



# FINAL PROJECT

Data Warehousing Optimization

Disusun oleh Kelompok 4







# Anggota Kelompok

- Adrian Nugi Saputra – ETL
  - Adrian Saputra – ETL
  - Ananda Salsabila – Data Visualization
  - Juan Fernando – Data Visualization
  - Charunia Camila – Data Visualization
- 

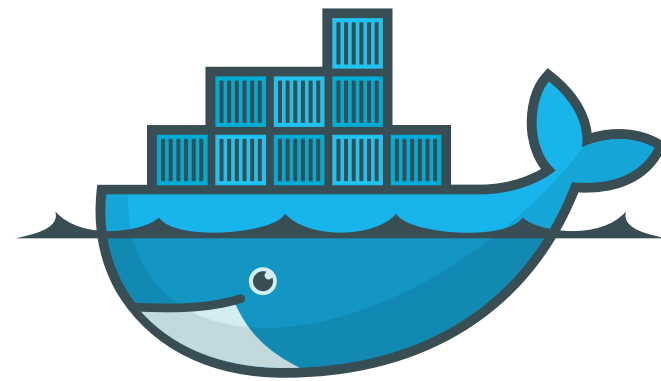


# Tujuan dan Rumusan Masalah

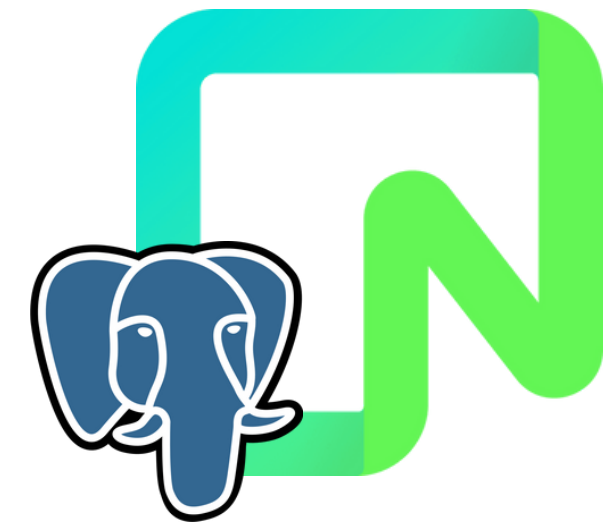


- 
- 
- 
- 
- Membangun Infrastruktur untuk pipeline data
  - Melakukan analisi data
- Tools apa yang digunakan dalam membangun infrastruktur?
  - Bagaimana proses pengolahan data dilakukan?
  - Apa yang akan dianalisis dari data tersebut?

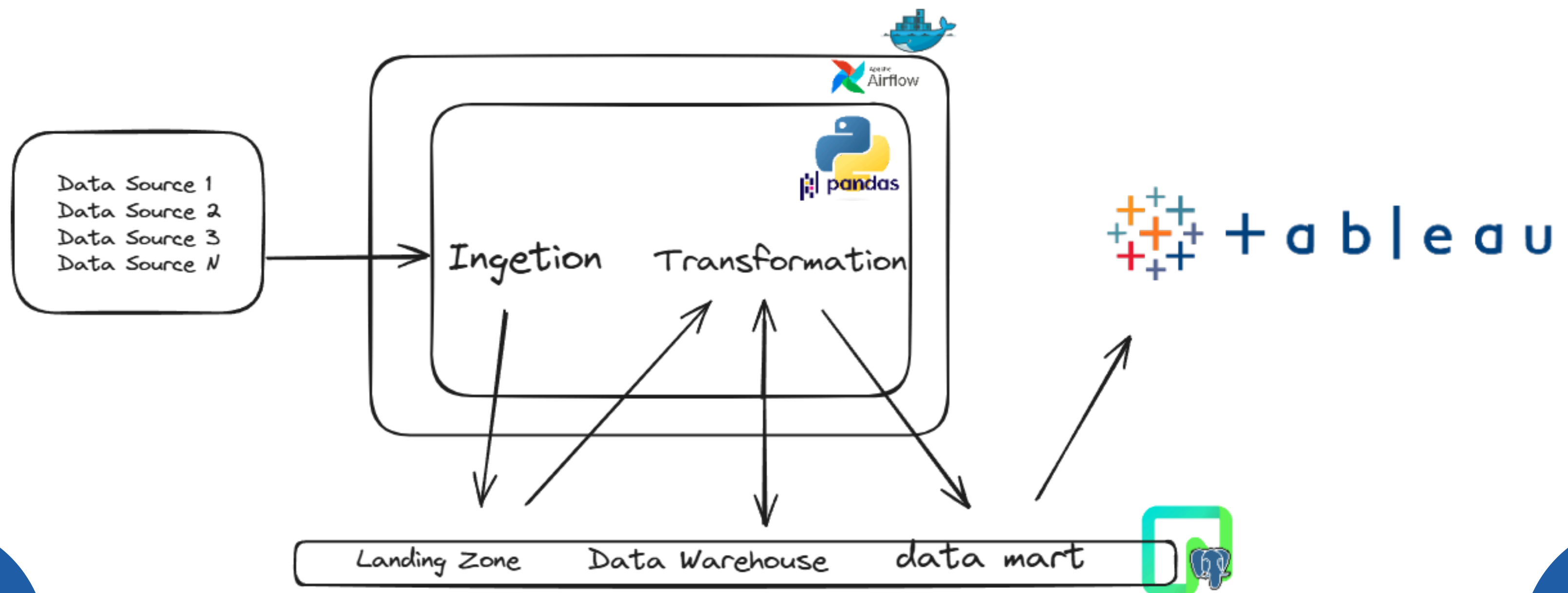
# TOOLS



Apache  
Airflow



# Arsitektur



# Ingest

- Memasukkan data dari berbagai macam format yang diberikan ke data warehouse dengan periode harian menggunakan Apache Airflow

Diharapkan untuk memasukkan 8 tabel dengan 5 format yang berbeda, yaitu

- **Coupons** dalam format *json*
- **Customers** dalam format *csv*
- **Login Attempts** dalam format *json*
- **Order Item** dalam format *avro*
- **Order** dalam format *parquet*
- **Product Category** dalam format *xls*
- **Product** dalam format *xls*
- **Supplier** dalam format *xls*

data

coupons.json  
customer\_0.csv  
customer\_1.csv  
customer\_2.csv  
customer\_3.csv  
customer\_4.csv  
customer\_5.csv  
customer\_6.csv  
customer\_7.csv  
customer\_8.csv  
customer\_9.csv

login\_attempts\_0.json  
login\_attempts\_1.json  
login\_attempts\_2.json  
login\_attempts\_3.json  
login\_attempts\_4.json  
login\_attempts\_5.json  
login\_attempts\_6.json  
login\_attempts\_7.json  
login\_attempts\_8.json  
login\_attempts\_9.json  
order.parquet  
order\_item.avro  
product.xls  
product\_category.xls  
supplier.xls

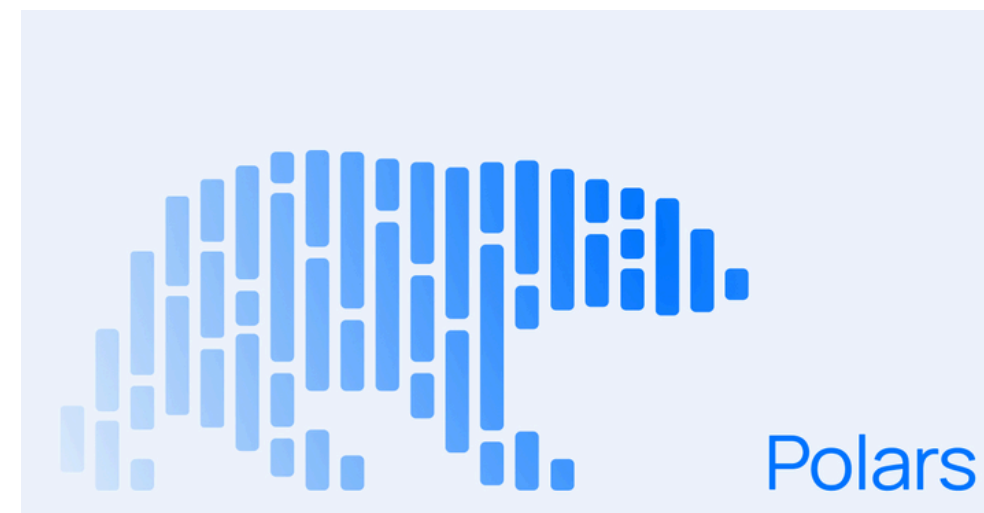


# Library

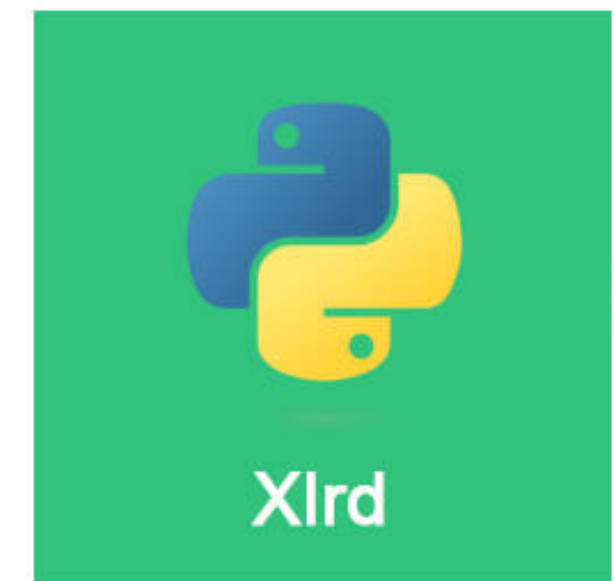
- Beberapa package yang digunakan dalam Ingestion data to landing zone



- json
- csv
- parquet



- avro



- xls

# Implementasi

- Melakukan Ingestion data to landing zone.

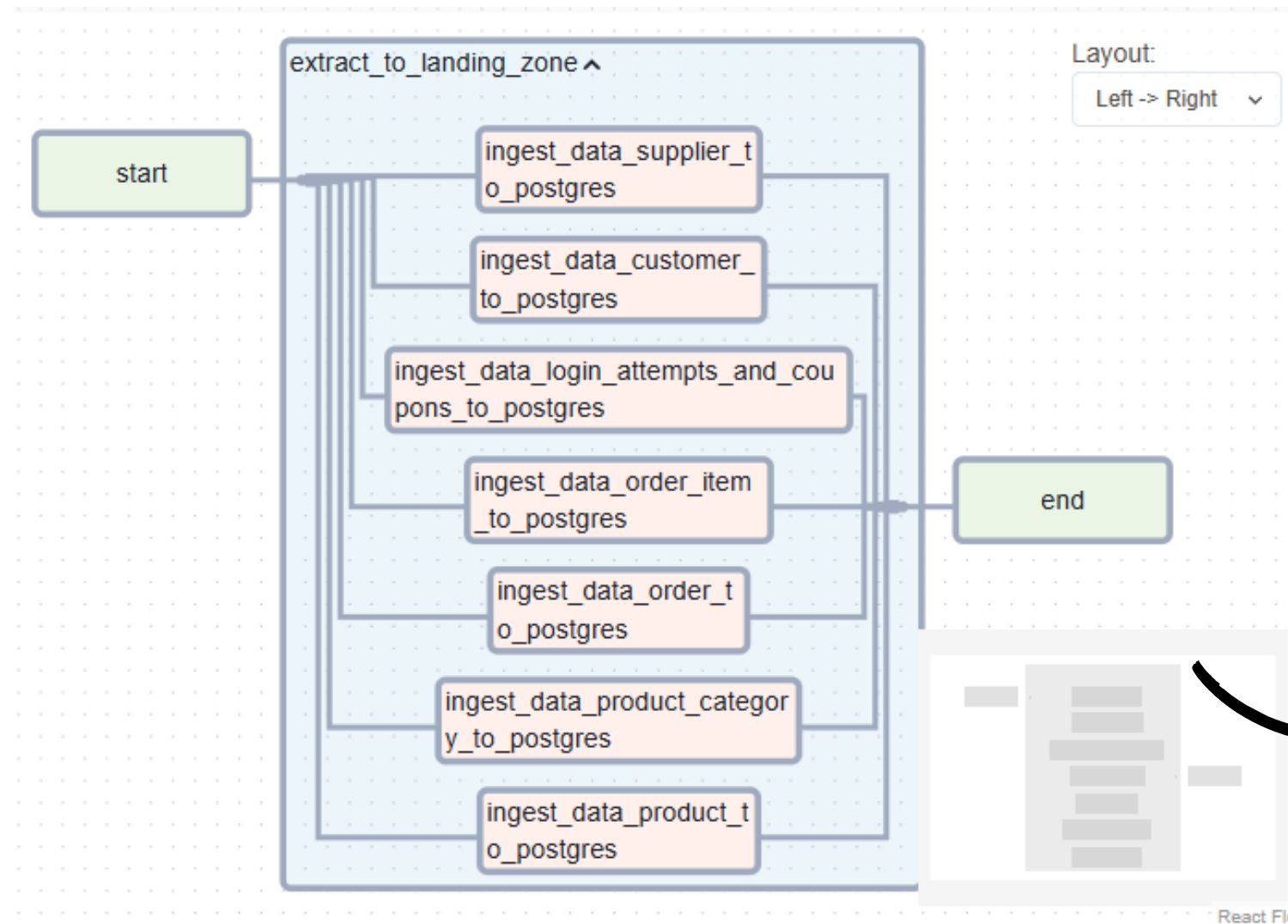
```
1 from datetime import datetime
2 import pandas as pd
3 import polars as pl
4 from airflow import DAG
5 from airflow.operators.python_operator import PythonOperator
6 from airflow.hooks.postgres_hook import PostgresHook
7
8 # Fungsi-fungsi untuk memasukkan data ke PostgreSQL (seperti yang sudah Anda definisikan)
9
10 def ingest_data_coupons_to_postgres():
11     hook = PostgresHook(postgres_conn_id="postgres_dw")
12     engine = hook.get_sqlalchemy_engine()
13     pd.read_json("data/coupons.json").to_sql("coupons", engine, if_exists="replace", index=False)
14
15 def ingest_data_customer_to_postgres():
16     hook = PostgresHook(postgres_conn_id="postgres_dw")
17     engine = hook.get_sqlalchemy_engine()
18     file_paths = [f"data/customer_{i}.csv" for i in range(10)]
19     for file_path in file_paths:
20         df = pd.read_csv(file_path)
21         df.to_sql("customer", engine, if_exists="append", index=False)
22
23 def ingest_data_login_attempts_to_postgres():
24     hook = PostgresHook(postgres_conn_id="postgres_dw")
25     engine = hook.get_sqlalchemy_engine()
26     file_paths = [f"data/login_attempts_{i}.json" for i in range(10)]
27     for file_path in file_paths:
28         df = pd.read_json(file_path)
```

```
77 # Definisikan default_args untuk DAG
78 default_args = {
79     "owner": "kelompok 4",
80     "depends_on_past": False,
81     "email_on_failure": False,
82     "email_on_retry": False,
83     # "retries": 1,
84 }
85
86 # Definisikan DAG
87 with DAG(
88     dag_id="ingestion",
89     default_args=default_args,
90     description="Ingest data to landing zone dan data warehouse postgres",
91     schedule_interval="@once",
92     start_date=make_aware(datetime.now()),
93     catchup=False,
94 ) as dag:
95     start_task = DummyOperator(task_id='start', dag=dag)
96     with TaskGroup('extract_to_landing_zone') as extract_to_landing_zone:
97         t1 = PythonOperator(
98             task_id="ingest_data_customer_to_postgres",
99             python_callable=ingest_data_customer_to_postgres,
100             dag=dag,
101         )
102         t2 = PythonOperator(
103             task_id="ingest_data_login_attempts_and_coupons_to_postgres",
104             python_callable=ingest_data_login_and_coupons_to_postgres,
105             dag=dag,
106         )
107
108     [t1, t2, t3, t4, t5, t6, t7]
109
110 end_task = DummyOperator(task_id='end', dag=dag)
111
112 # dependencies
113 start_task >> extract_to_landing_zone >> end_task
```



# DAG Ingestion

- Tampilan Ingestion di Neon.



The screenshot shows the Neon database interface. The top bar indicates the user 'Adrian Nugl Saputra' and the current database 'data\_warehouse' on the 'main' branch. The left sidebar shows navigation options: PROJECT (Dashboard, Branches, SQL Editor, Restore, Monitoring, Integrations, Settings, Quickstart), BRANCH (Overview, Tables), and RESOURCES (Feedback, Docs, Support, Changelog). The 'Tables' section is active, displaying a list of tables in the 'public' schema: coupons, customer, login\_attempts, order, order\_item, product, product\_category, and supplier. The 'coupons' table is selected, and its structure is shown in a table view with columns 'id' and 'discount\_percent'. The table contains 10 rows of data.

id	discount_percent
1	10
2	20
3	30
4	40
5	50
6	60
7	70
8	80
9	90
10	100

# Implementasi

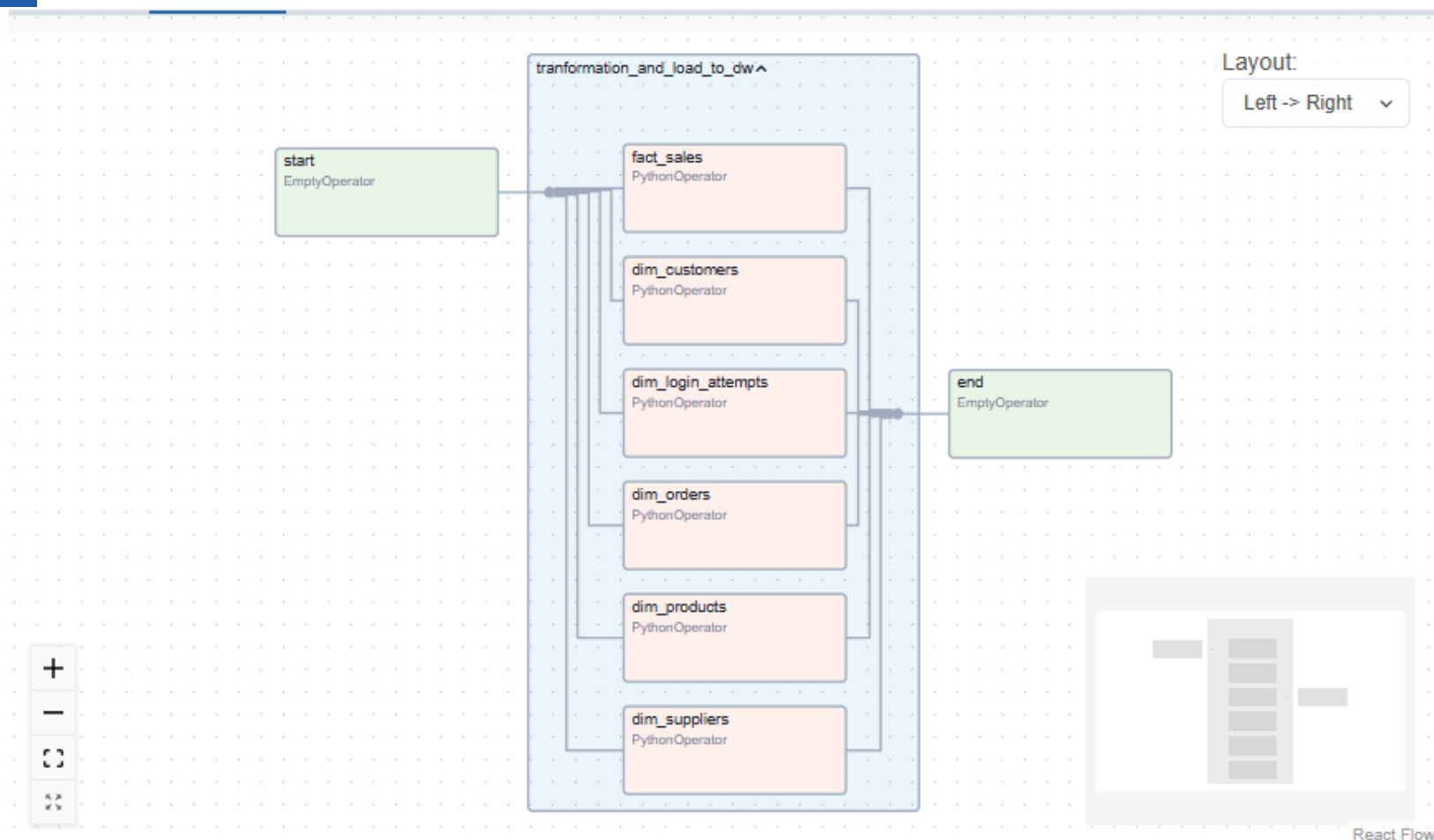
- Melakukan transformasi dan dimentional modeling ke data warehouse

```
transformation.py U x
dags > transformation.py
11
12
13 def transform_customer_data_from_landing_zone():
14     hook = PostgresHook(postgres_conn_id="postgres_dl")
15     engine = hook.get_sqlalchemy_engine()
16     hook2 = PostgresHook(postgres_conn_id="staging")
17     engine2 = hook2.get_sqlalchemy_engine()
18
19     # Define the table name pattern
20     table_prefix0 = "customer_"
21     table_prefix1 = "login_attempts_"
22     table_count = 10 # Define how many tables you expect
23
24     # Initialize an empty list to store DataFrames
25     customer_dataframes = []
26     login_dataframes = []
27
28     # Customers Table
29     for i in range(table_count):
30         table_name = f"{table_prefix0}{i}"
31         query = f"SELECT * FROM {table_name}"
32
33         # Fetch data into a Pandas DataFrame
34         df = hook.get_pandas_df(sql=query)
35
36         # Append the DataFrame to the list
37         customer_dataframes.append(df)
38
39     #login table
40     for i in range(table_count):
```

```
transformation.py U x
dags > transformation.py
76
77
78 # df_customers tranformation
79 df_customers = pd.concat(customer_dataframes, ignore_index=True)
80 df_customers.drop(columns='Unnamed: 0', inplace=True)
81 df_customers.dropna(inplace=True)
82 df_customers.drop_duplicates(inplace=True)
83 df_customers.to_sql('customers', engine2, if_exists="replace", index=False)
84
85 # df_login tranformation
86 df_login = pd.concat(login_dataframes, ignore_index=True)
87 df_login.drop(columns='Unnamed: 0', inplace=True)
88 df_login.dropna(inplace=True)
89 df_login.drop_duplicates(inplace=True)
90 df_login.to_sql('login_attempts', engine2, if_exists="replace", index=False)
91
92 # df_coupons tranformation
93 df_coupon.dropna(inplace=True)
94 df_coupon.drop_duplicates(inplace=True)
95 df_coupon.to_sql('coupons', engine2, if_exists="replace", index=False)
96
97 # df_order tranformation
98 df_order.dropna(inplace=True)
99 df_order.drop_duplicates(inplace=True)
100 df_order.to_sql('orders', engine2, if_exists="replace", index=False)
101
102 # df_order_item tranformation
103 df_order_item.dropna(inplace=True)
104 df_order_item.drop_duplicates(inplace=True)
105 df_order_item.to_sql('order_items', engine2, if_exists="replace", index=False)
```

# Implementasi

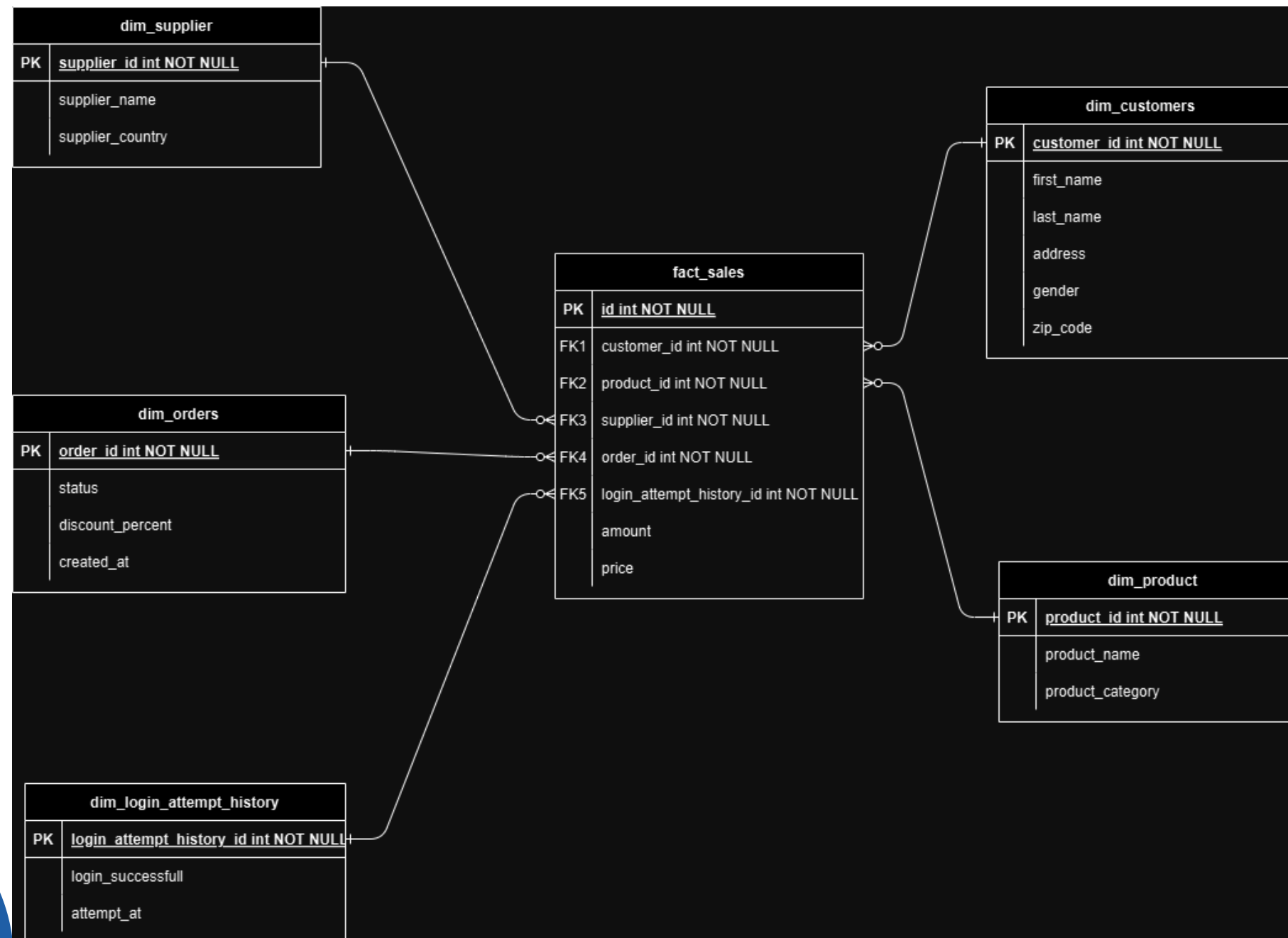
- DAG Transform And Load



Tables			<	>		Filters	Columns	Add record	50 rows • 322ms	<	50	0	>
database: dbb_project		<input type="checkbox"/>	id		first_name		last_name		gender		address		zip_code
schema: public		<input type="checkbox"/>	0		Katelyn		Hernandez		F		354 Brian Tu...		3981
Search tables		<input type="checkbox"/>	1		Timothy		Reynolds		M		854 James Da...		83703
dim_customers		<input type="checkbox"/>	2		Michelle		Mcintyre		F		0595 Pierce ...		73023
dim_login_attempts		<input type="checkbox"/>	3		Gary		Hoover		M		49129 Ward T...		57131
dim_orders		<input type="checkbox"/>	4		Laura		Mclaughlin		F		7825 Barker ...		5371
dim_products		<input type="checkbox"/>	5		Christopher		Pittman		M		396 Henson B...		84525
dim_suppliers		<input type="checkbox"/>	6		Katelyn		Hernandez		F		354 Brian Tu...		3981
fact_sales		<input type="checkbox"/>	7		Christopher		Hunt		M		897 Michael ...		66432
		<input type="checkbox"/>	8		Ann		Hobbs		F		97867 Kathry...		83615
		<input type="checkbox"/>	9		Andrew		Davis		M		61843 Brenna...		29258
		<input type="checkbox"/>	10		Melanie		Moore		F		4351 Valenti...		5079
		<input type="checkbox"/>	11		Timothy		Reynolds		M		854 James Da...		83703
		<input type="checkbox"/>	12		Michelle		Mcintyre		F		0595 Pierce ...		73023

# Implementasi

- Diagram Dimentional Modeling





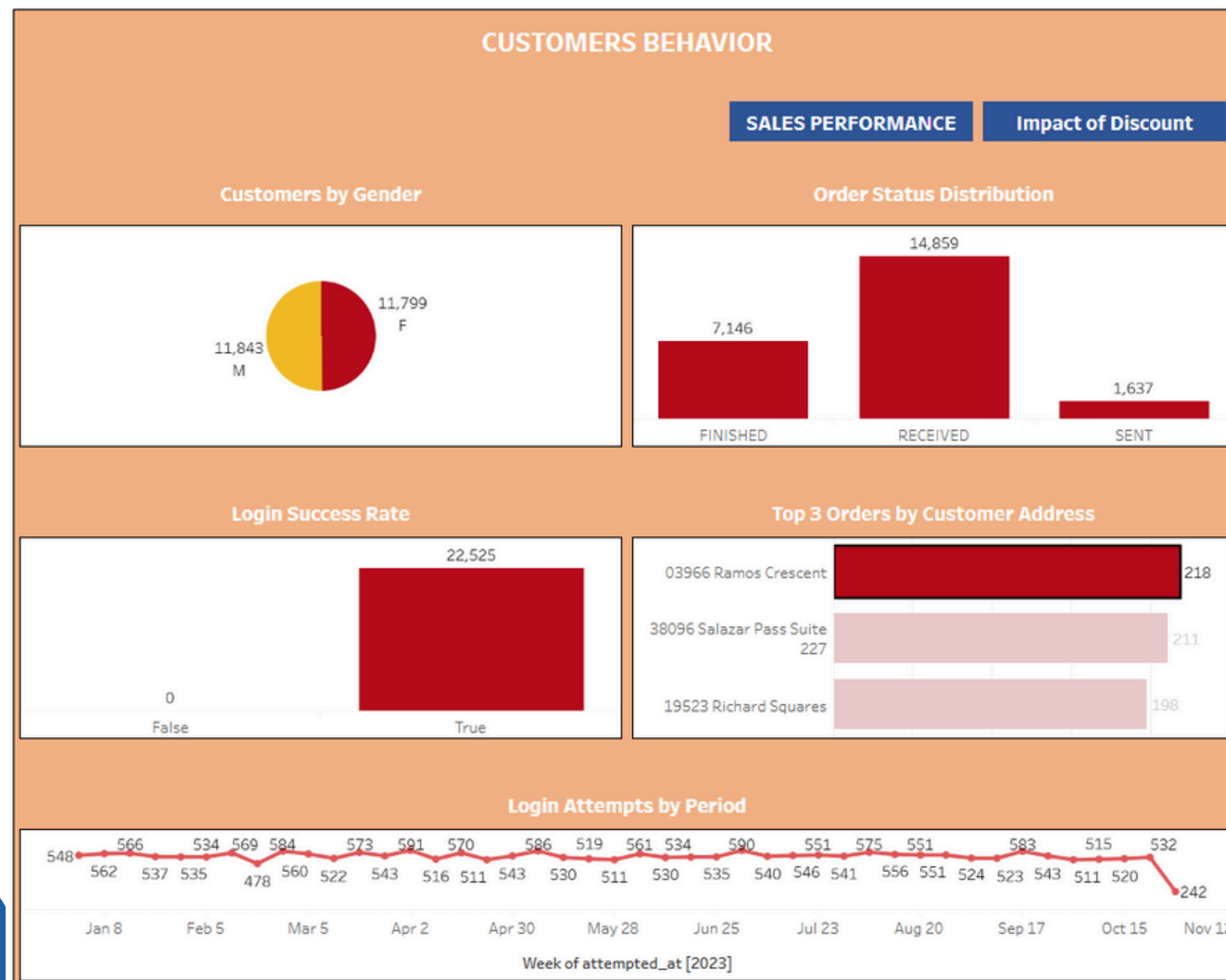
# Visualization

- Query

```
SELECT
    fs.order_date as order_date,
    dp.product_category as product_category,
    ds.supplier_name as supplier_name,
    dc.gender as customer_gender,
    dim_orders.order_status as order_status,
    fs.amount as amount,
    fs.price as price
FROM fact_sales fs
JOIN dim_products dp ON fs.product_id = dp.product_id
JOIN dim_suppliers ds ON fs.supplier_id = ds.supplier_id
JOIN dim_customers dc ON fs.customer_id = dc.id
JOIN dim_orders ON fs.order_id = dim_orders.order_id
```

# Implementasi

- Melakukan visualisasi data di Tableau



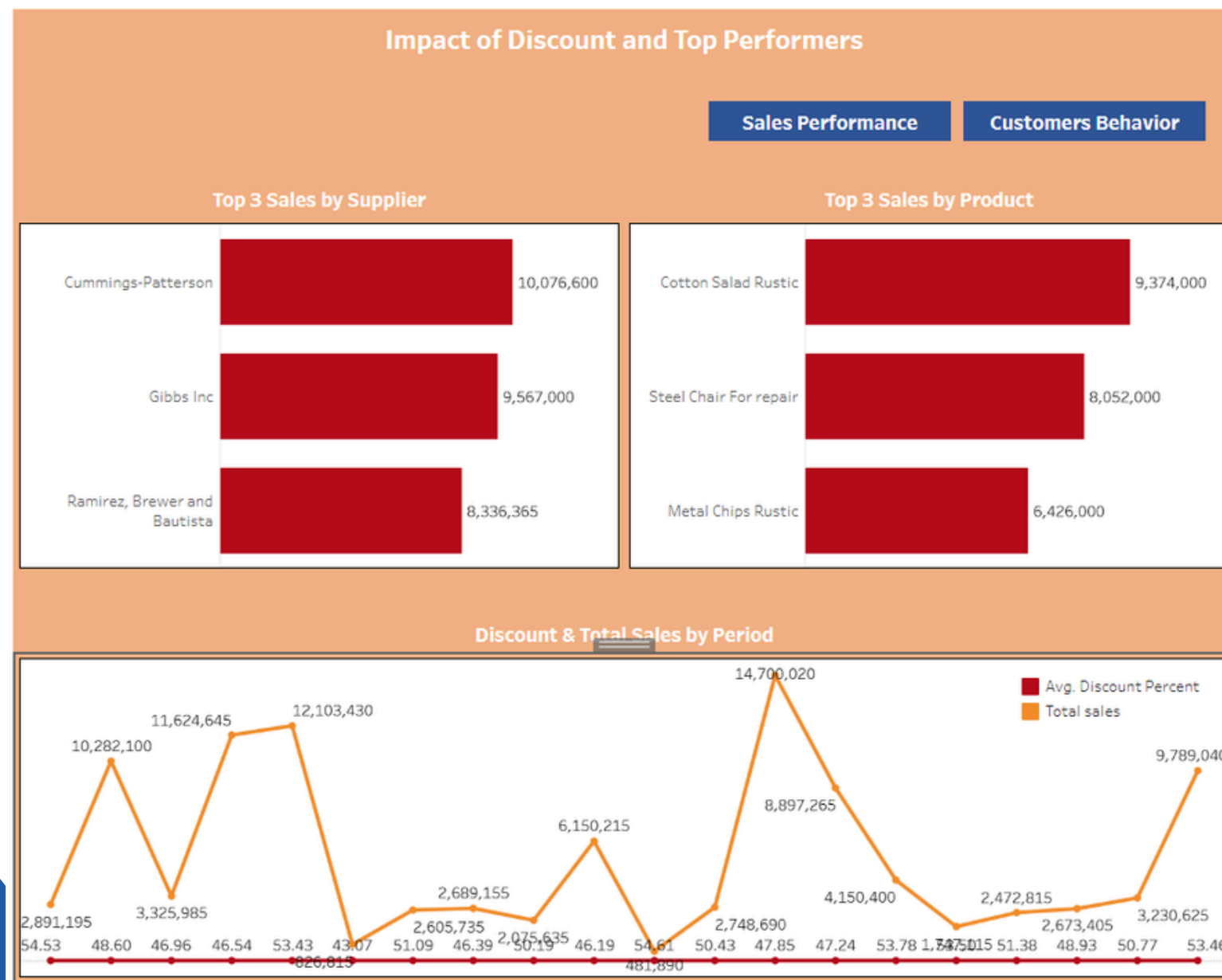
Dari dashboard diatas dapat diketahui bahwa

- Untuk tabel **Customer by Gender** diketahui bahwa pembelian terbanyak didominasi oleh laki-laki (Male).
- Untuk tabel **Order Status Distribution** diketahui bahwa produk terjual dari terbesar ke terkecil berada dalam status diterima, selesai, dalam dikirim.
- Untuk tabel **Login Success Rate** bahwa akses login yang dilakukan oleh pembeli berjalan sukses tidak pernah mengalami masalah.
- Untuk tabel **Top 3 Orders by Customer Customers Address** diberikan kode pos dari 3 orang teratas paling sering membeli produk yaitu 03966 Ramos Crescent, 38096 Salazar Pass Suite 227, dan 19523 Richard Squares.
- Untuk tabel **Login Attempts by Period** bahwa akses login yang dilakukan oleh pembeli paling banyak dilakukan sebanyak 591 kali pada 2 April 2023 dan paling sedikit dilakukan sebanyak 242 kali pada 29 Oktober 2023.



# Implementasi

- Melakukan visualisasi data di Tableau

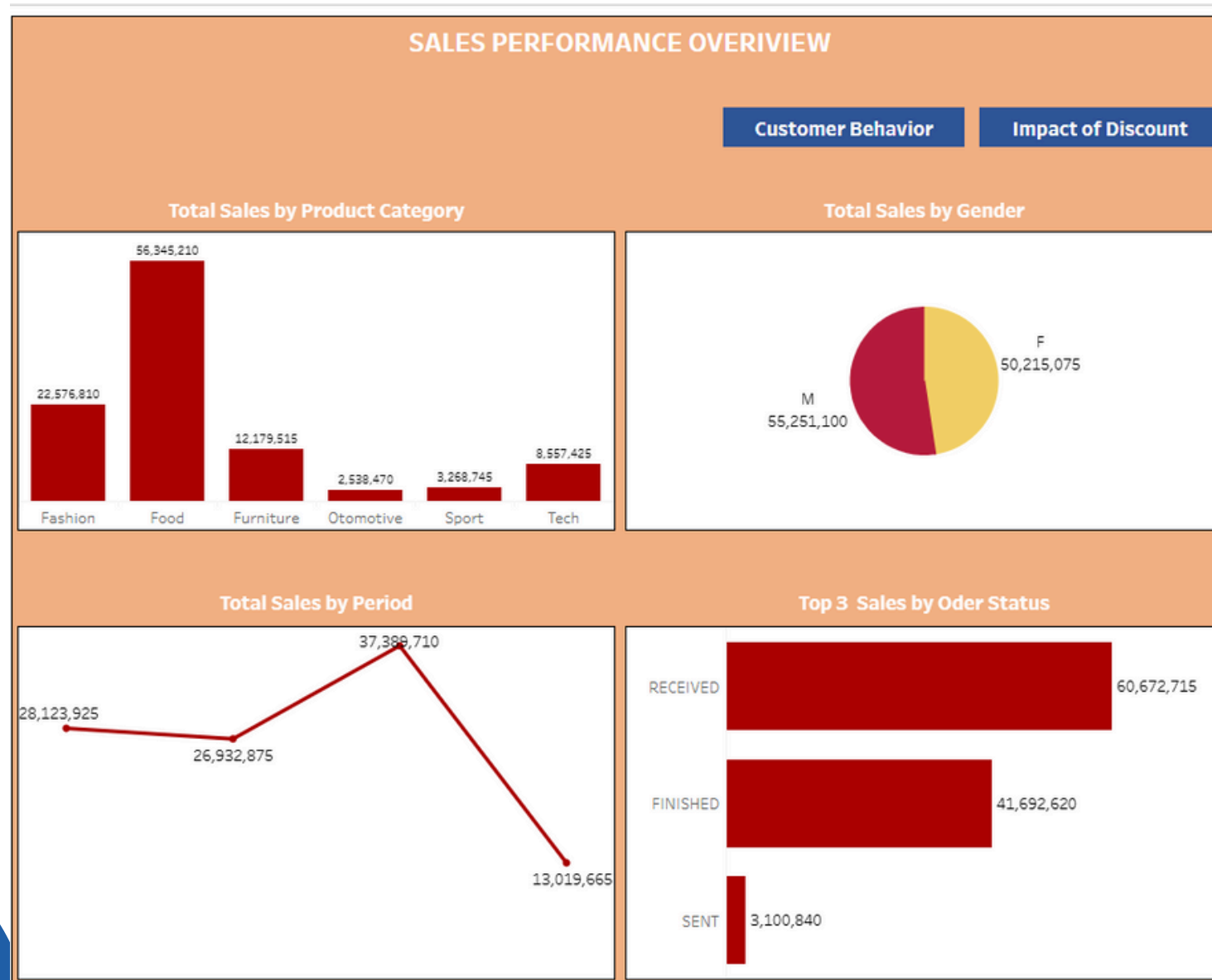


Dari dashboard diatas dapat diketahui bahwa

- Untuk tabel **Top 3 Sales by Supplier** bahwa 3 supplier terbesar dalam pengadaan produk berdasarkan penjualan secara berturut-turut dari tinggi ke rendah adalah Cumming Paterson, Gibbs inc, dan Ramirez, Brewer, and Bautista.
- Untuk tabel **Top 3 Sales by Product** bahwa 3 produk paling laris terjual secara berturut-turut dari tinggi ke rendah adalah Cotton Salad Rustic, Steel Chair for Repair, dan Metal Chips Rustic.
- Untuk tabel **Discount & Total Sales by Period** bahwa total sales paling banyak terjadi saat 13 Februari berada pada 14,700,020 sedangkan total sales paling sedikit terjadi saat 11 Februari berada pada 481.890

# Implementasi

- Melakukan visualisasi data di Tableau



Dari dashboard diatas dapat diketahui bahwa

- Untuk tabel **Total Sales by Product Category** kategori produk yang banyak terjual secara berturut-turut dari tinggi ke rendah adalah Food, Fashion, Furniture, Technology, Sport, dan Otomotif.
- Untuk tabel **Total Sales by Gender** diketahui bahwa produk terjual paling besar dilakukan oleh laki-laki (Male).
- Untuk tabel **Total Sales by Period** kategori penjualan produk berada pada titik tertinggi dan terendah secara berturut-turut pada 12 Februari dan 19 Februari.
- Untuk tabel **Total Sales by Order Status** diketahui bahwa produk terjual dari terbesar ke terkecil berada dalam status diterima, selesai, dalam dikirim.



# KENDALA & CHALLENGE



Dalam mengerjakan proyek terdapat beberapa poin yang menjadi challenge:

1. Data Modelling.
2. Menentukan visualisasi yang tepat.

Dalam mengerjakan proyek terdapat beberapa poin yang menjadi kendala:

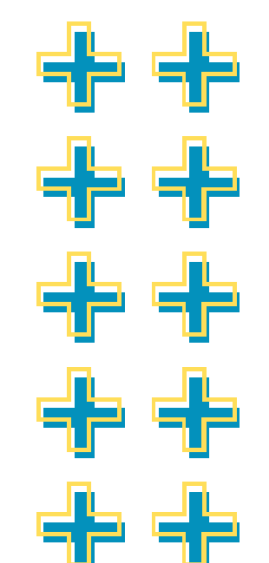
1. Environment IDX.
  2. Keterbatasan Storage Database.
- 



# IMPROVEMENT






Dalam mengerjakan proyek terdapat beberapa poin yang dapat dikembangkan:

1. Membuat Data Mart.
  2. Mengefisiensikan kode untuk mempercepat proses ETL.
  - 3 Membuat dashboard dinamis.
- 



**LINK**

- Tableau : [https://public.tableau.com/app/profile/adrian.saputra/viz/SalesPerformanceDashboard\\_17191592698690/Dashboard3?publish=yes](https://public.tableau.com/app/profile/adrian.saputra/viz/SalesPerformanceDashboard_17191592698690/Dashboard3?publish=yes)
  - Github : [https://github.com/Drians21/dbb\\_final\\_project](https://github.com/Drians21/dbb_final_project)
  - Github: <https://github.com/nugie86/dag/blob/main/dag.py>
- 
- 
- 



**TERIMA KASIH**