

Data Modeling

Kalbe Nutritional Data Scientist Virtual
Internship Program Batch September 2023

Presented by
Limatan Luviar

BACKGROUND STORY

Kamu adalah seorang Data Scientist di Kalbe Nutritionals dan sedang mendapatkan project baru dari tim inventory dan tim marketing. Dari tim inventory, kamu diminta untuk dapat membantu memprediksi jumlah penjualan (quantity) dari total keseluruhan product Kalbe

- Tujuan dari project ini adalah untuk mengetahui perkiraan quantity product yang terjual sehingga tim inventory dapat membuat stock persediaan harian yang cukup.
- Prediksi yang dilakukan harus harian.

Dari tim marketing kamu diminta untuk membuat cluster/segment customer berdasarkan beberapa kriteria.

- Tujuan dari project ini adalah untuk membuat segment customer.
- Segment customer ini nantinya akan digunakan oleh tim marketing untuk memberikan personalized promotion dan sales treatment.

EXPLORATORY DATA ANALYSIS

1. Berapa rata-rata umur customer jika dilihat dari marital statusnya

Input Query

```
select ("Marital Status") as marital_status,  
round(avg(age),2) as avg_age  
from pbi_ds.customer  
where "Marital Status" != ''  
group by "Marital Status"  
order by avg_age desc
```

Output Query

ABC marital_status ▼	123 avg_age ▼
Married	43.04
Single	29.38

EXPLORATORY DATA ANALYSIS

2. Berapa rata-rata umur customer jika dilihat dari gender nya ?

Input Query

```
select gender as gender,  
round(avg(age),2) as avg_gender_age  
from pbi_ds.customer  
group by gender  
order by avg_gender_age desc
```

Output Query

Grid		123 gender ▼	123 avg_gender_age ▼
	1	0	40.33
ext	2	1	39.14

EXPLORATORY DATA ANALYSIS

3. Tentukan nama store dengan total quantity terbanyak!

Input Query

```
select storename sc, sum(t.qty) as total_qty  
from pbi_ds.store s  
join pbi_ds.transaction t on s.storeid = t.storeid  
group by s.storename  
order by total_qty desc  
limit 1;
```

Output Query

	ABC sc ▼	123 total_qty ▼	
1	Lingga	2,777	

EXPLORATORY DATA ANALYSIS

4. Tentukan nama produk terlaris dengan total amount terbanyak!

Input Query

```
select ("Product Name") pn, sum(t.totalamount) as total_produk_terlaris
from pbi_ds.product p
join pbi_ds."transaction" t on p.productid = t.productid
group by p."Product Name"
order by total_produk_terlaris desc
limit 1;
```





Output Query

ABC pn	123 total_produk_terlaris
Cheese Stick	27,615,000

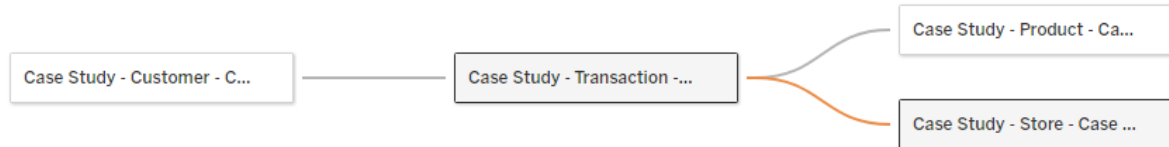
DATA INGESTION KE DALAM TABLEAU PUBLIC

Import table ke Tableau Public

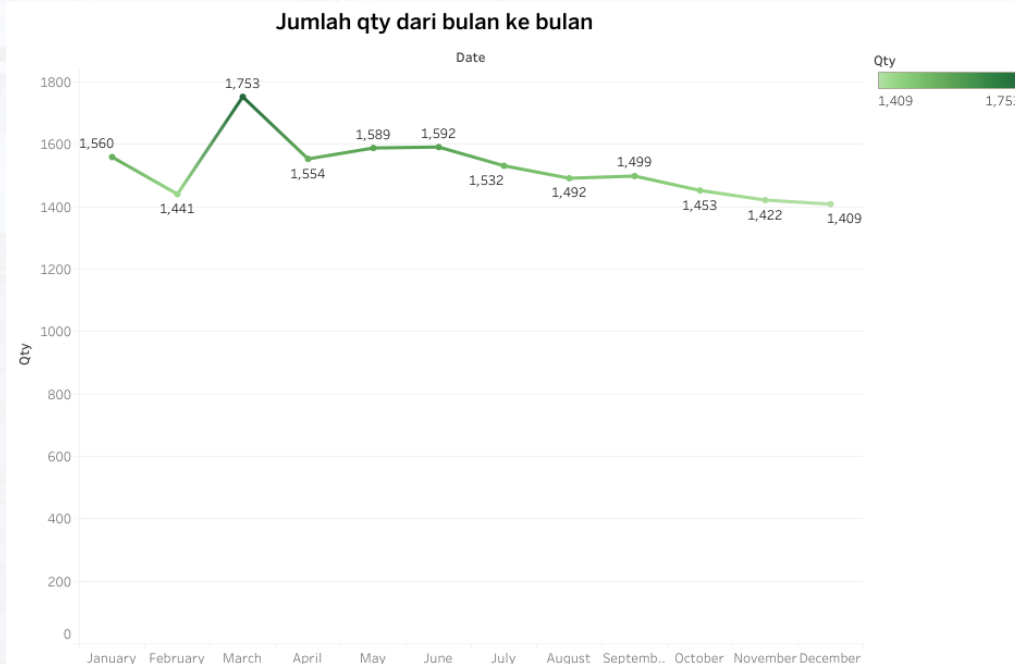
TABLE

-  Case Study - Customer.csv
-  Case Study - Product.csv
-  Case Study - Store.csv
-  Case Study - Transaction.csv

ERD TABLE

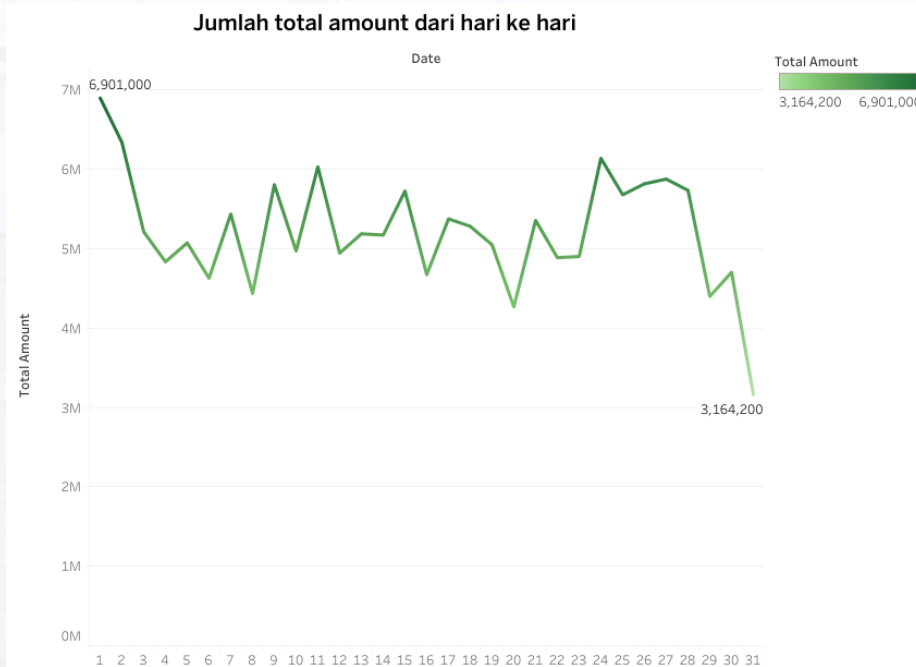


1. Jumlah qty dari bulan ke bulan



Terjadi Penurunan Qty dari Bulan ke Bulan

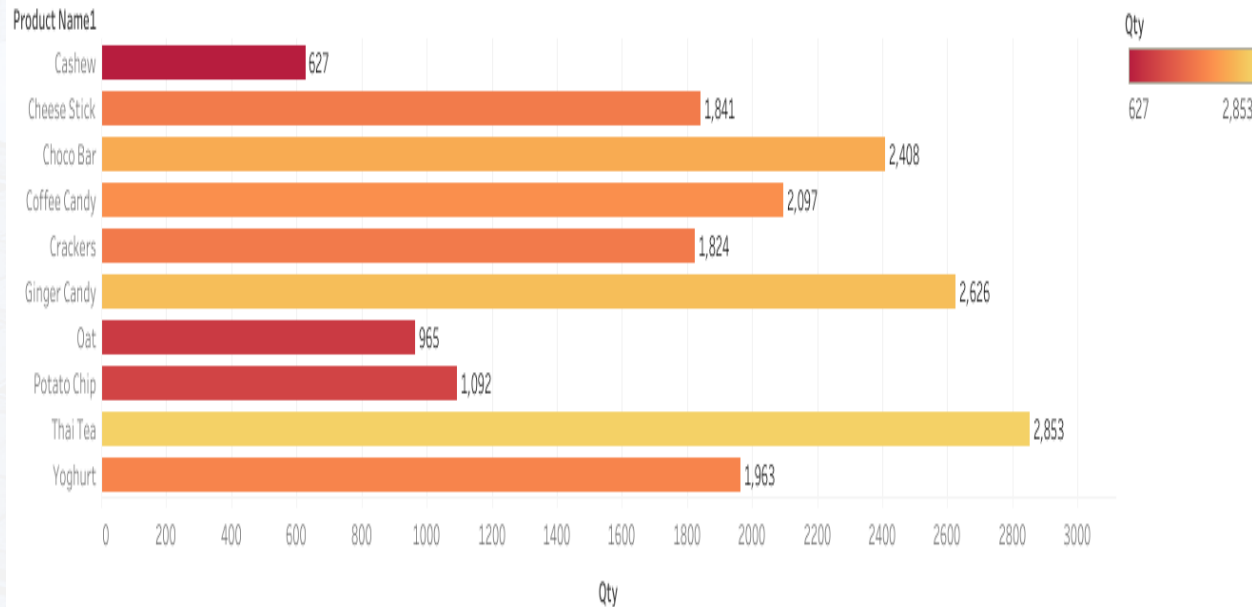
2. Jumlah total amount dari hari ke hari



Terjadi Penurunan total amount dari hari ke hari

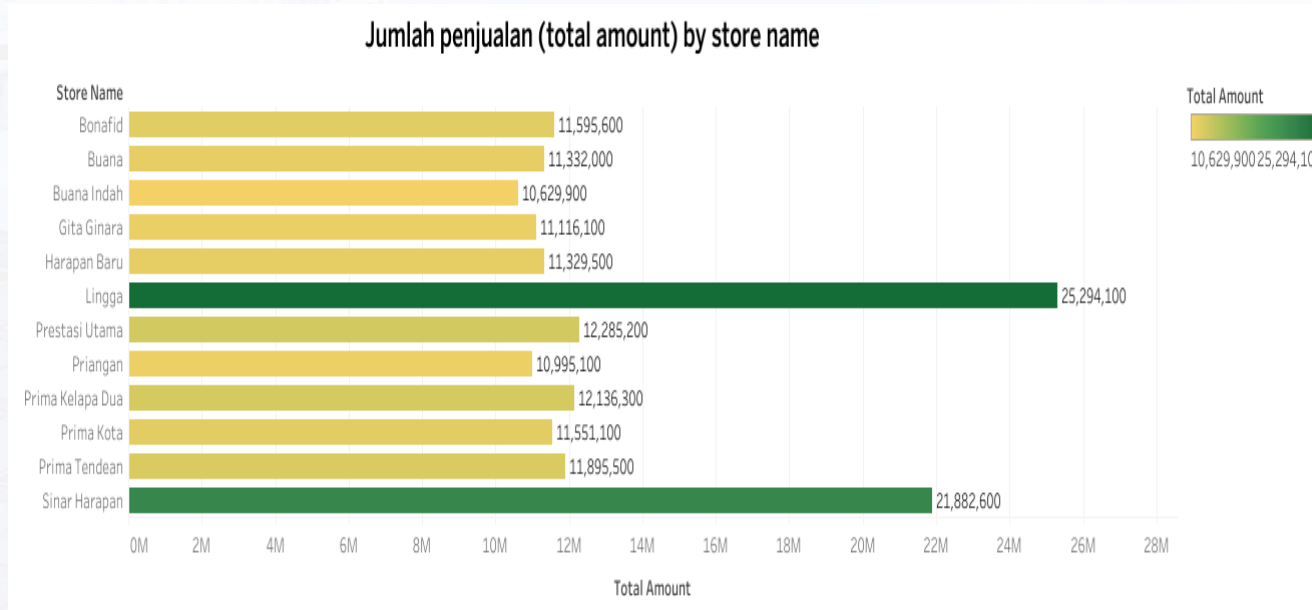
3. Jumlah penjualan (qty) by product

Jumlah penjualan (qty) by product



Thaitea adalah produk dengan penjualan terbanyak diikuti Ginger Candy

4. Jumlah penjualan (total amount) by store name



Lingga adalah store dengan penjualan terbanyak diikuti sinar harapan

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Import Library dan Import Data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import statsmodels.api as sm
from statsmodels.tsa.statespace.sarimax import SARIMAX
%matplotlib inline
```

```
df_customer = pd.read_csv('Case Study - Customer.csv', sep=';')
df_prod = pd.read_csv('Case Study - Product.csv', sep=';')
df_str = pd.read_csv('Case Study - Store.csv', sep=';')
df_tr = pd.read_csv('Case Study - Transaction.csv', sep=';')
```

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Data Cleansing and Preparation

```
# convert Date to datetime
df_tr['Date'] = pd.to_datetime(df_tr['Date'], format='%d/%m/%Y')
```

Convert tipe data date ke datetime pada dataset transaction

Cek Missing Values

```
print("Null Counts in df_customer:")
print(df_customer.isnull().sum())

print("\nNull Counts in df_prod:")
print(df_prod.isnull().sum())

print("\nNull Counts in df_str:")
print(df_str.isnull().sum())

print("\nNull Counts in df_tr:")
print(df_tr.isnull().sum())
```

Cek missing values pada setiap dataset

Cek Duplikat

```
print("nDuplicate Counts in df_customer:")
print(df_customer.duplicated().any())

print("\nDuplicate Counts in df_prod:")
print(df_prod.duplicated().any())

print("\nDuplicate Counts in df_str:")
print(df_str.duplicated().any())

print("\nDuplicate Counts in df_tr:")
print(df_tr.duplicated().any())
```

Cek duplikat pada setiap dataset

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Gabungkan Keempat Dataset menggunakan merge

```
df_merge = pd.merge(df_tr, df_customer, on=['CustomerID'])
df_merge = pd.merge(df_merge, df_prod.drop(columns=['Price']), on=['ProductID'])
df_merge = pd.merge(df_merge, df_str, on=['StoreID'])
```

df_merge.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5020 entries, 0 to 5019
Data columns (total 18 columns):
 #   Column          Non-Null Count  Dtype  
---  -
 0   TransactionID    5020 non-null   object  
 1   CustomerID       5020 non-null   int64   
 2   Date             5020 non-null   datetime64[ns]
 3   ProductID        5020 non-null   object  
 4   Price            5020 non-null   int64   
 5   Qty              5020 non-null   int64   
 6   TotalAmount      5020 non-null   int64   
 7   StoreID          5020 non-null   int64   
 8   Age              5020 non-null   int64   
 9   Gender           5020 non-null   int64   
10   Marital Status   4976 non-null   object  
11   Income           5020 non-null   object  
12   Product Name     5020 non-null   object  
13   StoreName        5020 non-null   object  
14   GroupStore       5020 non-null   object  
15   Type             5020 non-null   object  
16   Latitude         5020 non-null   object  
17   Longitude        5020 non-null   object  
dtypes: datetime64[ns](1), int64(7), object(10)
memory usage: 745.2+ KB
```

Gunakan info() untuk melihat info

Tampilkan kolom menggunakan fungsi df.sample()

	TransactionID	CustomerID	Date	ProductID	Price	Qty	TotalAmount	StoreID	Age	Gender	Marital Status	Income	Product Name	StoreName	GroupStore	Type
1269	TR7808	132	2022-04-02	P7	9400	2	18800	1	41	0	Married	17,69	Coffee Candy	Prima Tendean	Prima	Modern Trad
4395	TR95050	446	2022-08-24	P5	4200	4	16800	11	57	0	Married	7,81	Thai Tea	Sinar Harapan	Prestasi	Genera Trad
1494	TR96549	96	2022-05-14	P8	16000	2	32000	7	55	0	Married	13,67	Oat	Buana Indah	Buana	Genera Trad
2886	TR4401	342	2022-01-12	P4	12000	5	60000	3	48	1	Married	13,55	Potato Chip	Prima Kota	Prima	Modern Trad
1390	TR14050	431	2022-07-11	P10	15000	3	45000	1	40	1	Married	9,51	Cheese Stick	Prima Tendean	Prima	Modern Trad

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Membuat data time series

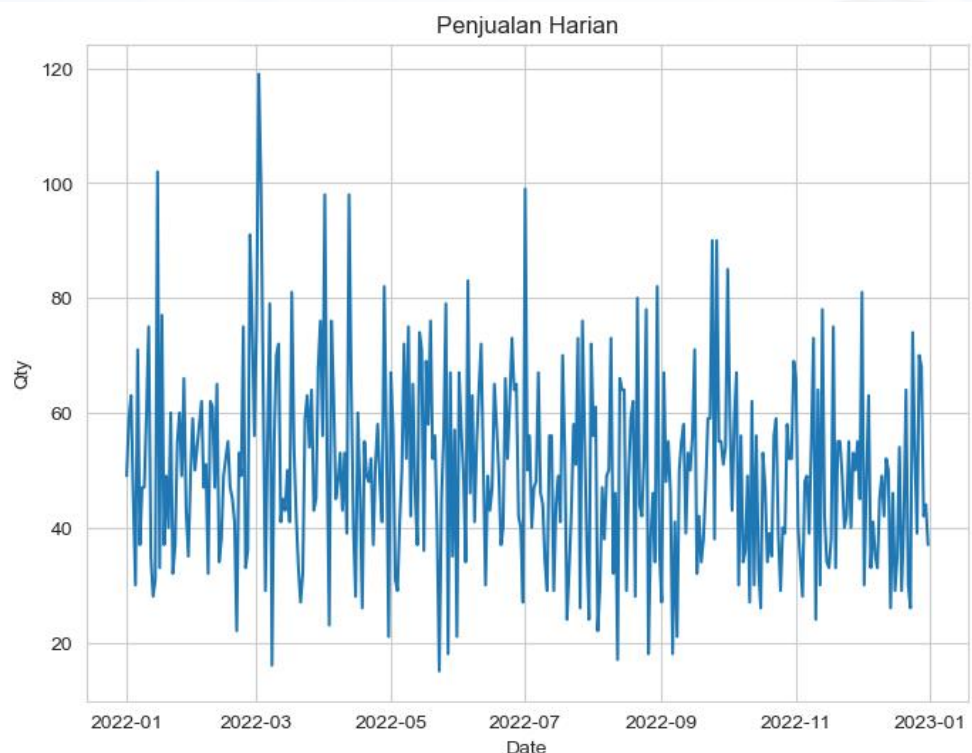
```
df_regresi = df_merge.groupby(['Date']).agg({'Qty':'sum'}).reset_index()
df_regresi['Date'] = pd.to_datetime(df_regresi['Date'], format='%d/%m/%Y')
df_regresi.sort_values(by='Date', inplace=True)
df_regresi.set_index('Date', inplace=True)
df_regresi.head()
```

Qty	
Date	
2022-01-01	49
2022-01-02	59
2022-01-03	63
2022-01-04	45
2022-01-05	30

Menampilkan timeplot untuk melihat penjualan harian

```
plt.figure(figsize=(8,6))
sns.set_style('whitegrid')
sns.lineplot(data=df_regresi, x='Date', y='Qty', legend=False)
plt.title('Penjualan Harian')
plt.xlabel('Date')
plt.ylabel('Qty')
plt.show()
```

Hasil Plot:



MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Memisahkan data

```
from sklearn.model_selection import train_test_split

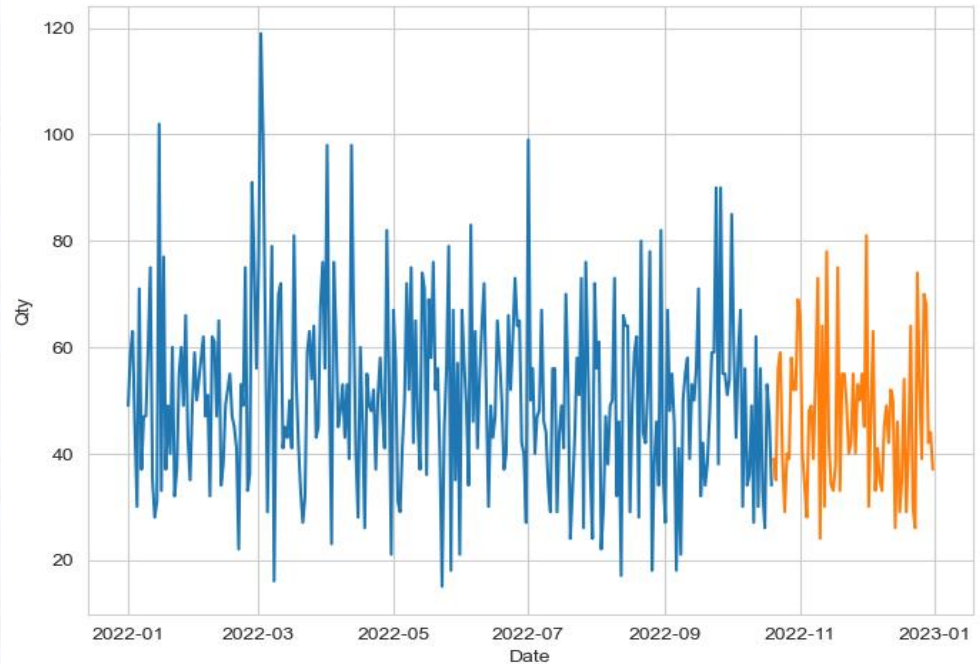
train, test = train_test_split(df_regresi, test_size=0.2, shuffle=False)
print(train.shape, test.shape)

(292, 1) (73, 1)
```

Membuat Lineplot train data dan test data

```
plt.figure(figsize=(8,6))
sns.set_style('whitegrid')
sns.lineplot(data=train, x=train.index, y=train['Qty'])
sns.lineplot(data=test, x=test.index, y=test['Qty'])
plt.show()
```

Hasil Plot:



MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Membuat Model *Machine Learning Regression* (Time Series)

Model ARIMA

```
from statsmodels.tsa.arima.model import ARIMA

# Menentukan nilai p,q dan d
p = 2
q = 2
d = 2

# Membuat Model ARIMA dengan Parameter yang telah ditentukan
model = ARIMA(train, order=(p,q,d))

# Melatih Model dengan menggunakan data test
model_fit = model.fit()
start_x = len(train)
end_x = len(train) + len(test) - 1
predictions = model_fit.predict(start=start_x, end=end_x, dynamic=False)
```

Untuk machine learning menggunakan model ARIMA

Mean Squared Error

```
# Evaluasi ke-1
from sklearn.metrics import mean_squared_error

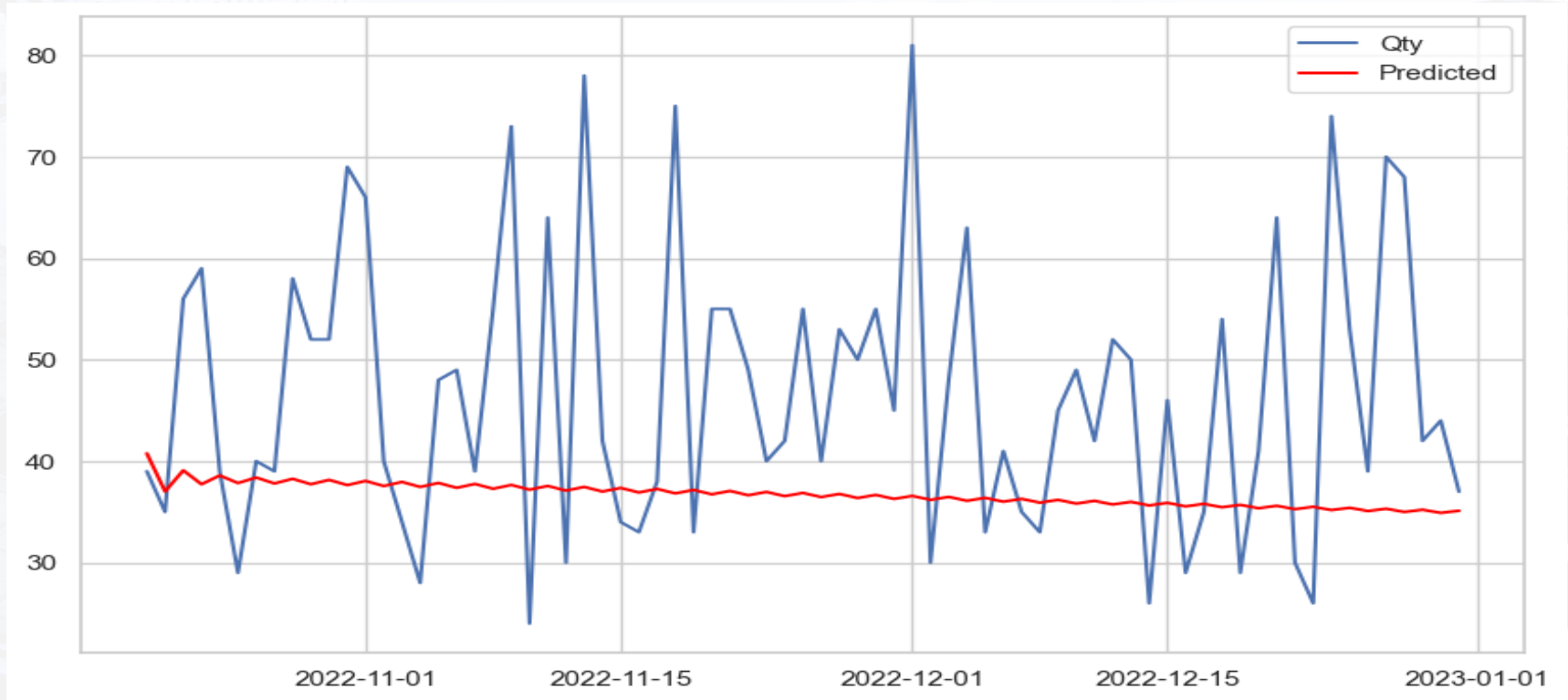
mse = mean_squared_error(test, predictions)
print(f"Mean Squared Error: {mse}")
```

Mean Squared Error: 286.80988202957997

Hasil Mean Squared Error adalah
286.80988202957997

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Hasil Plot



MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Membuat *Machine Learning Model Clustering*

```
aggregated = df_merge.groupby('CustomerID').agg({'TransactionID': 'count',  
                                                'Qty': 'sum',  
                                                'TotalAmount': 'sum'}).reset_index()  
aggregated
```

CustomerID	TransactionID	Qty	TotalAmount	
0	1	17	60	623300
1	2	13	57	392300
2	3	15	56	446200
3	4	10	46	302500
4	5	7	27	268600
...
442	443	16	59	485100
443	444	18	62	577700
444	445	18	68	587200
445	446	11	42	423300

Groupby customer id dengan transactionID, qty, dan total amount

Lakukan Clustering menggunakan K-Means

```
# Melakukan Clustering Menggunakan K-Means
```

```
X = aggregated[['TransactionID', 'Qty', 'TotalAmount']]
```

```
n_clusters = 4
```

```
from sklearn.cluster import KMeans
```

```
# Membuat Model kmeans
```

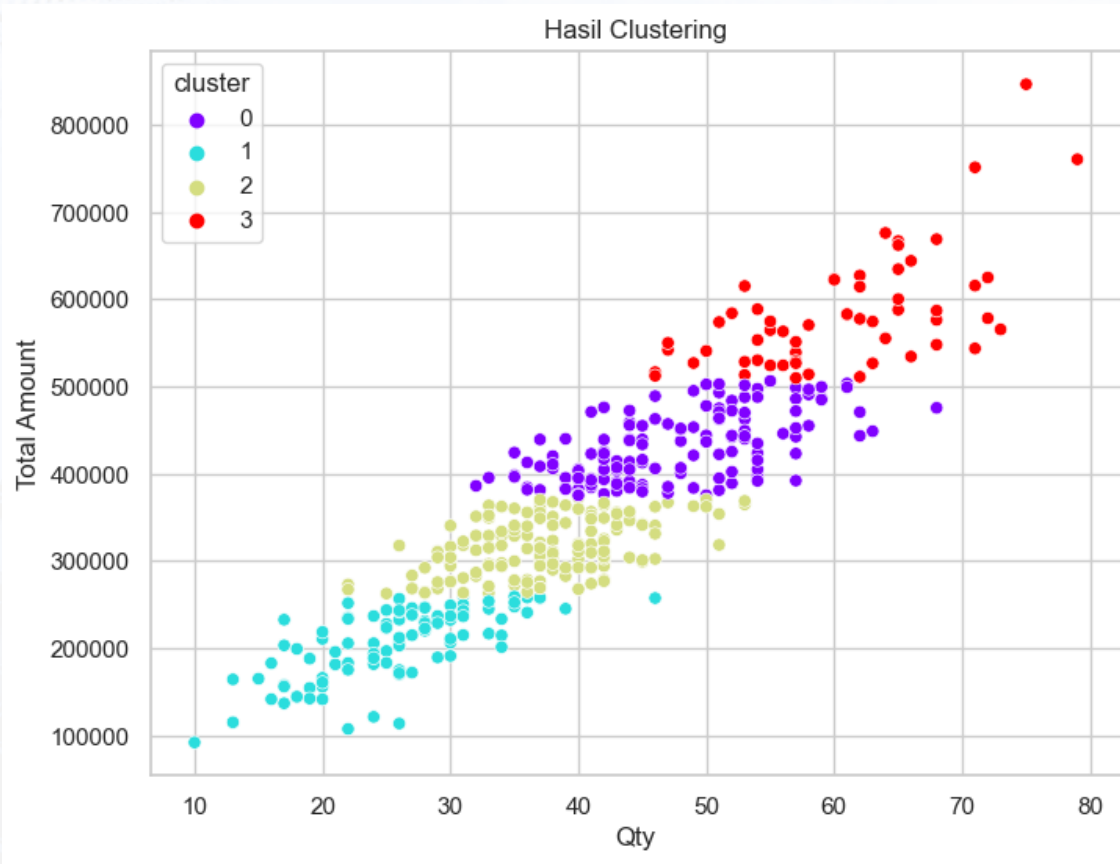
```
kmeans = KMeans(n_clusters=n_clusters, random_state=42)
```

```
# melakukan clustering pada data
```

```
aggregated['cluster'] = kmeans.fit_predict(X)
```

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Hasil Plot



MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Mencari WCSS dan Menampilkannya dalam plot

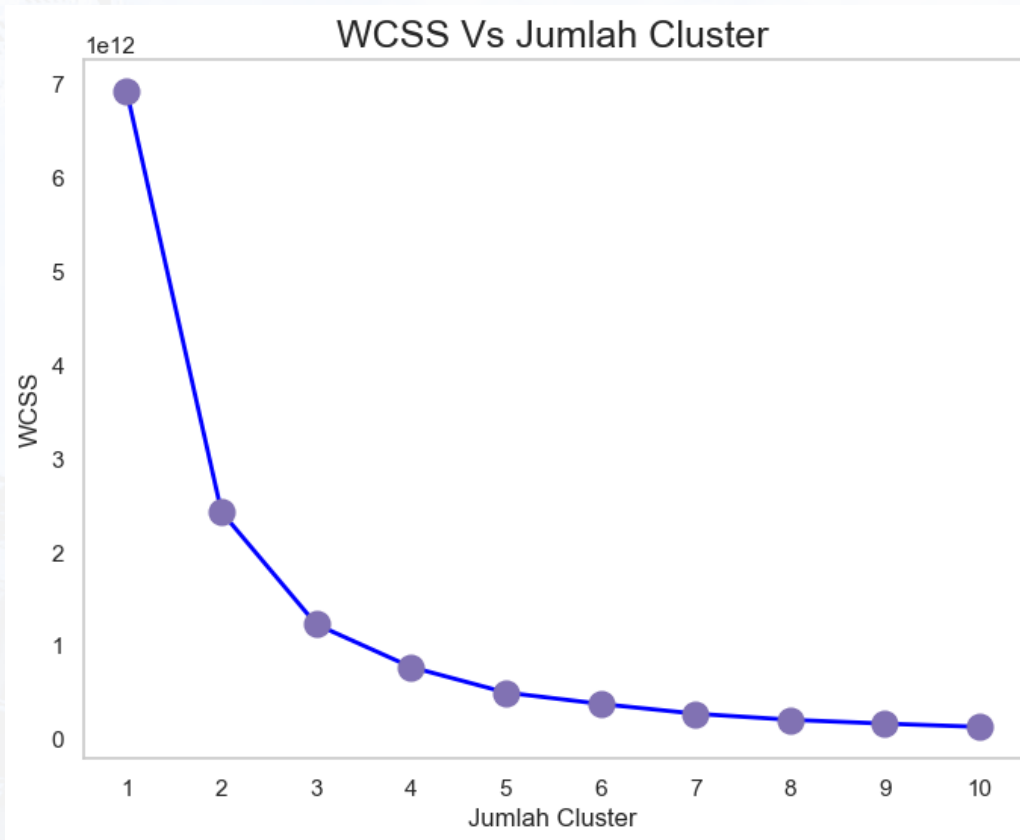
```
wcss = []  
for n in range(1,11):  
    model_1 = KMeans(n_clusters=n, init='k-means++', n_init=10, max_iter=100, tol=0.0001, random_state=100)  
    model_1.fit(X)  
    wcss.append(model_1.inertia_)  
print(wcss)
```

```
[6928031859602.738, 2434662706463.3955, 1233033389389.9624, 776512017046.2605, 504081360603.5857, 382593519595.5847, 2777669013  
61.2123, 212941531954.89276, 171724378723.10638, 136882282484.87318]
```

```
plt.figure(figsize=(8,6))  
plt.plot(list(range(1,11)), wcss, color='blue', marker='o', linewidth=2, markersize=12, markerfacecolor='m', markeredgecolor='m')  
plt.title('WCSS Vs Jumlah Cluster', fontsize=18)  
plt.xlabel('Jumlah Cluster')  
plt.ylabel('WCSS')  
plt.xticks(list(range(1,11)))  
plt.grid()  
plt.show()
```

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Hasil Plot



EXPLORATORY DATA ANALYSIS MENGGUNAKAN DBEAVER

Evaluasi seberapa compact data dalam sebuah kluster terhadap pusat kluster masing-masing

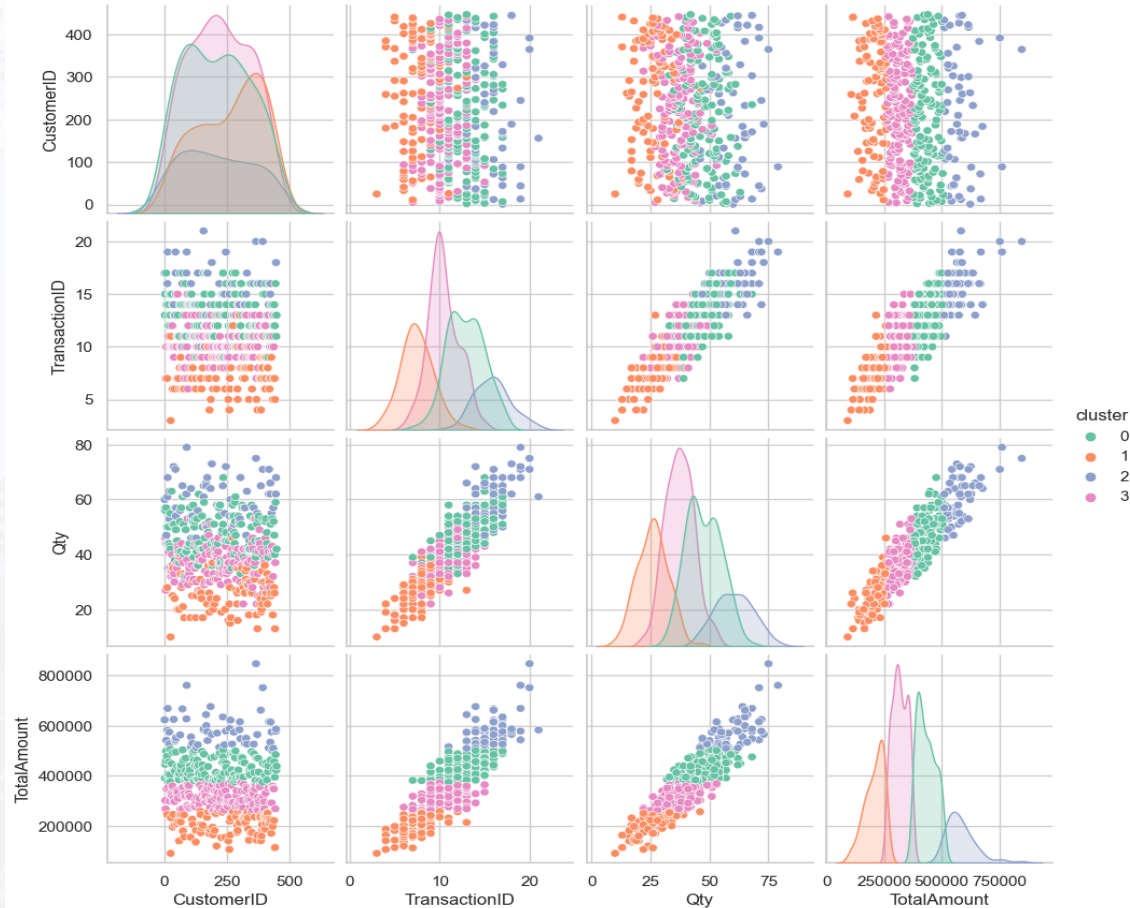
```
# mengevaluasi seberapa compact data dalam sebuah kluster terhadap pusat kluster masing-masing
model_1 = KMeans(n_clusters=4, init='k-means++', n_init=10, max_iter=300, tol=0.0001, random_state=100)
model_1.fit(X)
label_1 = model_1.labels_
centroids_1= model_1.cluster_centers_
```

```
aggregated['cluster'] = model_1.labels_
aggregated.head()
```

CustomerID	TransactionID	Qty	TotalAmount	cluster	
0	1	17	60	623300	2
1	2	13	57	392300	0
2	3	15	56	446200	0
3	4	10	46	302500	3
4	5	7	27	268600	3

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Hasil Plot



MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Hasil Clustering

```
df_cluster_mean = aggregated.groupby('cluster').agg({'CustomerID':'count',  
                                                    'TransactionID':'mean',  
                                                    'Qty':'mean',  
                                                    'TotalAmount':'mean'})  
df_cluster_mean.sort_values('CustomerID', ascending=False)
```

	CustomerID	TransactionID	Qty	TotalAmount
cluster				
3	152	10.414474	37.263158	316792.763158
0	140	12.871429	47.521429	431575.000000
1	98	7.448980	25.867347	206380.612245
2	57	15.877193	60.421053	583240.350877

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Cluster 1

- Cluster dengan jumlah pelanggan terbanyak
- Karakteristik dari cluster ini adalah cluster menempati posisi ketiga dari setiap metrik (transaction, quantity, total amount)

Rekomendasi

- Membangun hubungan baik dengan pelanggan
- Melakukan survey untuk mengembangkan minat pelanggan terbanyak

Cluster 2

Karakteristik dari cluster ini adalah cluster menempati posisi tertinggi kedua dari setiap metrik (transaction, quantity, total amount)

Rekomendasi

- Memberikan promo secara rutin untuk meningkatkan transaksi
- Melakukan peningkatan penjualan produk dengan harga tinggi

MODEL PREDIKTIF MENGGUNAKAN REGRESI DAN MEMBUAT CLUSTERING

Cluster 3

- Karakteristik pelanggan dengan nilai terendah dari setiap metrik (transaction, quantity, total amount)

Rekomendasi

- Memberikan diskon yang signifikan untuk meningkatkan transaksi
- Memberikan promo pada transaksi dengan jumlah barang yang lebih tinggi
- Melakukan survey untuk mengidentifikasi potensi pengembangan produk

Cluster 0

- Cluster dengan jumlah paling sedikit
- Karakteristik dari cluster ini adalah cluster menempati posisi tertinggi dari setiap metrik (transaction, quantity, total amount)

Rekomendasi

- Menawarkan promo program loyalitas untuk mempertahankan transaksi
- Melakukan survey kepuasan pelanggan
- Mendorong peningkatan penjualan produk dengan harga yang lebih tinggi

Link Github and LinkedIn

https://github.com/LimatanL/kalbe_nutrionals_VIX.git
<https://www.linkedin.com/in/limatanluviar/>

Thank You



Rakamin
Academy



KALBE
Nutritional