Web Scraping

dags > web_scraping_carmudi.py > web_scraping > extract_web > convert_to_number

```
29  def web_scraping():
45      def extract_web(param1, param3):
47          # for url in urls:
48          for i in range(1, param3 + 1):
49              url = f"{param1}{i}"
50
51              # Request ke website
52              response = requests.get(url, headers=headers)
53
54              # Parsing HTML dengan BeautifulSoup
55              soup = BeautifulSoup(response.text, "html.parser")
56
57              # Scrape data mobil
58              items = soup.find_all("article", class_="listing")  # Sesuaikan dengan class dari Carmudi
59              # print(items)
60              def convert_to_number(rp_string):
61                  return int(rp_string.replace("Rp", "").replace(".", "").strip())
62
63              for item in items:
64                  try:
65                      judul = item.find("h2", class_="listing__title").text.strip()
66                      tahun = judul.split(" ")[0]
```

DAGs : Web Scraping

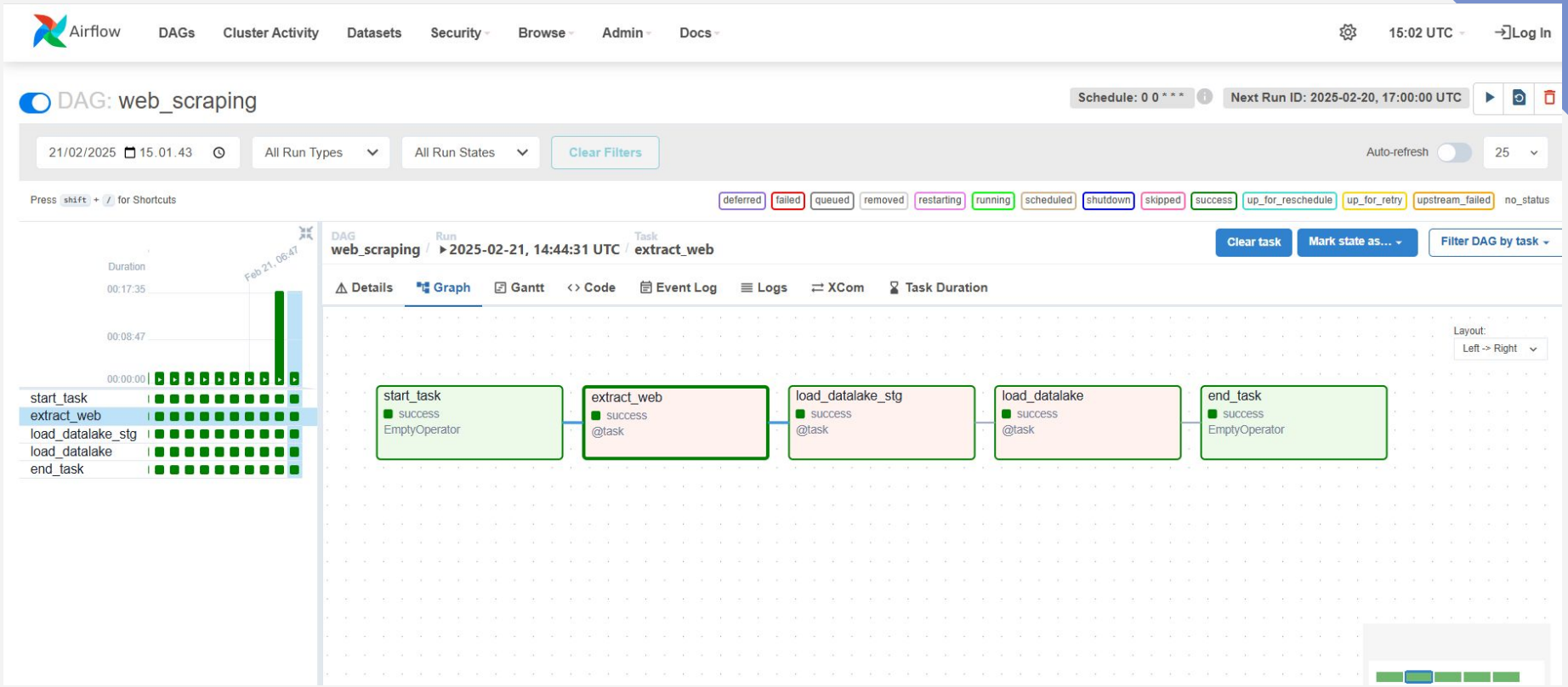
dags > web_scraping_carmudi.py > web_scraping > load_datalake_stg

```
29  def web_scraping():
111
112     @task
113     def load_datalake_stg(df, table_name):
114         # ========== postgres ==========
115
116         # DATABASE_URL = "postgresql://ajinusa:ajinusa@ajinusa-de8-postgres:5432/de8_final_project"
117
118         # Membuat engine untuk koneksi ke PostgreSQL
119         # engine = sa.create_engine(DATABASE_URL)
120         engine = PostgresHook("de8_final_project").get_sqlalchemy_engine()
121
122         # Menyimpan DataFrame ke PostgreSQL (untuk tarikan pertama kali)
123         # df.to_sql("carmudi_data_staging", engine, if_exists='replace', index=False)
124
125         # Menyimpan DataFrame ke PostgreSQL (untuk tarikan terupdate halaman 1 dan 2)
126         df.to_sql("carmudi_data_staging", engine, if_exists='append', index=False)
127
```

DAGs : Web Scraping

```
def web_scraping():
    def load_datalake(table_name):
                query = """
                with cte_data as (
                    select distinct judul, harga, tahun, merek, nama_mobil, kilometer, transmisi, lokasi, dealer, phone_number, max(link) l
                    from carmudi_data_staging
                    group by 1,2,3,4,5,6,7,8,9,10
                )

                select * from cte_data a
                """


                df_read_datalake = pd.read_sql(query, connection)

                jakarta_tz = pytz.timezone('Asia/Jakarta')
                df_read_datalake['snapshot_dt'] = pd.to_datetime(datetime.now(jakarta_tz))
                df_read_datalake['snapshot_dt'] = df_read_datalake['snapshot_dt'].dt.tz_convert('Asia/Jakarta').astype(str)

                df_read_datalake.to_sql(table_name, engine, if_exists='replace', index=False)
```

DAGs : Web Scraping



DAGs : Create Datamart

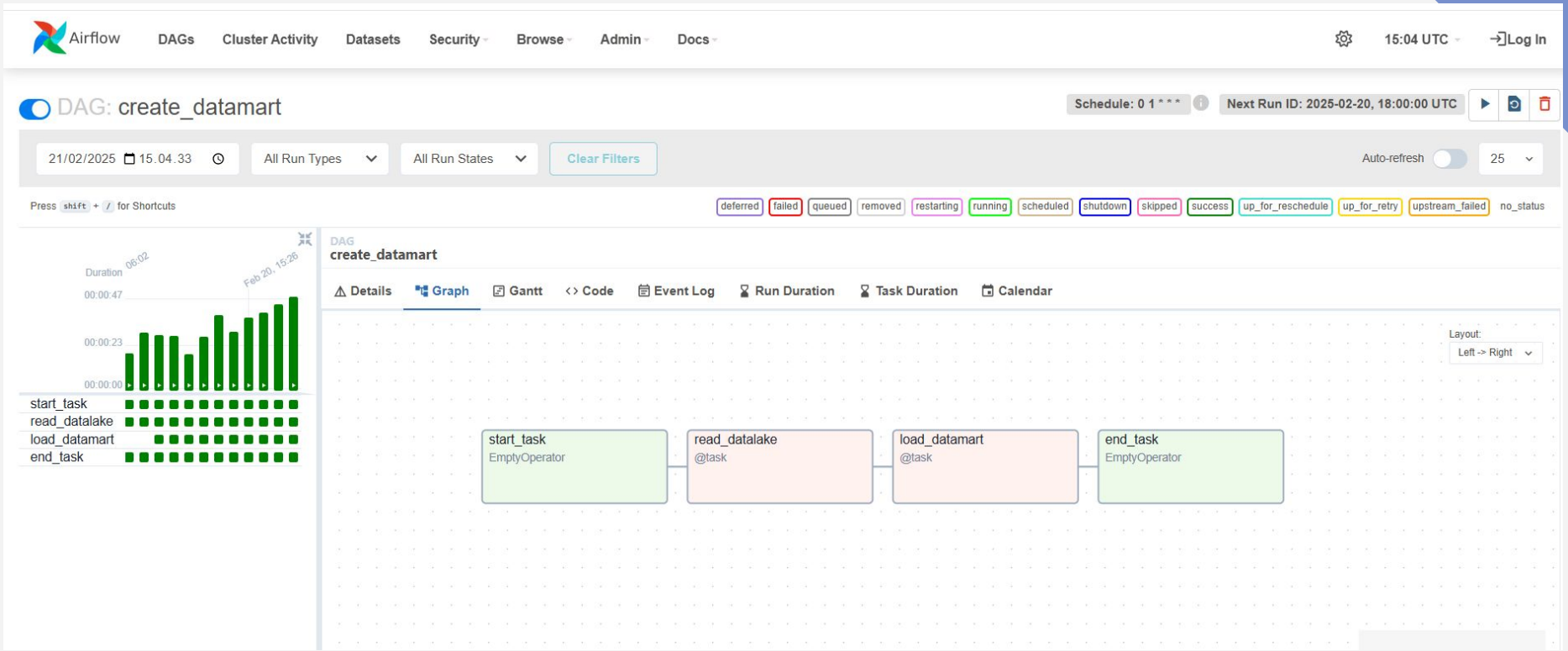
```
21  def create_datamart():
27      @task
28      def read_datalake(table_name):
29          # read postgres
30          engine = PostgresHook("de8_final_project").get_sqlalchemy_engine()
31          try:
32              # Koneksi dan query
33              with engine.connect() as connection:
34                  print("Koneksi ke Postgres berhasil!")
35
36                  query = "SELECT * FROM "+table_name
37
38                  # Menjalankan query dan mengubahnya ke dalam DataFrame
39                  df_read_datalake = pd.read_sql(query, connection)
40                  return df_read_datalake
41
42          except Exception as e:
43              print("Error koneksi:", e)
44
```

DAGs : Create Datamart

```
def create_datamart():
    def load_datamart(df_datalake, table_name):
        def dm_carmudi_mobil_dijual():
            df_result = spark.sql("select nama_mobil, count(*) jumlah from "+table_name+" group by 1 order by 2 desc")  # SQL Query
            df_result.show()
            df_result = df_result.toPandas()

            # insert to mysql datamart
            try:
                with engine.connect() as connection:
                    print("Koneksi ke MySQL berhasil!")

                print(df_result)
                df_result.to_sql("dm_carmudi_mobil_dijual", con=engine, if_exists='replace', index=False)

            except Exception as e:
                print("Error koneksi:", e)

        def dm_carmudi_mobil_avg():
            df_result = spark.sql("select concat(nama_mobil,\" (\",tahun,\")\") nama_mobil, avg(harga) avg_harga from "+table_name+" group
            df_result.show()
            df_result = df_result.toPandas()
```

LEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

DAGs : Create Datamart



Datalake Overview

```
SELECT * FROM carmudi_data_staging LIMIT 100
```

Search Results | Export | Cost: 55ms | 1 2 3 4 ... 186 | Total 18564

judul text	harga bigint	tahu text	merek text	nama_mobil text	kilometer text	transmisi text	lokasi text	dealer text
2015 Mercedes-Benz S400L 3.0 Exclusive Sedan	595000000	2015	Mercedes-Benz	Mercedes-Benz S400L	35 - 40K KM	Automatic	Banten	Deale
2018 Honda HR-V 1.5 E Special Edition SUV	253000000	2018	Honda	Honda HR-V	30 - 35K KM	Automatic	Jawa Barat	Deale
2022 Mitsubishi Xpander 1.5 Ultimate MPV - TDI	229000000	2022	Mitsubishi	Mitsubishi Xpander	30 - 35K KM	Automatic	DKI Jakarta	Deale
2014 Honda Odyssey 2.4 Prestige MPV - , UNIT I	250000000	2014	Honda	Honda Odyssey	95 - 100K KM	Automatic	DKI Jakarta	Deale
2013 Mercedes-Benz SLK250 1.8 AMG Convertib	735000000	2013	Mercedes-Benz	Mercedes-Benz SLK250	40 - 45K KM	Automatic	Banten	Deale
2008 Toyota Corolla Altis 1.8 V Sedan	119500000	2008	Toyota	Toyota Corolla Altis	124000 KM	Automatic	Jawa Barat	Deale
2018 Toyota Sienta 1.5 G MPV	179000000	2018	Toyota	Toyota Sienta	30 - 35K KM	Automatic	Yogyakarta	Deale
2011 Volkswagen Golf 1.4 TSI Hatchback - , TDP	145000000	2011	Volkswagen	Volkswagen Golf	90 - 95K KM	Automatic	DKI Jakarta	Deale
2017 Daihatsu Sigra 1.2 R Deluxe MPV	100000000	2017	Daihatsu	Daihatsu Sigra	65 - 70K KM	Automatic	DKI Jakarta	Deale

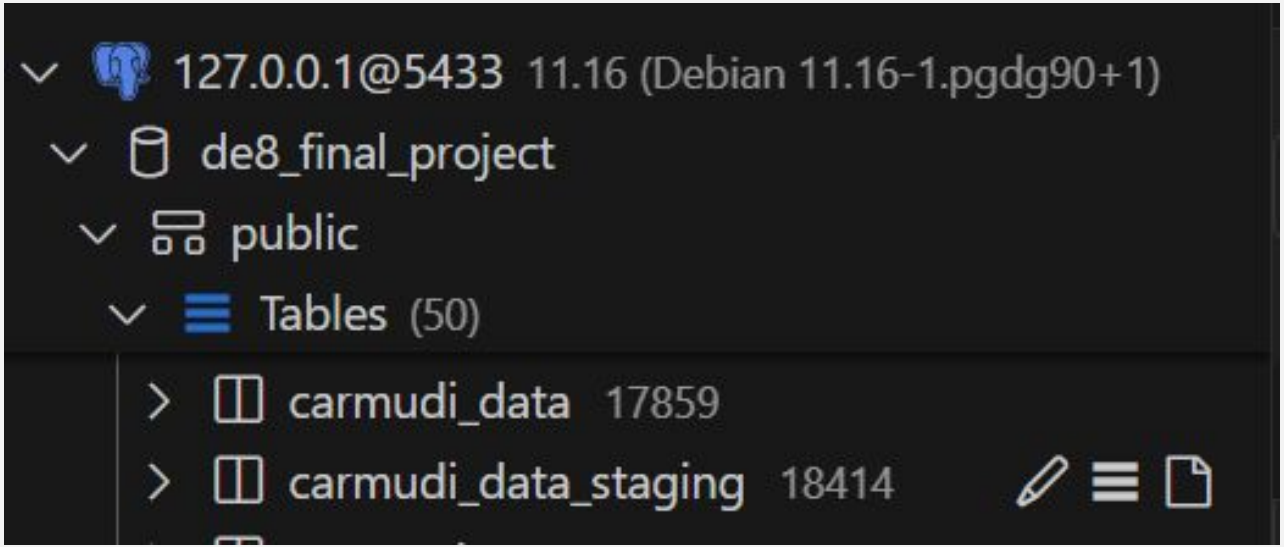
Datalake Overview

```
SELECT * FROM carmudi_data_staging LIMIT 100
```

Search Results | Export | Cost: 55ms | 1 2 3 4 ... 186 | Total 18564

merek text	nama_mobil text	kilometer text	transmisi text	lokasi text	dealer text	phone_number text	link text	snapshot_dt text
Mercedes-Benz	Mercedes-Benz S400L	35 - 40K KM	Automatic	Banten	Dealer	+6282112769067	https://www.carmudi.co.id/c	2025-02-21 13:53:55
Honda	Honda HR-V	30 - 35K KM	Automatic	Jawa Barat	Dealer	+6282112769067	https://www.carmudi.co.id/c	2025-02-21 13:53:55
Mitsubishi	Mitsubishi Xpander	30 - 35K KM	Automatic	DKI Jakarta	Dealer	+6282112769067	https://www.carmudi.co.id/c	2025-02-21 13:53:55
Honda	Honda Odyssey	95 - 100K KM	Automatic	DKI Jakarta	Dealer	+6282112769067	https://www.carmudi.co.id/c	2025-02-21 13:53:55
Mercedes-Benz	Mercedes-Benz SLK250	40 - 45K KM	Automatic	Banten	Dealer	+6282112769067	https://www.carmudi.co.id/c	2025-02-21 13:53:55
Toyota	Toyota Corolla Altis	124000 KM	Automatic	Jawa Barat	Dealer	+6282112769067	https://www.carmudi.co.id/c	2025-02-21 13:53:55
Toyota	Toyota Sienta	30 - 35K KM	Automatic	Yogyakarta	Dealer	+6282112769067	https://www.carmudi.co.id/c	2025-02-21 13:53:55
Volkswagen	Volkswagen Golf	90 - 95K KM	Automatic	DKI Jakarta	Dealer	+6282112769067	https://www.carmudi.co.id/c	2025-02-21 13:53:55
Daihatsu	Daihatsu Sigra	65 - 70K KM	Automatic	DKI Jakarta	Dealer	+6282112769067	https://www.carmudi.co.id/c	2025-02-21 13:53:55

Data Pipeline: Processing & Transformation



Data Pipeline: Processing & Transformation

```
SELECT judul, dealer, harga, kilometer, transmisi, lokasi, phone_number, count(link) total_link
FROM carmudi_data_staging
group by 1,2,3,4,5,6,7 order by 8 desc
```

Search Results | Export | Cost: 64ms | 1 2 3 4 ... 179 | Total 17864

judul text	dealer text	harga bigint	kilo text	transm text	lokasi text	phone_number text	total_link
2021 Suzuki SX4 S-Cross 1.5 Hatchback	Sales Agent	212000000	85 - 90K	Automatic	Jawa Barat	+6282112769067	5
2017 Porsche Panamera 2.9 4S Hatchback	Dealer	2288000000	5 - 10K K	Automatic	DKI Jakarta	+6282112769067	4
2019 Nissan Serena 2.0 Highway Star MPV	Dealer	290000000	35 - 40K	Automatic	Jawa Timur	+6282112769067	4
2022 Suzuki Ertiga 1.5 Hybrid GX MPV	Sales Agent	203000000	35 - 40K	Automatic	Jawa Barat	+6282112769067	4
2023 Mercedes-Benz C300 2.0 AMG Line Sedan	Dealer	935000000	0 - 5K KM	Automatic	DKI Jakarta	+6282112769067	4
2024 Toyota Kijang Innova 2.4 G MPV	Sales Agent	370000000	20 - 25K	Manual	Jawa Barat	+6282112769067	4
2020 Honda Brio 1.2 Satya E Hatchback	Dealer	135000000	50 - 55K	Automatic	DKI Jakarta	+6282112769067	4
2021 Toyota Avanza 1.5 Veloz MPV	Sales Agent	190000000	70 - 75K	Automatic	DKI Jakarta	+6282112769067	4

Data Pipeline: Processing & Transformation

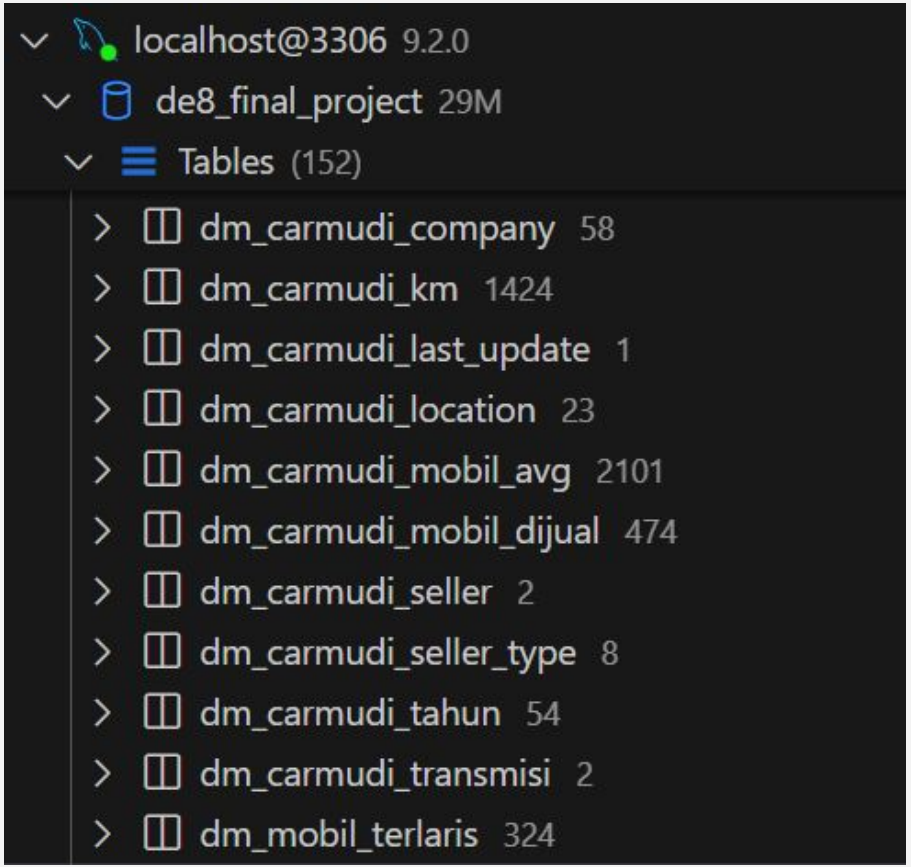
Properties | DATA | Log | ER | Monitor

```
SELECT judul, dealer, harga, kilometer, transmisi, lokasi, phone_number, count(link) total_link
FROM carmudi_data
group by 1,2,3,4,5,6,7 order by 8 desc
```

Search Results | Export | Cost: 64ms | 1 2 3 4 ... 179 | Total 17858

judul text	dealer text	harga bigint	kilometer text	transmisi text	lokasi text	phone_number text	total_link
2005 Honda CR-V 2.4 SUV	Private	87000000	135 - 140K KM	Automatic	Jawa Barat	+6282112769067	1
2009 Toyota Fortuner 2.5 G	Sales Agent	195000000	200 - 205K KM	Manual	Jawa Timur	+6282112769067	1
2021 BMW X1 1.5 sDrive18i	Dealer	535000000	40 - 45K KM	Automatic	DKI Jakarta	+6282112769067	1
2021 Mitsubishi Xpander Cr	Dealer	231000000	61284 KM	Automatic	DKI Jakarta	+6282112769067	1
2014 Lexus RX 270 2.7 SUV	Dealer	335000000	70 - 75K KM	Automatic	DKI Jakarta	+6282112769067	1
2023 Mercedes-Benz C300	Dealer	975000000	0 - 5K KM	Automatic	DKI Jakarta	(NULL)	1
2017 Honda BR-V 1.5 S SUV	Dealer	178000000	81098 KM	Manual	Jawa Tengah	+6282112769067	1
2004 Toyota Land Cruiser Cy	Dealer	425000000	140000 KM	Automatic	Sumatera Utara	+6282112769067	1

Data Pipeline: Processing & Transformation



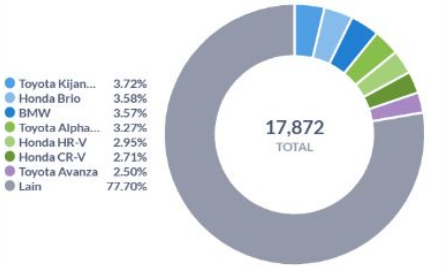
Data Visualization

Final Project DE8 - Arifin Satria Ajinusa

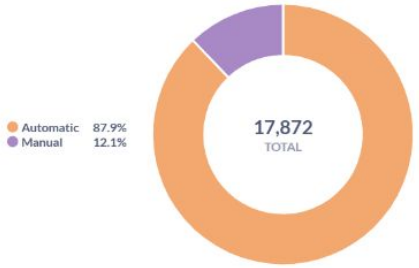
Dashboard Penjualan Mobil Bekas pada Platform Carmudi.com

2025-02-21 21:59:20.149976+07:00
Last Updated Data

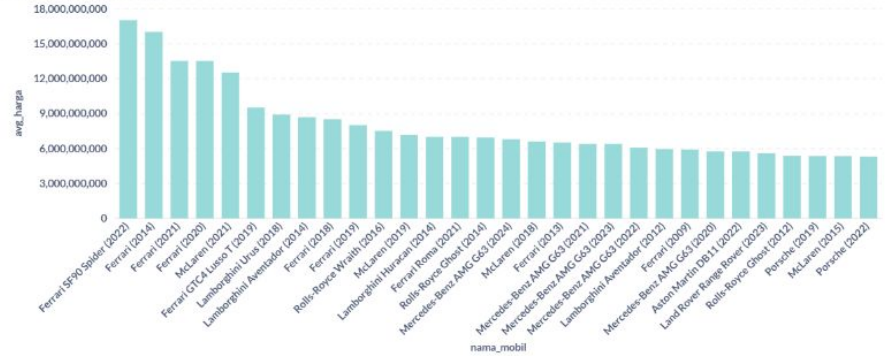
Top 10 Mobil Paling Banyak Dijual



Berdasarkan Transmisi



Top 30 Rata-rata Harga Mobil

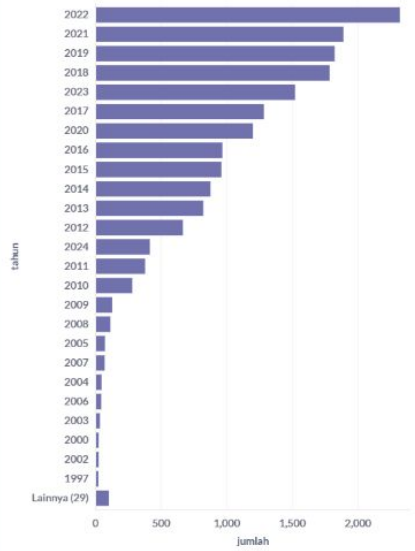


Berdasarkan Tahun Mobil

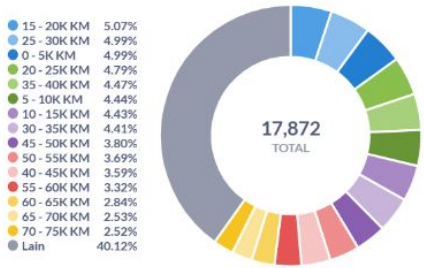
Berdasarkan KM

nama_mobil

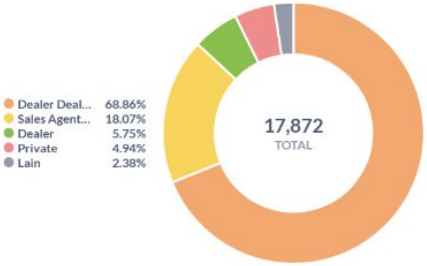
Berdasarkan Tahun Mobil



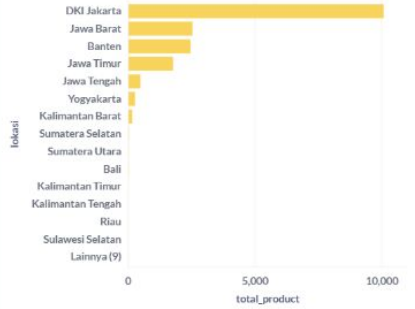
Berdasarkan KM



Berdasarkan Tipe Dealer

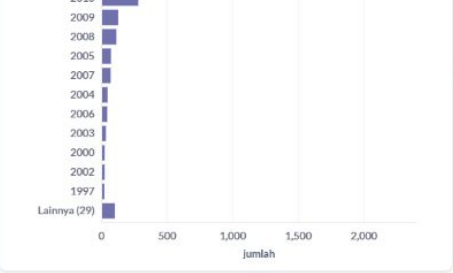


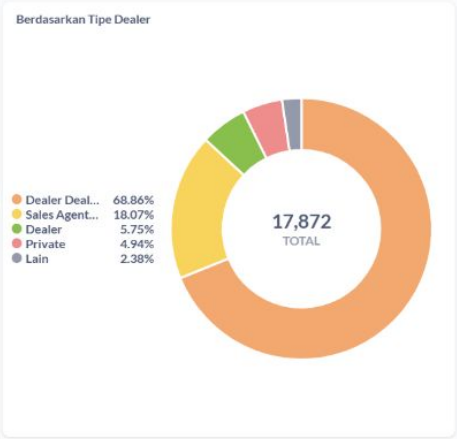
Berdasarkan Lokasi Penjual

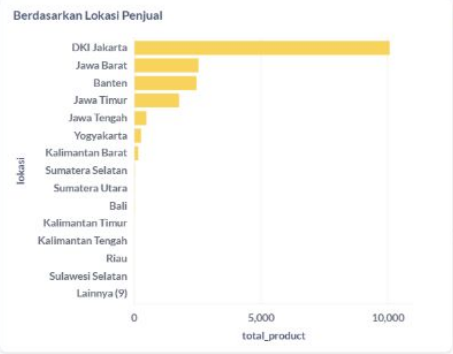


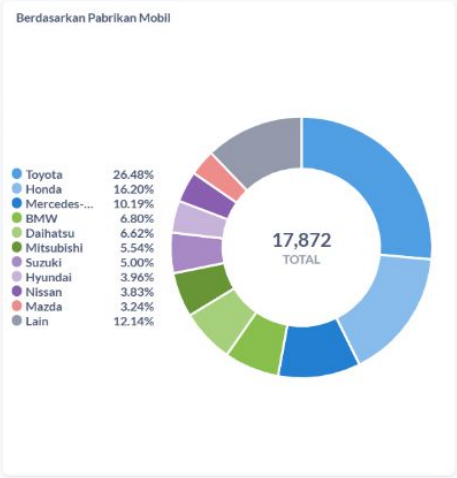
Berdasarkan Pabrikan Mobil

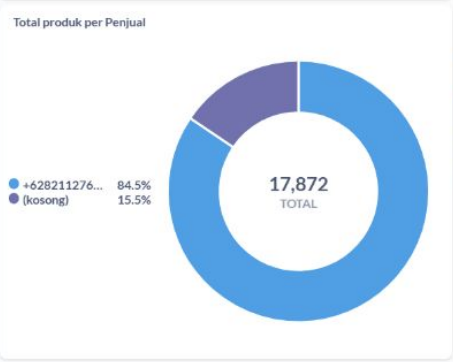
Total produk per Penjual


2009
2008
2005
2007
2004
2006
2003
2000
2002
1997
Lainnya (29)
0
500
1,000
1,500
2,000
jumlah


Berdasarkan Tipe Dealer
Dealer Deal... 68.86%
Sales Agent... 18.07%
Dealer 5.75%
Private 4.94%
Lain 2.38%
17,872
TOTAL


Berdasarkan Lokasi Penjual
DKI Jakarta
Jawa Barat
Banten
Jawa Timur
Jawa Tengah
Yogyakarta
Kalimantan Barat
Sumatera Selatan
Sumatera Utara
Bali
Kalimantan Timur
Kalimantan Tengah
Riau
Sulawesi Selatan
Lainnya (9)
lokasi
0
5,000
10,000
total_product


Berdasarkan Pabrikan Mobil
Toyota 26.48%
Honda 16.20%
Mercedes-... 10.19%
BMW 6.80%
Daihatsu 6.62%
Mitsubishi 5.54%
Suzuki 5.00%
Hyundai 3.96%
Nissan 3.83%
Mazda 3.24%
Lain 12.14%
17,872
TOTAL


Total produk per Penjual
+628211276... 84.5%
(kosong) 15.5%
17,872
TOTAL

Conclusion & Recommendation

Platform yang dibangun dalam proyek ini mampu mengolah, menyimpan, dan menyajikan data penjualan mobil bekas dari Carmudi.co.id secara otomatis dan terstruktur melalui data pipeline. Dengan kombinasi web scraping, Apache Airflow, data warehouse, dan Metabase, sistem ini dapat menyajikan dashboard interaktif yang memberikan insight tentang tren harga, distribusi merek, serta pola penjualan berdasarkan lokasi dan spesifikasi kendaraan.

Terima Kasih.