

Machine Learning Project

**Kalbe Nutritionals Data Scientist
Project Based Internship**

Presented by
Firman Maulana

My Experience

- Membuat prototype aplikasi bank sampah
- Membuat Database aplikasi bank sampah mobile



Firman Maulana

About You

Saat ini saya sedang menempuh pendidikan S1 Teknik Informatika di Universitas Brawijaya. Sepanjang perjalanan akademis, saya telah memperoleh landasan dalam bahasa pemrograman seperti Java,Sql dan Python.

Case Study

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.

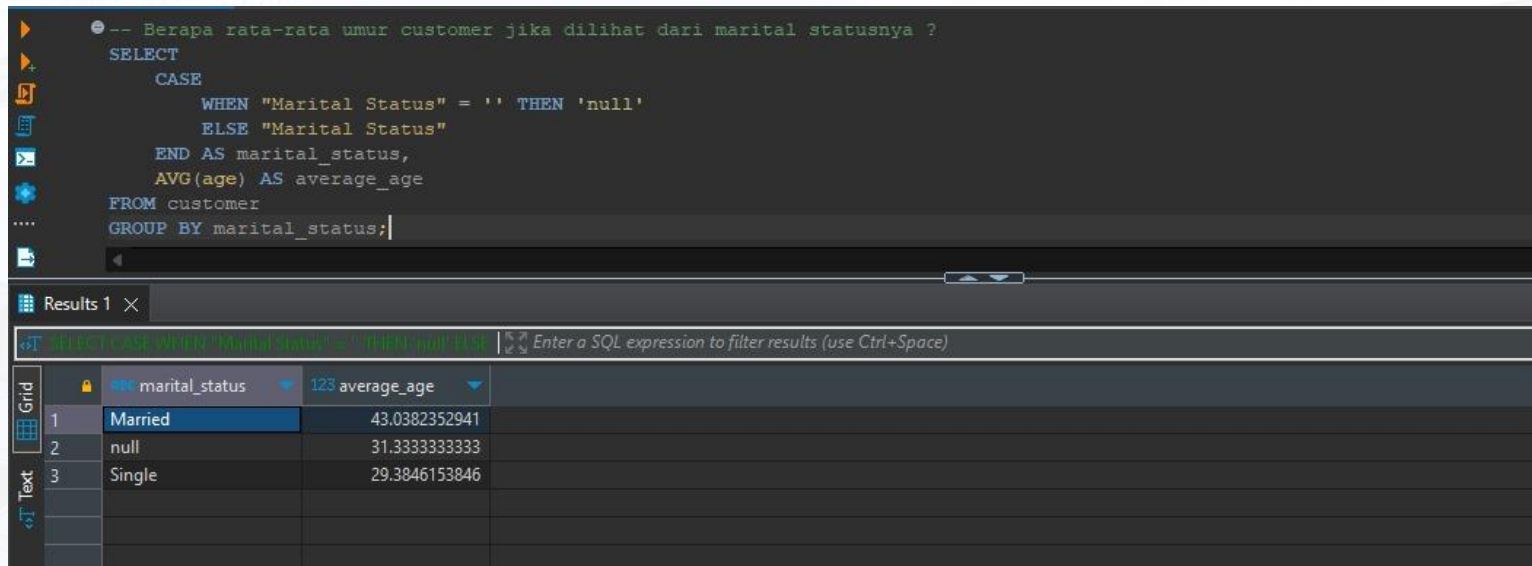
- ☐ Tujuan dari project ini adalah untuk membuat segment customer.
- ☐ Segment customer ini nantinya akan digunakan oleh tim marketing untuk memberikan personalized promotion dan sales treatment.

CHALLENGE DBEAVER

Melakukan exploratory data analysis di Dbeaver

Query 1

Berapa umur rata-rata umur customer jika dilihat dari marital statusnya?



The screenshot shows a SQL query in a dark-themed editor. The query is a SELECT statement with a CASE WHEN clause to handle null marital status, followed by an AVG function to calculate the average age, grouped by marital status. Below the editor, the 'Results 1' tab is active, displaying a table with 3 rows and 2 columns: 'marital_status' and 'average_age'. The first row shows 'Married' with an average age of 43.0382352941. The second row shows 'null' with an average age of 31.3333333333. The third row shows 'Single' with an average age of 29.3846153846.

```
-- Berapa rata-rata umur customer jika dilihat dari marital statusnya ?
SELECT
  CASE
    WHEN "Marital Status" = '' THEN 'null'
    ELSE "Marital Status"
  END AS marital_status,
  AVG(age) AS average_age
FROM customer
GROUP BY marital_status;
```

	marital_status	average_age
1	Married	43.0382352941
2	null	31.3333333333
3	Single	29.3846153846

Didapatkan hasil, rata-rata:

- Married : 43 tahun
- Single : 29 tahun

Dari data juga didapatkan terdapat missing value pada marital status

Query 2

Berapa umur rata-rata umur customer jika dilihat dari gendernya?

```
-- Berapa rata-rata umur customer jika dilihat dari gender nya ?
SELECT
  CASE
    WHEN gender = 0 THEN 'Female'
    WHEN gender = 1 THEN 'Male'
  END AS gender_label,
  AVG(age) AS average_age
FROM customer
GROUP BY gender;
```

Results 1 X

SELECT CASE WHEN gender = 0 THEN 'Female' WHEN gender = 1 THEN 'Male' END AS gender_label, AVG(age) AS average_age FROM customer GROUP BY gender

Grid	gender_label	average_age
1	Female	40.326446281
2	Male	39.1414634146

Didapatkan hasil, rata-rata:

- Female : 40 tahun
- Male : 39 tahun

Query 3

Tentukan nama store dengan total quantity terbanyak!

```
-- Tentukan nama store dengan total quantity terbanyak!  
SELECT store.storename, SUM(t.qty) AS total_qty  
FROM store  
JOIN "Transaction" t ON t.storeid = store.storeid  
GROUP BY store.storename  
ORDER BY total_qty DESC  
LIMIT 1;
```

store 1 X

SELECT store.storename, SUM(t.qty) AS total_qty FROM store

	storename	total_qty
1	Lingga	2,777

Dari data didapatkan hasil store dengan total quantity penjualan terbanyak adalah store Lingga sebanyak 2.777pcs

Query 4

Tentukan nama produk terlaris dengan total amount terbanyak!

```
SELECT product."Product Name", SUM(t.totalamount) AS total_amount
FROM product
JOIN "Transaction" t ON t.productid = product.productid
GROUP BY product."Product Name"
ORDER BY total_amount DESC
LIMIT 1;
```

product 1 x

Enter a SQL expression to filter results (use Ctrl+Space)

	Product Name	total_amount
1	Cheese Stick	27,615,000

Dari data didapatkan hasil produk terlaris dengan total amount terbanyak adalah Cheese Stick dengan total amount Rp. 27.615.000

CHALLENGE TABLEAU

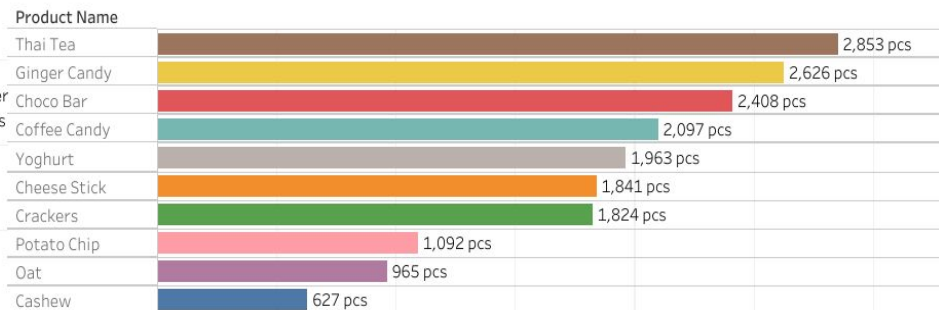
Memvisualisasi data dengan membuat dashboard

DASHBOARD

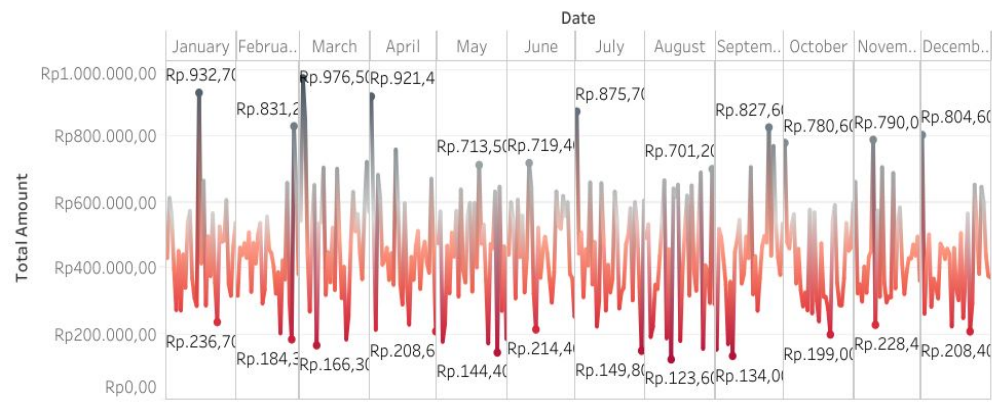
Jumlah Penjualan PerBulan



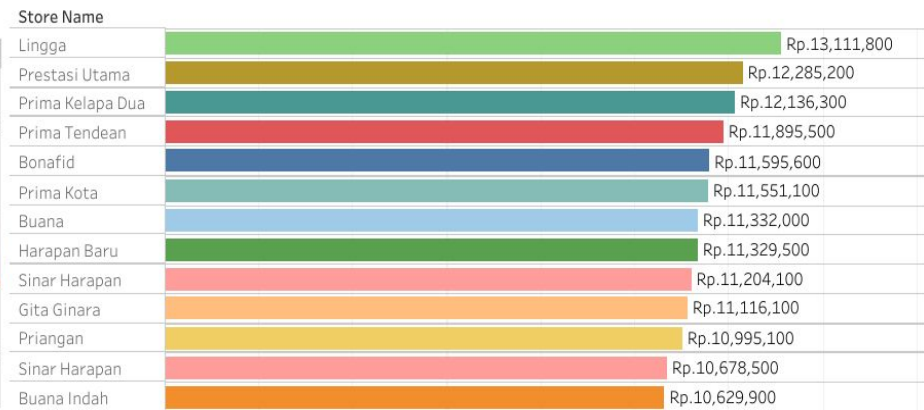
Jumlah Penjualan by Product



Jumlah Total Amount PerHari



Jumlah Penjualan by Store Name



CHALLENGE MACHINE LEARNING

Membuat model prediktif menggunakan metode regression dan membuat clustering

Membaca data csv

Convert CSV Files to Dataframe

```
df_customer = pd.read_csv('Customer.csv', delimiter= ';')
df_product = pd.read_csv('Product.csv', delimiter= ';')
df_store = pd.read_csv('Store.csv', delimiter= ';')
df_transaction = pd.read_csv('Transaction.csv', delimiter= ';')
```

Melakukan data cleansing

Data cleansing df_customer

```
[ ] df_customer['Income'] = df_customer['Income'].replace('[,]', '.', regex=True).astype('float')
```

Data cleansing df_Store

```
[ ] df_store['Latitude'] = df_store['Latitude'].replace('[,]', '.', regex=True).astype('float')
df_store['Longitude'] = df_store['Longitude'].replace('[,]', '.', regex=True).astype('float')
```

Data cleansing df_transaction

```
[ ] df_transaction['Date'] = pd.to_datetime(df_transaction['Date'])
```

```
<ipython-input-17-433e6c690dce>:1: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (the default) was specified. This may lead to inconsistently parsed dates! Specify a format to ensure consistent parsing.
df_transaction['Date'] = pd.to_datetime(df_transaction['Date'])
```

```
[ ] #Detecting the Missing Value
df_customer.isnull()
```


Menggabungkan data

Data Merge

```
[ ] df_merge = pd.merge(df_transaction, df_customer, on='CustomerID')
df_merge = pd.merge(df_merge, df_product.drop(columns=['Price']), on='ProductID')
df_merge = pd.merge(df_merge, df_store, on='StoreID')
```

df_merge.head()

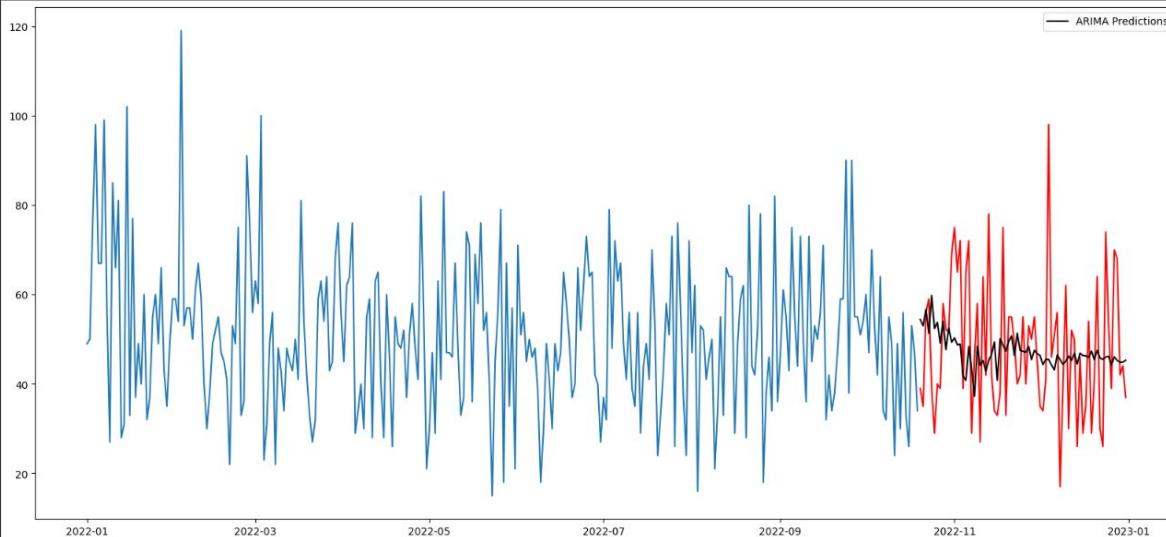
	TransactionID	CustomerID	Date	ProductID	Price	Qty	TotalAmount	StoreID	Age	Gender	Marital	Status	Income	Product Name	StoreName	GroupStore	Type	Latitude	Longitude
0	TR11369	328	2022-01-01	P3	7500	4	30000	12	36	0	Married		10.53	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554
1	TR89318	183	2022-07-17	P3	7500	1	7500	12	27	1	Single		0.18	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554
2	TR9106	123	2022-09-26	P3	7500	4	30000	12	34	0	Married		4.36	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554
3	TR4331	335	2022-08-01	P3	7500	3	22500	12	29	1	Single		4.74	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554
4	TR6445	181	2022-10-01	P3	7500	4	30000	12	33	1	Married		9.94	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554

Membuat model machine learning regression (time series)

```
y_pred_df['predictions'] = ARIMAmodel.predict(start=y_pred_df.index[0], end=y_pred_df.index[-1])
y_pred_df.index = df_test.index
y_pred_out = y_pred_df['predictions']

# Evaluate and plot the results
eval(df_test['Qty'], y_pred_out)
plt.figure(figsize=(20, 9))
plt.plot(df_train['Qty'])
plt.plot(df_test['Qty'], color='red')
plt.plot(y_pred_out, color='black', label='ARIMA Predictions')
plt.legend()
plt.show()
```

RMSE value: 15.47240015957555
MAE value: 12.357079716384185



Garis Biru: Data yang digunakan untuk melatih model. Data ini untuk membantu model memahami tren dan pola dalam data.

Garis Merah: Data aktual yang kita gunakan untuk menguji seberapa baik model dapat meramal. Kami ingin melihat sejauh mana prediksi model kami cocok dengan data sebenarnya. Garis Hitam: Hasil prediksi dari model. Model ARIMA mencoba untuk menebak data masa depan berdasarkan apa yang telah dipelajari dari data pelatihan.

Dengan membandingkan garis merah dan hitam, dapat dilihat apakah model efektif dalam meramal. Semakin dekat kedua garis ini, semakin baik model dalam memprediksi data sebenarnya.

Membuat model machine learning clustering

```
# Convert 'cluster' column to categorical data type
df_cluster['cluster'] = df_cluster['cluster'].astype('category')

# Create the scatter plot using Seaborn
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Qty', y='TotalAmount', data=df_cluster, hue='cluster', palette='Set1', s=70)
plt.xlabel('Quantity')
plt.ylabel('Total Amount')
plt.title('KMeans Clustering Customer Segmentation')
plt.legend(title='Cluster')
plt.show()
```



Berdasarkan segmentasi pelanggan dengan 3 cluster, dapat mensegmentasi setiap cluster untuk promosi dan perlakuan penjualan yang lebih personal.

Pertama kita akan melihat clustering KMeans berdasarkan kuantitas dan jumlah total untuk setiap cluster

Cluster 0 adalah pelanggan baru
Cluster 1 adalah loyal customers
Cluster 2 adalah calon loyal customers

Thank You



Rakamin
Academy



KALBE
Nutritional

Video Presentation Here

https://drive.google.com/drive/folders/1tAX9EnSclifnEhKuTD2zk1UtVhu_cJcx?usp=sharing

Insert Your Link Here

<https://github.com/Firmnm/Dashboard-Segmentation-Customer-Kalbe-Nutritionals.git>

https://drive.google.com/drive/folders/1tAX9EnScIfnEhKuTD2zk1UtVhu_cJcx?usp=sharing