DOCUMENTATION

MANDIRI INHEALTH DATA ENGINEER TECHNICAL TEST

Mochammad Aditya Putra Suhendar

Data Engineer 1.3yoe at Akuntplus

LinkedIn GitHub This task REPO Email WhatsApp

1. ETL

Tools: Pentaho, MariaDB, DBeaver

Disini saya menggunakan PDI (Pentaho Data Integration), filenya dapat dilihat dengan format di atas di GitHub Repo saya dengan nama file 'mih-de-test-pdi.ktr'. For additional notes, saya menggunakan PDI versi 9.3.

EXTRACT

Saya melakukan ekstaksi data yang sudah disediakan dengan 4 CSV dan 1 JSON dengan box peruntukannya masing-masing.

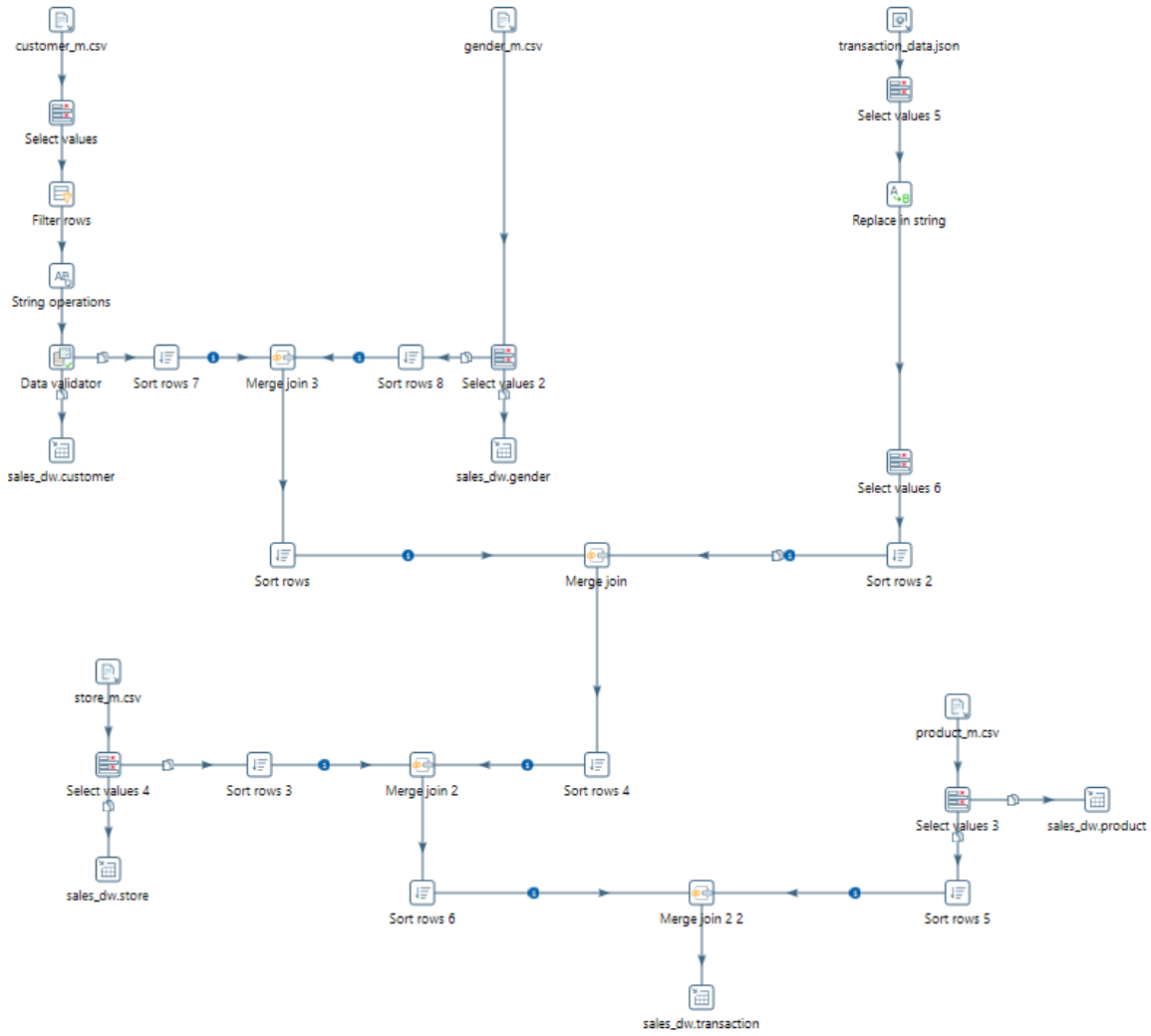
TRANSFORM

Pada proses ini saya banyak melibatkan pada perubahan nama column agar tidak pusing dan dapat membedakan mana table biasa dan transaction, pada table transaction saya mengubah nominal yang varchar karena ada symbol mata uang '$' dengan cara menghapusnya dan dijadikan number agar dapat menjawab case jika ditanyakan total pengeluaran kedepannya.

LOAD

Kemuadian step terakhir data di load ke MariaDB system localhost dengan MySQL ke setiap-setiap table masing-masing. Kecuali table transaction harus dijoinkan terlebih dahulu.

Berikut adalah gambaran kasar ETL yang sudah saya bangun:



Sebelum dijalankan saya membuat databasenya terlebih dahulu dan juga table-tablenya agar sinkron antara keinginan, petunjuk dan juga default data type yang dibaca oleh Pentaho.

```
create database sales_dw;
use sales_dw;
```

Membuat table gender:

```
     create table gender (
       GenderID int(11) not null primary key,
       Gender varchar(10),
);
```

Membuat table customer:

```
     create table customer (
       CustomerID int(11) not null primary key,
```

```
        GenderID int(11),
        FirstName varchar(20),
        LastName varchar(20),
        Email varchar(28),
        foreign key (GenderID) references gender
      (GenderID)
);
```

Membuat table store:

```
      create table store (
        StoreID int(11) not null primary key,
        StoreName varchar(100),
        StoreLocation varchar(20)
);
```

Membuat table product:

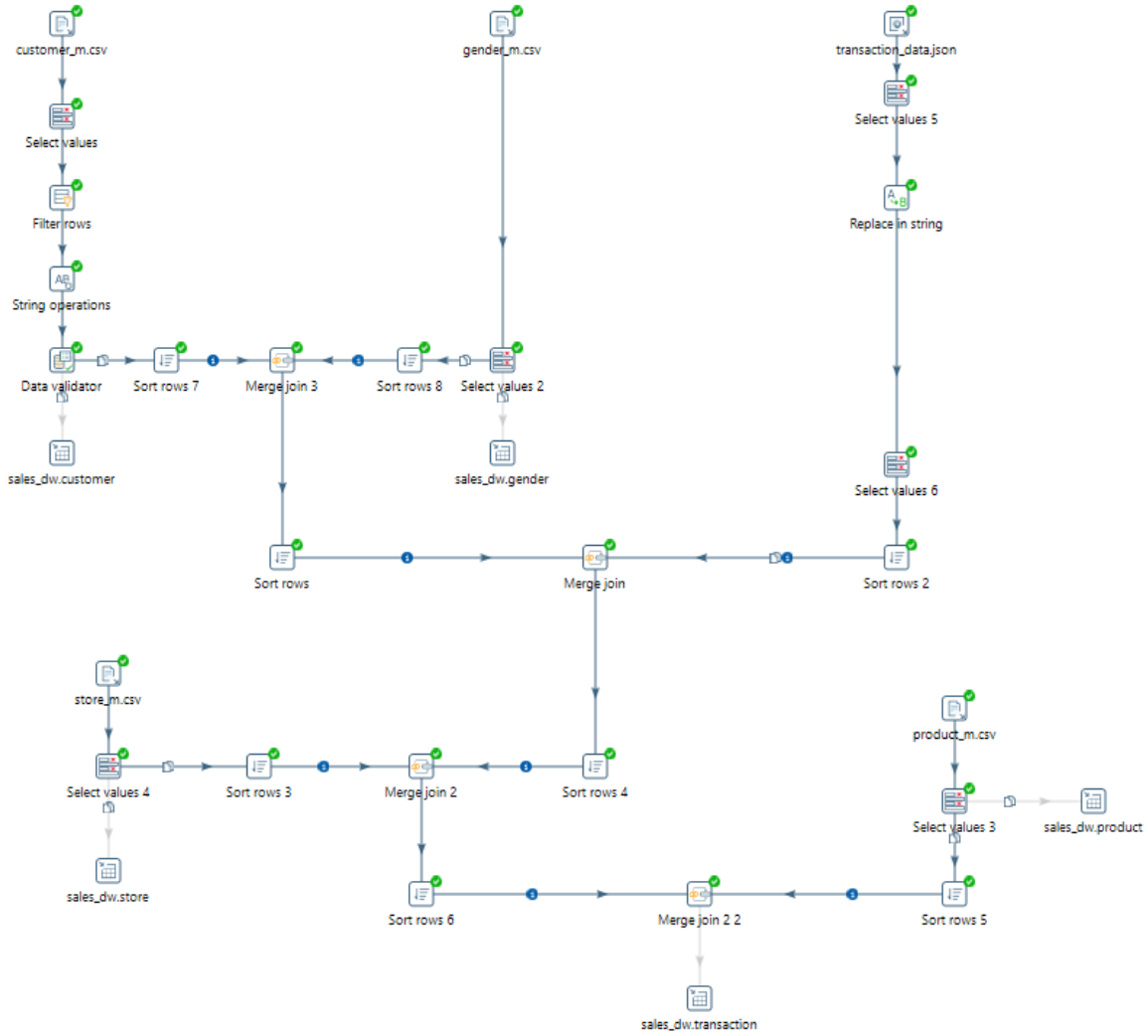
```
      create table product (
        ProductID int(11) not null primary key,
        ProductName varchar(50),
);
```

Membuat table transaction:

```
      create table transaction (
        trx_id uuid not null primary key,
        customer_id int(11),
        product_id int(11),
        store_id int(11),
        item_quantity int(11),
        item_price decimal(10,2),
        trx_date date,
        foreign key (customer_id) references customer
      (CustomerID),
        foreign key (product_id) references product
      (ProductID),
        foreign key (store_id) references store (StoreID)
);
```

Baru setelah itu saya jalankan ETL di Pentaho yang hasilkan akan seperti dibawah ini:

DISCLAIMER: Mohon untuk diabaikan bagian untuk Loadnya tidak saya nyalakan karena sudah saya jalankan pada saat awal namun tidak terdokumentasikan, tapi yang pasti berhasil jika ingin dikosongkan terlebih dahulu table-tablenya.



CASE 1-6

Tools: DBeaver

 # CASE 1

 # Menampilkan 5 nama lengkap customer dengan pengeluaran paling banyak, urutan dari terbesar ke terkecil

```
select c.CustomerID,
    concat(c.FirstName, ' ', c.LastName) as FullName,
    sum(t.item_quantity*t.item_price) as TotalExpenses
from customer c
join transaction t on c.CustomerID = t.customer_id
group by c.CustomerID
order by TotalExpenses desc
limit 5;
```

	CustomerID	FullName	TotalExpenses
1	25	Otis Spere	2,166.41
2	10	Vinnie McCarron	2,119.6
3	14	Constantino Coton	2,035.09
4	40	Mureil Cansdall	1,933.06
5	12	Audrie Tunesi	1,708.32

 # CASE 2

 # Menampilkan 5 kota dengan total item terjual paling banyak, urutan dari terbesar ke terkecil

```
select s.StoreID,
    s.StoreLocation,
    sum(t.item_quantity) as TotalItemSold
from store s
join transaction t on t.store_id = s.StoreID
group by s.StoreID
order by TotalItemSold desc
limit 5;
```

	StoreID	StoreLocation	TotalItemSold
1	1	Aimin	84
2	25	Songnim	73
3	15	Malim	64
4	20	Yanaoca	60
5	24	Melchor Ocampo	58

 # CASE 3

 # Menampilkan 5 product yang paling banyak dibeli oleh laki laki, urutan dari terkecil ke terbesar

```sql
select p.ProductID,
    p.ProductName,
    sum(t.item_quantity) as Total_Item_Sold_by_Ma
from product p
join transaction t on t.product_id = p.ProductID
join customer c on t.customer_id = c.CustomerID
join gender g  on g.GenderID = c.GenderID
where g.gender = 'Male'
group by p.ProductName
order by Total_Item_Sold_by_Male desc
limit 5;
```

	ProductID	ProductName	Total_Item_Sold_by_M
1	8	Vermacelli - Sprinkles, Assorted	
2	2	Versatainer Nc - 9388	
3	3	Chinese Foods - Cantonese	
4	7	Plasticknivesblack	
5	5	Arizona - Green Tea	

 # CASE 4

 # Menampilkan 5 toko yang menjual product paling banyak pada bulan januari (akumulasi penjualan produk yang terjual hanya di bulan januari di semua tahun) , urutan dari terbesar ke terkecil

```sql
select s.StoreID,
    s.StoreName,
    sum(t.item_quantity) as Store_with_Most_Sales_Quantity_Ja
from store s
join transaction t on s.StoreID = t.store_id
where extract(month from t.trx_date) = 1
group by s.StoreName
order by Store_with_Most_Sales_Quantity_January desc
limit 5;
```

	StoreID	StoreName	Store_with_Most_Sales_Quantity_Janu
1	18	REMEDYREPACK INC.	
2	20	McKesson	
3	15	Reckitt Benckiser LLC	
4	11	Uriel Pharmacy Inc.	
5	17	L. Perrigo Company	

 # CASE 5

 # Menampilkan 5 email customer yang berasal dari pemerintahan (.gov) yang memiliki total pembelian terbesar
 # di kota yang paling banyak menjual product (total item quantity) paling banyak kepada pembeli perempuan.
 # HINT : cari terlebih dahulu kota apa yang paling banyak menjual barang untuk perempuan.
 # kemudian di kota tersebut cari 5 customer dari pemerintahan yang melakukan pembelian terbesar

```sql
# Cari kota dengan total penjualan terbanyak kepada perempuan
select s.StoreLocation, # Harusnya ini bisa pake CTE tapi MySQL disini bisanya pakai subqueries
    sum(t.item_quantity) as Total_Item_Sold_by_Female
from store s
join transaction t on t.store_id = s.StoreID
join customer c on c.CustomerID = t.customer_id
join gender g on g.GenderID = c.GenderID
where g.Gender = 'Female'
group by s.StoreLocation
order by Total_Item_Sold_by_Female desc
limit 1

# Cari 5 customer dari pemerintah dengan pembelian terbesar di kota tersebut
select c.CustomerID,
       c.Email,
       s.StoreLocation,
       sum(t.item_price*t.item_quantity) as TotalExpenses
from customer c
join transaction t on t.customer_id = c.CustomerID
join store s on s.StoreID = t.store_id
where s.StoreLocation = (
    select s.StoreLocation # Subqueries
    from store s
    join transaction t on t.store_id = s.StoreID
    join customer c on c.CustomerID = t.customer_id
    join gender g on g.GenderID = c.GenderID
    where g.Gender = 'Female'
    group by s.StoreLocation
    order by sum(t.item_quantity) desc
    limit 1
)
and c.Email like '%.gov%'
group by c.CustomerID, c.Email
order by TotalExpenses desc
limit 5;
```

	CustomerID	Email	StoreLocation	TotalExper
1	46	pmendoza19@ed.gov	Aimin	
2	10	vmccarron9@census.gov	Aimin	
3	44	bmcclintock17@epa.gov	Aimin	
4	40	mcansdall13@usa.gov	Aimin	
5	15	vemenye@nsw.gov.au	Aimin	

 # CASE 6
 # Menampilkan 5 nama lengkap customer dengan pengeluaran paling banyak,
 # disertai dengan informasi produk apa saja yang pernah dibeli (unique),
 # dan informasi toko mana saja tempat produk tersebut dijual (unique)

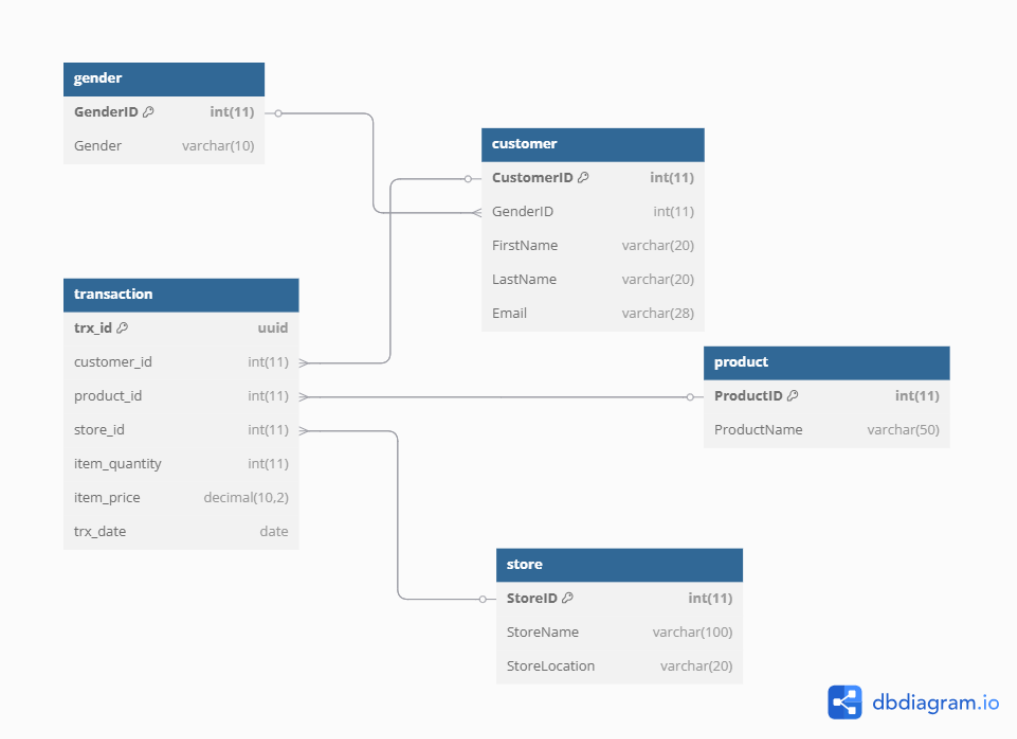
```sql
select concat(c.FirstName, ' ', c.LastName) as FullName,
    sum(t.item_quantity*t.item_price) as TotalExpenses,
    group_concat(distinct p.ProductName) as PurchasedProd
    group_concat(distinct s.StoreName) as StoreLocations
from customer c
join transaction t on c.CustomerID = t.customer_id
join product p on p.ProductID = t.product_id
join store s on s.StoreID = t.store_id
group by c.CustomerID , FullName
order by TotalExpenses desc
limit 5
```

	FullName	TotalExpenses	PurchasedProducts	StoreLocations
1	Otis Spere	2,166.41	Arizona - Green Tea,Chinese Foo	Golden State Medical
2	Vinnie McCarron	2,119.6	Arizona - Green Tea,Chinese Foo	Antigen Laboratories,
3	Constantino Coton	2,035.09	Chinese Foods - Cantonese,Pork	Allermed Laboratories
4	Mureil Cansdall	1,933.06	Arizona - Green Tea,Plasticknives	Allermed Laboratories
5	Audrie Tunesi	1,708.32	Arizona - Green Tea,Plasticknives	Antigen Laboratories,

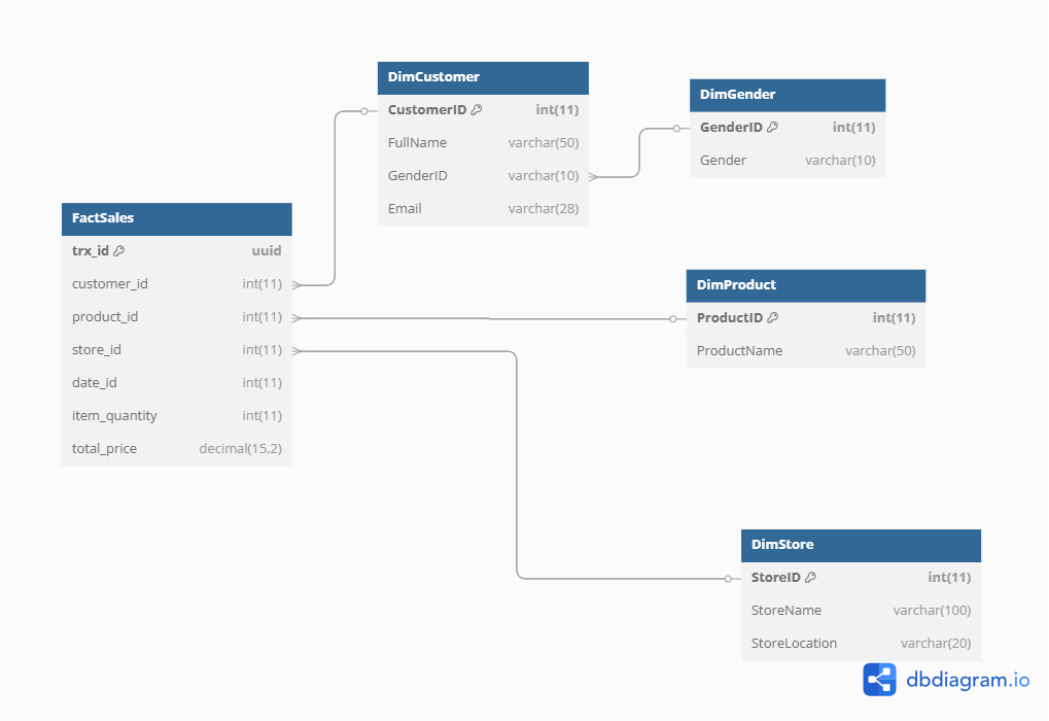
ERD OLTP & OLAP

Tools: dbdiagram.io

Saya menggunakan dbdiagram.io untuk membuat ERD OLTP dan OLAP model, berikut adalah ERD OLTP:



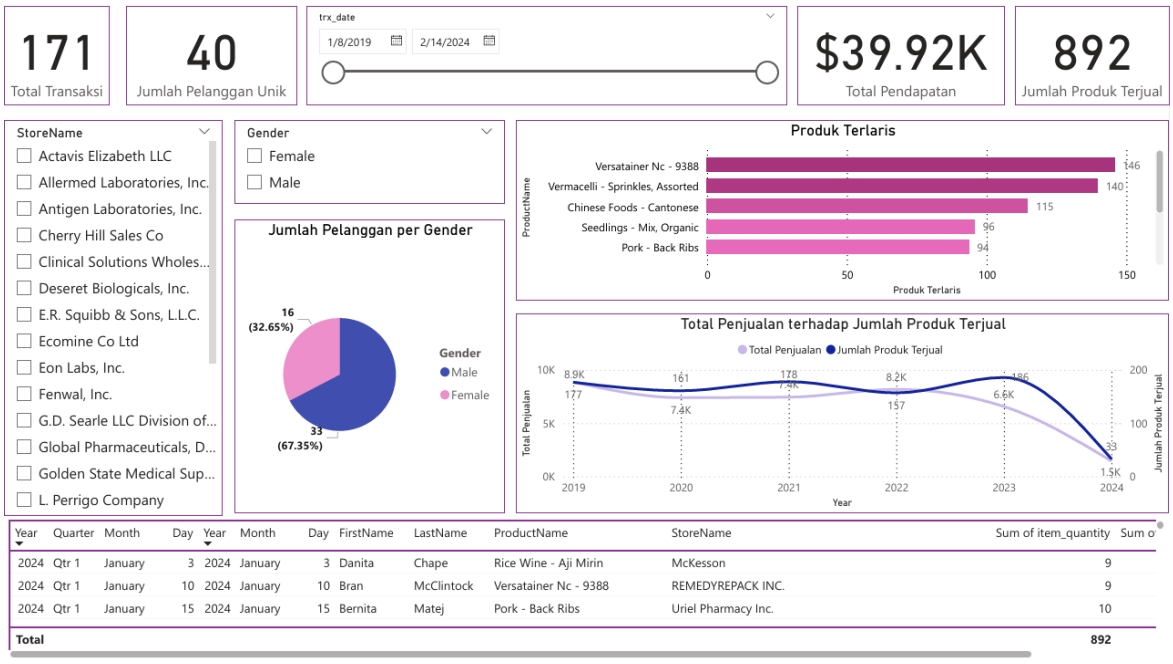
Berikut adalah model OLAP berbentuk star schema:



VISUALIZATION

Tools: PowerBI

Saya menggunakan PowerBI untuk melakukan visualization dengan beberapa charts, cards, dan juga filter slicers

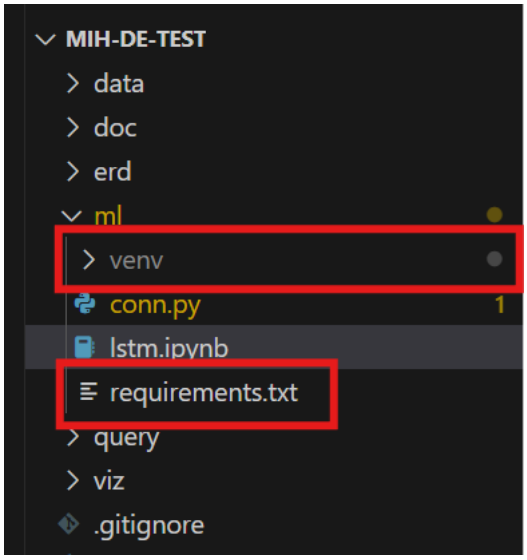


SALES PREDICTION USING LSTM

Tools: VSC

Pada tahap ini saya mencoba untuk membangun model LSTM dengan tujuan untuk memprediksi penjualan dari data transaction selama 30 hari kedepan, semua progress dan perkembangan dapat dilihat pada GitHub REPO.

1. Pembuatan VENV, .gitignore dan Install requirements.txt



Pembuatan cirtual environtment (venv) ditujukan untuk membuat suatu lingkungan dengan library dependencies untuk suatu project. Kemudian venv dimasukan kedalam .gitignore agar tidak ke push ke repo karena size yang besar. Requirements.txt berisi library yang perlu diinstall dan nantinya akan masuk kedalam venv yang sudah dibuat.

2. Pembuatan conn.py

```
ml > conn.py > ...
import os
import mariadb
import pandas as pd

# Konfigurasi koneksi ke MariaDB
# Untuk keamanan sebenarnya bisa menggunakan JSON dan dimasukan ke .gitignore, namun
db_config = {
    "host": "localhost",
    "port": 3306,
    "user": "root",
    "password": "password.1",
    "database": "sales_dw"
}

# Path folder tujuan
save_path = r"C:\Users\HP\Documents\Belajar Adit\mih-de-test\data"
file_name = "transaction.csv"
full_path = os.path.join(save_path, file_name)

try:
    # Membuat koneksi ke MariaDB
    conn = mariadb.connect(**db_config)
    cursor = conn.cursor()

    # Mengeksekusi query untuk mengambil data dari tabel transaction
    table_name = "transaction"
    query = f"SELECT * FROM {table_name}"

    # Membaca tabel ke dalam DataFrame
    df = pd.read_sql_query(query, conn)

    # Menyimpan ke CSV di folder yang ditentukan
    df.to_csv(full_path, index=False, encoding="utf-8")
```

Pembuatan file ini dimaksudkan untuk mendapatkan connection dari dan ke MariaDB untuk merubah table dan datanya menjadi csv yang kemudian diimport ke local. Disini saya tidak menggunaka JSON untuk credentialsnya karena menggunakan localhost, jika mengunakan database server diwajibkan untuk menggunakan JSON dan dimasukan kedalam .gitignore agar tidak dapat diakses sembarang orang.

3. **LSTM.ipynb**

Karena keterbatasan waktu dokumen ini terlampir di GitHub REPO. **Masih banyak yang perlu diperbaiki dan on progress, akan tetap saya lanjutkan untuk progress ini sementara dokumen ini saya kumpul terlebih dahulu sebelum 26/03/2025 16.00 WIB.**