Forecasting Sales dan Customer Segmentation

Virtual Internship Experience VIX at Kalbe Nutritionals

Presented by
Axel Ivanda Tanjung

Axel Ivanda Tanjung

About You

A results-driven professional and an enthusiastic learner with a strong interest in analytics and data science, also has some experience in supply chain. Convenient for analysis, forecasting, and end-to-end machine learning.
Interested in using data to generate insight, findings, and suggestions to address problems.

Experience

- AI ML Fellowship - PACMANN AI
- Cloud Practitioners AWS Re/Start
- AI Mastery Skill Academy

Case Study

Rakamin Academy

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.

Forecasting Sales

Data Preparation

Rakamin Academy

Import Common Package

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Read DataFrame (csv) as pd

```
df_customer = pd.read_csv(".\Final_Project\Dataset\Case_Study_Customer.csv", sep=";")
df_product = pd.read_csv(".\Final_Project\Dataset\Case_Study_Product.csv", sep=";")
df_store = pd.read_csv(".\Final_Project\Dataset\Case_Study_Store.csv", sep=";")
df_transaction = pd.read_csv(".\Final_Project\Dataset\Case_Study_Transaction.csv", sep=";")
```

Data Cleansing

Rakamin Academy

Data Cleansing untuk df_customer

	CustomerID	Age	Gender	Marital Status	Income
0	1	55	1	Married	5,12
1	2	60	1	Married	6,23
2	3	32	1	Married	9,17
3	4	31	1	Married	4,87
4	5	58	1	Married	3,57

	CustomerID	Age	Gender
count	447.000000	447.000000	447.000000
mean	224.000000	39.782998	0.458613
std	129.182042	12.848719	0.498842
min	1.000000	0.000000	0.000000
25%	112.500000	30.000000	0.000000
50%	224.000000	39.000000	0.000000
75%	335.500000	50.500000	1.000000
max	447.000000	72.000000	1.000000

```python
# Check missing data
df_customer.isna().sum()
```

✓ 0.0s

```
CustomerID        0
Age               0
Gender            0
Marital Status    3
Income            0
dtype: int64
```

```python
df_customer['Marital Status'].value_counts()/len(df_customer['Marital Status'])
```

✓ 0.0s

```
Married    0.760626
Single     0.232662
Name: Marital Status, dtype: float64
```

```python
# Berdasarkan data wrangling pada kolom Marital Status, terdapat 76% dengan status Married.
# Maka asumsikan NaN value dengan 'Married'
df_customer['Marital Status'] = df_customer['Marital Status'].fillna("Married")
```

Data Cleansing

Rakamin Academy

Data Cleansing untuk df_customer

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 447 entries, 0 to 446
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   CustomerID      447 non-null    int64
 1   Age             447 non-null    int64
 2   Gender          447 non-null    int64
 3   Marital Status  447 non-null    object
 4   Income          447 non-null    object
dtypes: int64(3), object(2)
memory usage: 17.6+ KB
```

```python
# Perbaiki tipe data
for i in range(len(df_customer['Income'])):
    df_customer['Income'][i] = df_customer['Income'][i].replace(",", ".")
```

```python
df_customer['Income'] = df_customer['Income'].astype(float)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 447 entries, 0 to 446
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   CustomerID      447 non-null    int64
 1   Age             447 non-null    int64
 2   Gender          447 non-null    int64
 3   Marital Status  447 non-null    object
 4   Income          447 non-null    float64
dtypes: float64(1), int64(3), object(1)
memory usage: 17.6+ KB
```

Data Cleansing

Rakamin Academy

Data Cleansing untuk df_transaction

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5020 entries, 0 to 5019
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   TransactionID  5020 non-null   object
 1   CustomerID     5020 non-null   int64 
 2   Date           5020 non-null   object
 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 
dtypes: int64(5), object(3)
memory usage: 313.9+ KB
```

```python
# Perbaiki tipe data untuk Date
df_transaction['Date'] = df_transaction['Date'].astype('datetime64')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5020 entries, 0 to 5019
Data columns (total 8 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         
dtypes: datetime64[ns](1), int64(5), object(2)
memory usage: 313.9+ KB
```

Merge DataFrame

Rakamin Academy

Gabungkan seluruh dataframe (df_transaction, df_customer, df_product, df_store)

```
df_merge = df_transaction.merge(df_customer, how='left')
df_merge = df_merge.merge(df_product, how='left')
df_merge = df_merge.merge(df_store, how='left')
```

```
# Check missing values
df_merge.head().isna().sum()
```

✓ 0.0s

```
TransactionID     0
CustomerID        0
Date              0
ProductID         0
Price             0
Qty               0
TotalAmount       0
StoreID           0
Age               0
Gender            0
Marital Status    0
Income            0
Product Name      0
StoreName         0
GroupStore        0
Type              0
Latitude          0
Longitude         0
dtype: int64
```

Grouping Quantity

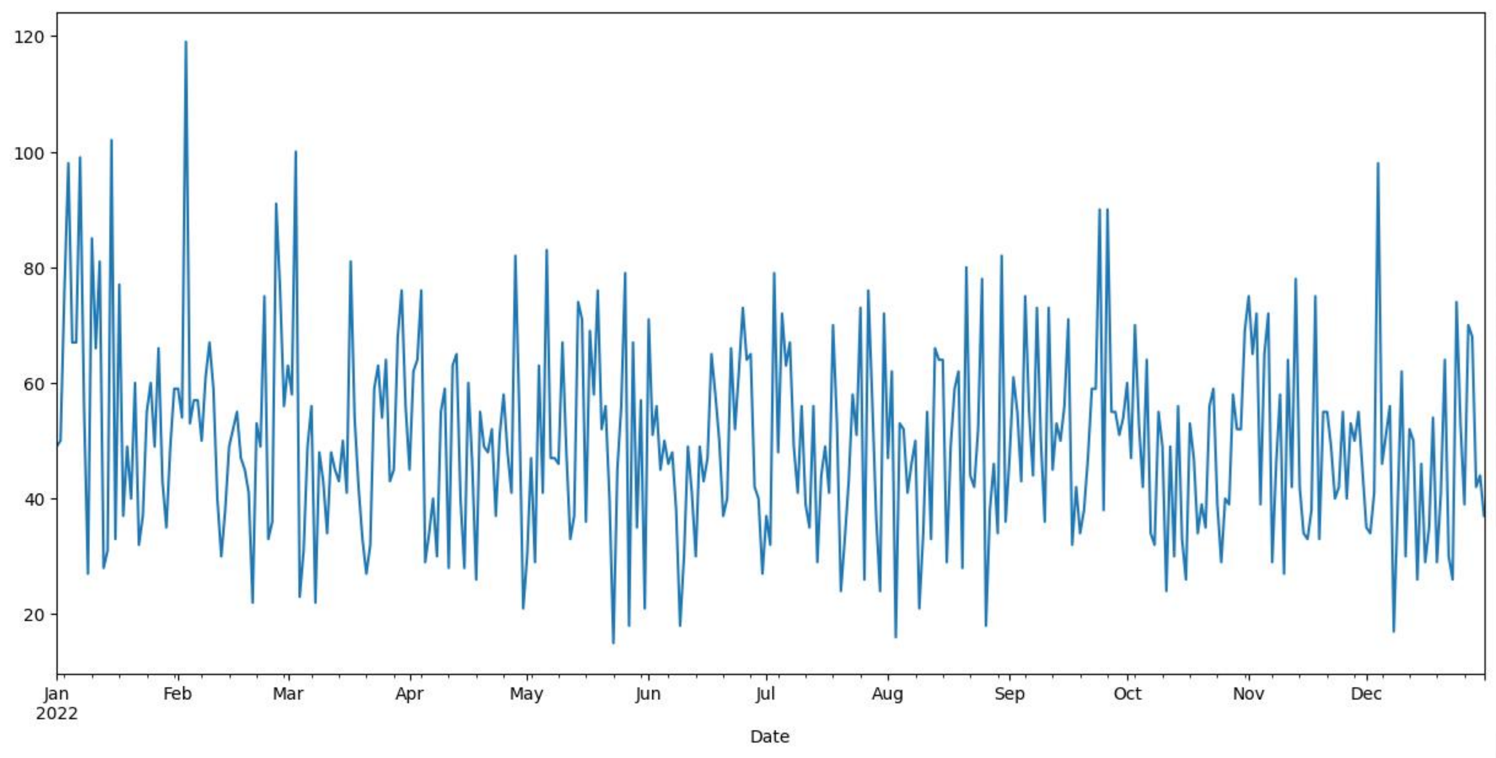
Rakamin Academy

Grouping berdasarkan “Date” dan jumlahkan berdasarkan “Qty”

```
# Membuat permodelan dengan frekuensi day, week, month
df_agg_day = df_merge.groupby("Date").agg('sum')
```

	CustomerID	Price	Qty	TotalAmount	StoreID	Age	Gender	Income
Date								
2022-01-01	2740	124300	49	431200	88	494	7	103.83
2022-01-02	2625	75600	50	317300	93	517	7	98.78
2022-01-03	5271	143600	76	544500	144	716	6	128.16
2022-01-04	5810	271900	98	921400	196	1097	12	237.56
2022-01-05	4237	166500	67	476400	146	815	11	154.10

Plotting Qty vs Date



Regression (ARIMA)

Rakamin Academy

Melakukan pengujian terhadap dataset (Stationer / Non-Stationer)

```
# Mengecek apakah dataset stationer
from statsmodels.tsa.stattools import adfuller

df_model_d = df_qty_day.copy()

test_result_d=adfuller(df_model_d)
```

- H0: The null hypothesis: It is a statement about the population that either is believed to be true or is used to put forth an argument unless it can be shown to be incorrect beyond a reasonable doubt.
- H1: The alternative hypothesis: It is a claim about the population that is contradictory to H0 and what we conclude when we reject H0.
- #Ho: It is non-stationary
- #H1: It is stationary

Regression (ARIMA)

Rakamin Academy

```
def adfuller_test(sales):
    result=adfuller(sales)
    labels = ['ADF Test Statistic','p-value','#Lags Used','Number of Observations']
    for value,label in zip(result,labels):
        print(label+' : '+str(value) )

    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data is stationary")
    else:
        print("weak evidence against null hypothesis,indicating it is non-stationary ")

adfuller_test(df_model_d)
```

```
ADF Test Statistic : -19.018782802299725
p-value : 0.0
#Lags Used : 0
Number of Observations : 364
strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data is stationary
```

Difference and Seasonality

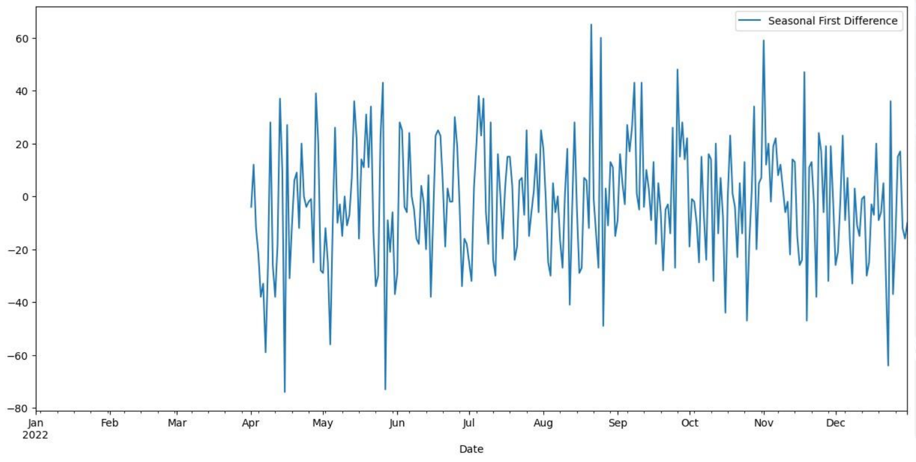
Rakamin Academy

```
df_model_d['Sales First Difference'] = df_model_d['Qty'] - df_model_d['Qty'].shift(1)
df_model_d['Seasonal First Difference'] = df_model_d['Qty'] - df_model_d['Qty'].shift(90)
df_model_d.head()
```

	Qty	Sales First Difference	Seasonal First Difference
Date			
2022-01-01	49	NaN	NaN
2022-01-02	50	1.0	NaN
2022-01-03	76	26.0	NaN
2022-01-04	98	22.0	NaN
2022-01-05	67	-31.0	NaN

```
# Again testing if data is stationary
adfuller_test(df_model_d['Seasonal First Difference'].dropna())
```

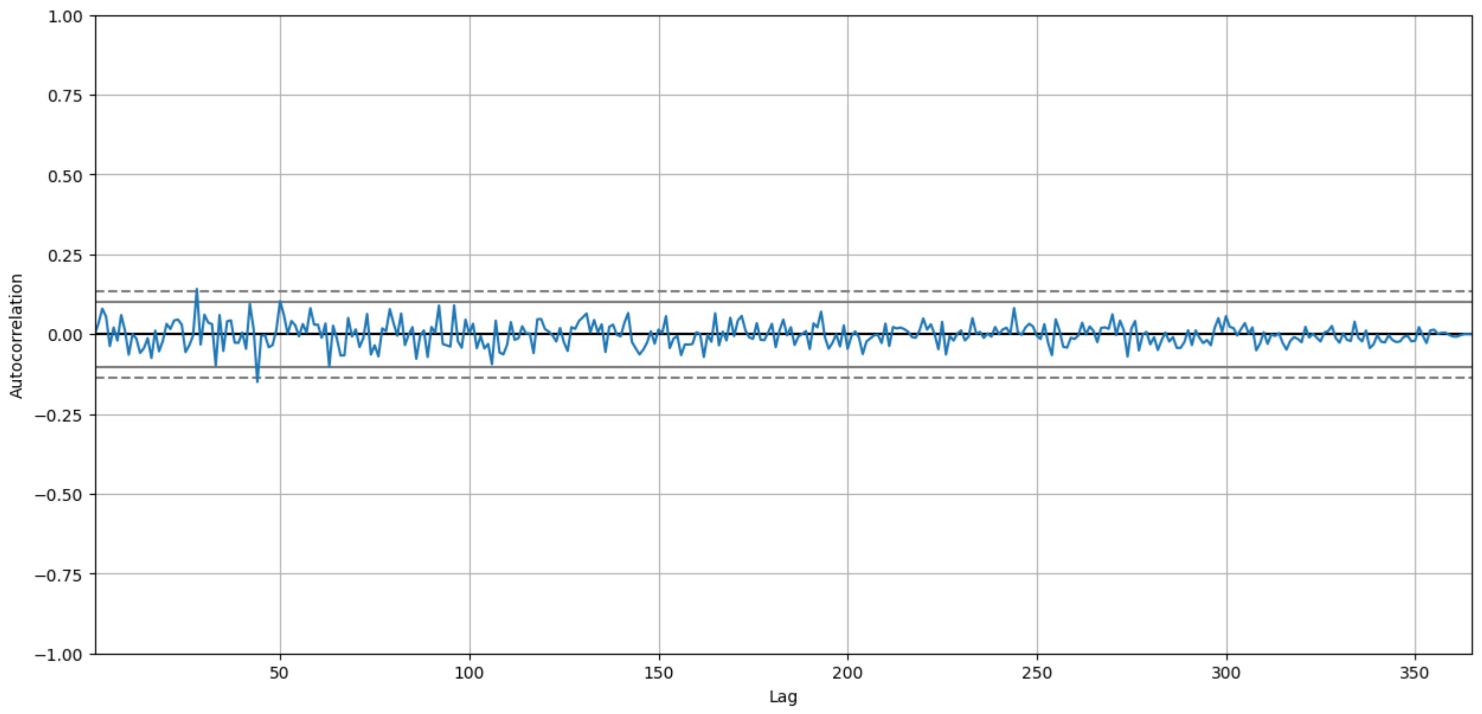
```
df_model_d['Seasonal First Difference'].plot()
plt.legend()
```



Auto Correlation

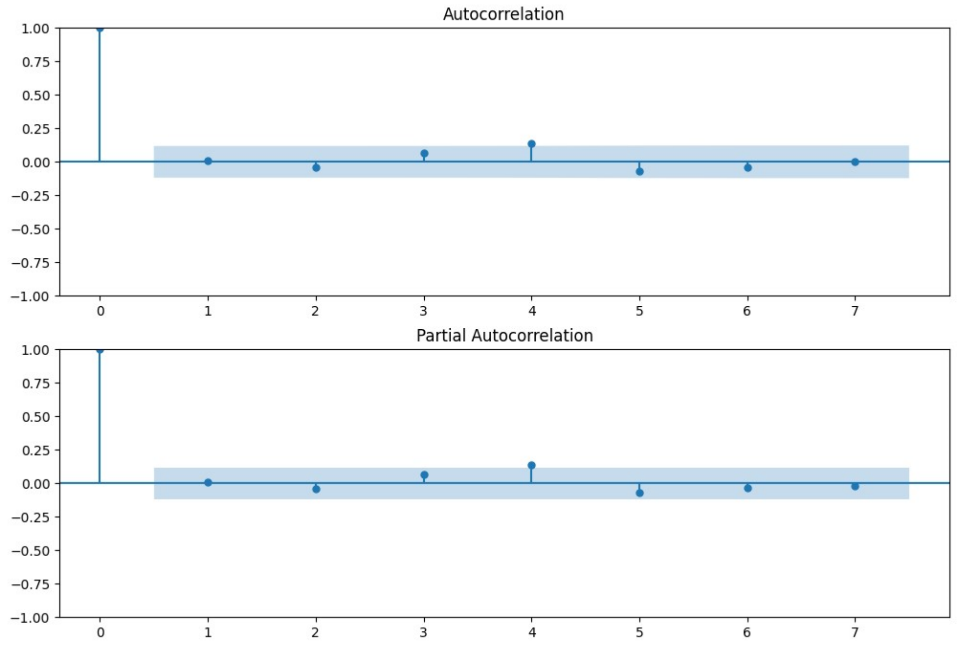
Rakamin Academy

```
from pandas.plotting import autocorrelation_plot
autocorrelation_plot(df_model_d['Qty'])
plt.show()
```



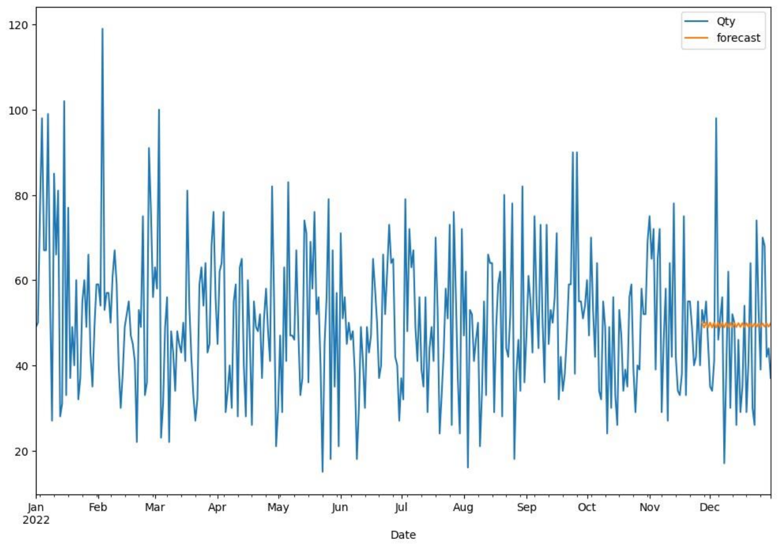
Auto Correlation

```
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
import statsmodels.api as sm
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(df_model_d['Seasonal First Difference'].dropna(),lags=7,ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(df_model_d['Seasonal First Difference'].dropna(),lags=7,ax=ax2)
```



SARIMAX (Non-Seasonal)

```
# For non-seasonal data
#p=1, d=1, q=0 or 1

import statsmodels.api as sm
model_d = sm.tsa.arima.ARIMA(df_model_d['Qty'], order=(1,1,2))
model_fit_d=model_d.fit()
model_fit_d.summary()
```



SARIMAX Results

Dep. Variable:	Qty	No. Observations:	365
Model:	ARIMA(1, 1, 2)	Log Likelihood	-1542.753
Date:	Sun, 01 Oct 2023	AIC	3093.506
Time:	12:38:22	BIC	3109.094
Sample:	01-01-2022	HQIC	3099.701
	- 12-31-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.9858	0.038	-25.850	0.000	-1.061	-0.911
ma.L1	-0.0050	0.050	-0.100	0.921	-0.104	0.094
ma.L2	-0.9599	0.050	-19.297	0.000	-1.057	-0.862
sigma2	278.4989	20.493	13.590	0.000	238.333	318.664

Ljung-Box (L1) (Q):	0.07	Jarque-Bera (JB):	11.02
Prob(Q):	0.80	Prob(JB):	0.00
Heteroskedasticity (H):	0.69	Skew:	0.40
Prob(H) (two-sided):	0.04	Kurtosis:	3.29

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

SARIMAX (Seasonal)

Rakamin Academy

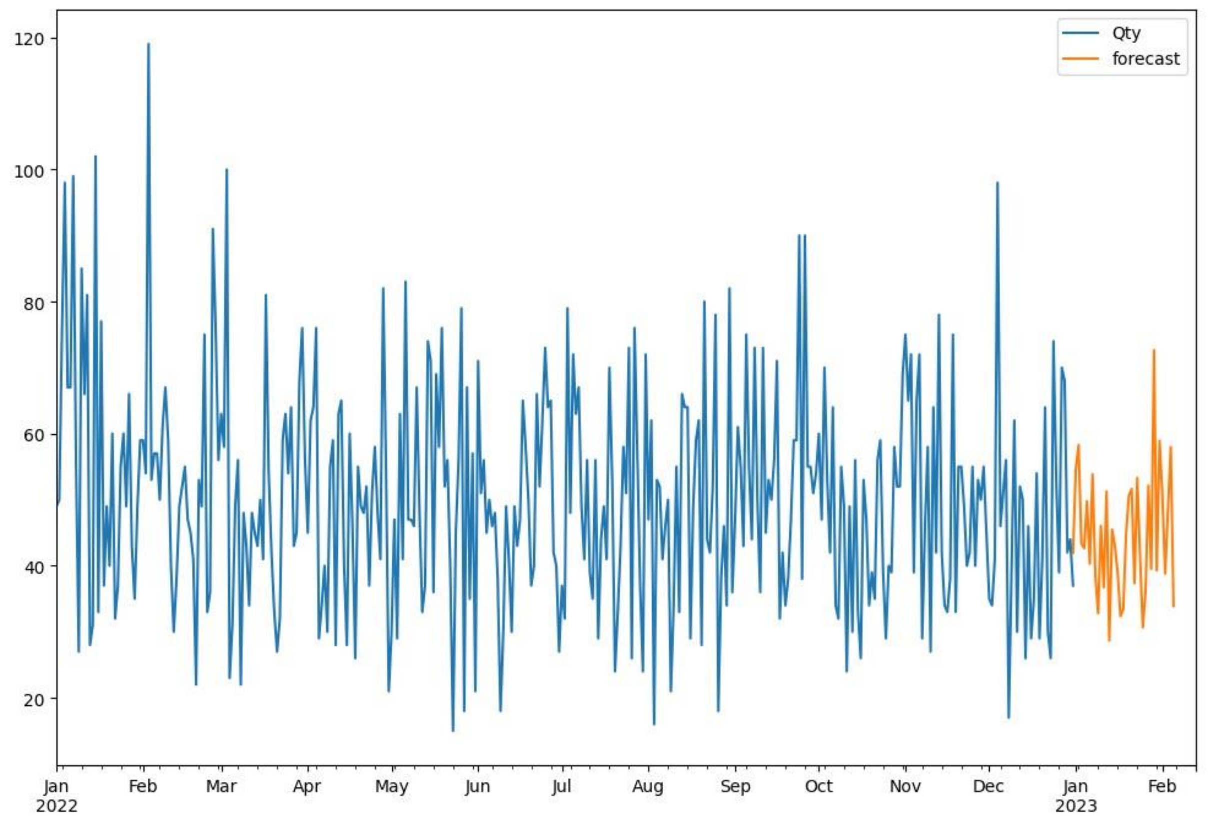
```
import statsmodels.api as sm
model_dd=sm.tsa.statespace.SARIMAX(df_model_d['Qty'],order=(1, 1, 1),seasonal_order=(1,1,1,90))
results_dd=model_dd.fit()
df_model_d['forecast']=results_dd.predict(start=330,end=364,dynamic=True)
df_model_d[['Qty','forecast']].plot(figsize=(12,8))
```

```
from pandas.tseries.offsets import DateOffset
future_dates_d=[df_model_d.index[-1]+ DateOffset(days=x)for x in range(0,45)]
future_datest_df_d=pd.DataFrame(index=future_dates_d[1:],columns=df_model_d.columns)
```

```
future_df_d=pd.concat([df_model_d,future_datest_df_d])
```

```
future_df_d['forecast'] = results_dd.predict(start = 364, end = 400, dynamic= True)
future_df_d[['Qty', 'forecast']].plot(figsize=(12, 8))
```

SARIMAX (Seasonal)



Customer Segmentation

Data Grouping

Rakamin Academy

Group CustomerID berdasarkan Transaction ID (Count), Qty (Sum), dan TotalAmount (sum)

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

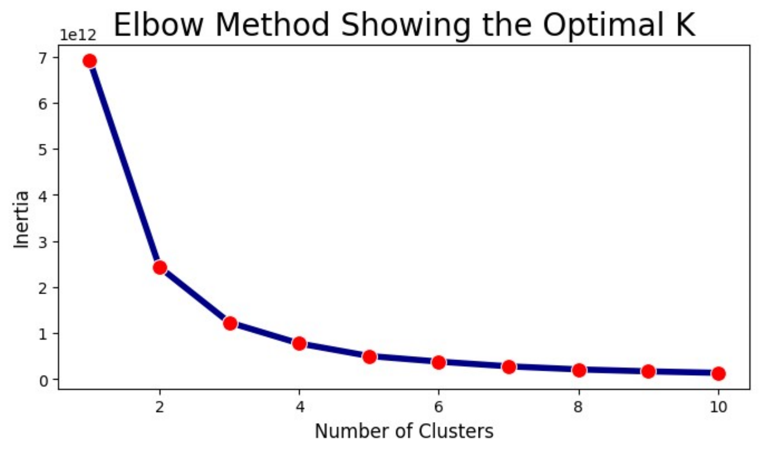
	TransactionID	Qty	TotalAmount
CustomerID			
1	17	60	623300
2	13	57	392300
3	15	56	446200
4	10	46	302500
5	7	27	268600

Elbow Methods

Rakamin Academy

```python
# Create place holder for inertia (empty list)
inertia = []

# Iteration
for k in range(1, 11):
    # Create k means
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)

    # Fit the object
    kmeans.fit(df_cluster)

    # Append the result
    inertia.append(kmeans.inertia_)
```

```python
plt.figure(figsize=(8,4))

# plt.plot(inertia)
sns.lineplot(x=range(1,11), y=inertia, color='#000087',
             linewidth = 4, marker='o', markersize=10,
             markerfacecolor='red')

plt.xlabel('Number of Clusters', fontsize = 12)
plt.ylabel('Inertia', fontsize=12)
plt.title('Elbow Method Showing the Optimal K', fontsize=20)
```



Fitting & Labeling

Rakamin Academy

```
# Fit for the best data
best_kmeans_cluster = KMeans(n_clusters=3,
                             random_state=42,
                             n_init=10) # Number of times the k-means algorithm will be run with different initial centroids

# Fit the model
best_kmeans_cluster.fit(df_cluster)
```

```
# Prediction
label_data = best_kmeans_cluster.predict(df_cluster)
label_data
```

Generate Centroid

Rakamin Academy

```
# Generate the coordinate centroid
centroids = best_kmeans_cluster.cluster_centers_
```

```
# Create DataFrame
df_centroid = pd.DataFrame(centroids,
                           columns=['TransactionID', 'Qty', 'TotalAmount']).rename_axis('Cluster', axis='index')

# Centroid DataFrame
df_centroid
```

Cluster	TransactionID	Qty	TotalAmount
0	11.752688	42.779570	384003.225806
1	8.508772	29.888889	241425.730994
2	15.322222	58.088889	548162.222222

Define Segment

- **Cluster 0** --> Middle Customer
  - **Characteristic** : Medium Quantity, Medium Frequency, Medium Monetary
- **Cluster 1** --> Highly Potential Churn
  - **Characteristic** : Low Quantity, Low Frequency, Low Monetary
- **Cluster 2** --> Enthusiastic Customer
  - **Characteristic** : High Quantity, High Frequency, High Monetary

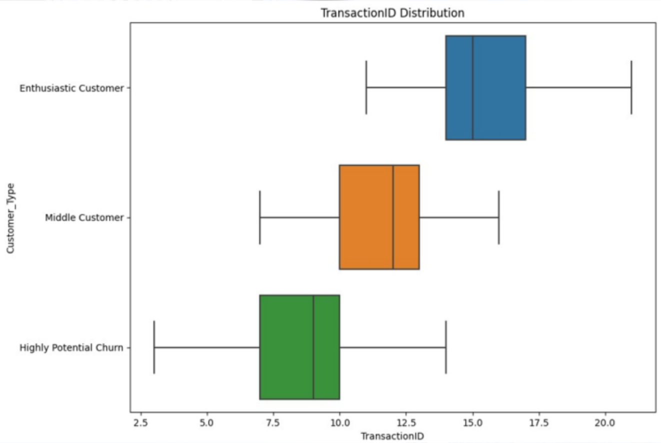
Assign Label

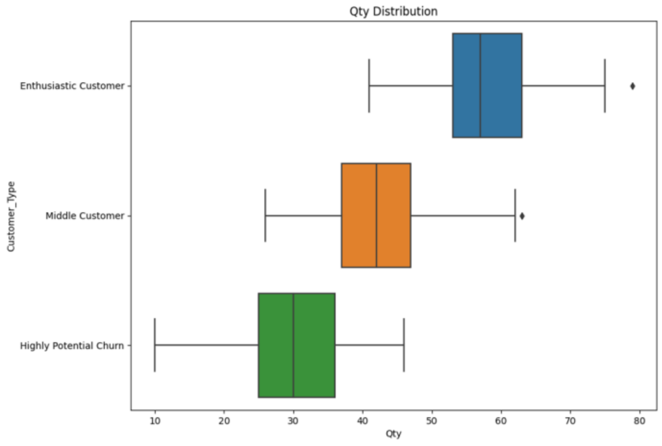
Rakamin Academy

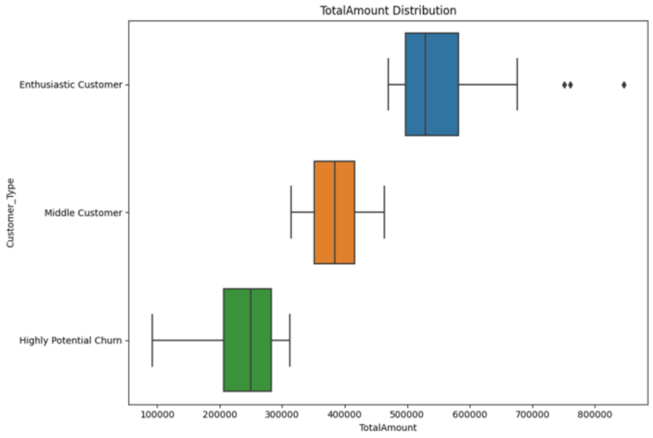
```python
# Create function to assign the label
def assign_label(value):
    """
    Function to assign cluster label

    Parameters:
    ----------
    value : int
        Number cluster

    Returns:
    --------
    label : str
        Label of cluster (Middle Customer, Highly Potential Churn, and Enthusiastic Customer)

    """
    if value == 0:
        return "Middle Customer"
    elif value == 1:
        return "Highly Potential Churn"
    else:
        return "Enthusiastic Customer"
# Create a column 'Customer_Type' based on 'K_Cluster' values
df_cluster["Customer_Type"] = df_cluster['label'].apply(assign_label)

# Check the result
df_cluster.head()
```

	TransactionID	Qty	TotalAmount	label	Customer_Type
CustomerID					
1	17	60	623300	2	Enthusiastic Customer
2	13	57	392300	0	Middle Customer
3	15	56	446200	0	Middle Customer
4	10	46	302500	1	Highly Potential Churn
5	7	27	268600	1	Highly Potential Churn

Segment Boxplot







General Recommendation

Enthusiastic Customer

- General Characteristic (high quantity, high frequency & monetary)
  - Quantity : 58 item
  - Frequency : 15 transactions
  - Monetary : \Rp. 548.162
- General marketing initiatives:
  - Provide tailored product suggestions based on their previous purchases.
  - Establish a customer loyalty program with tiers of prizes and special advantages.

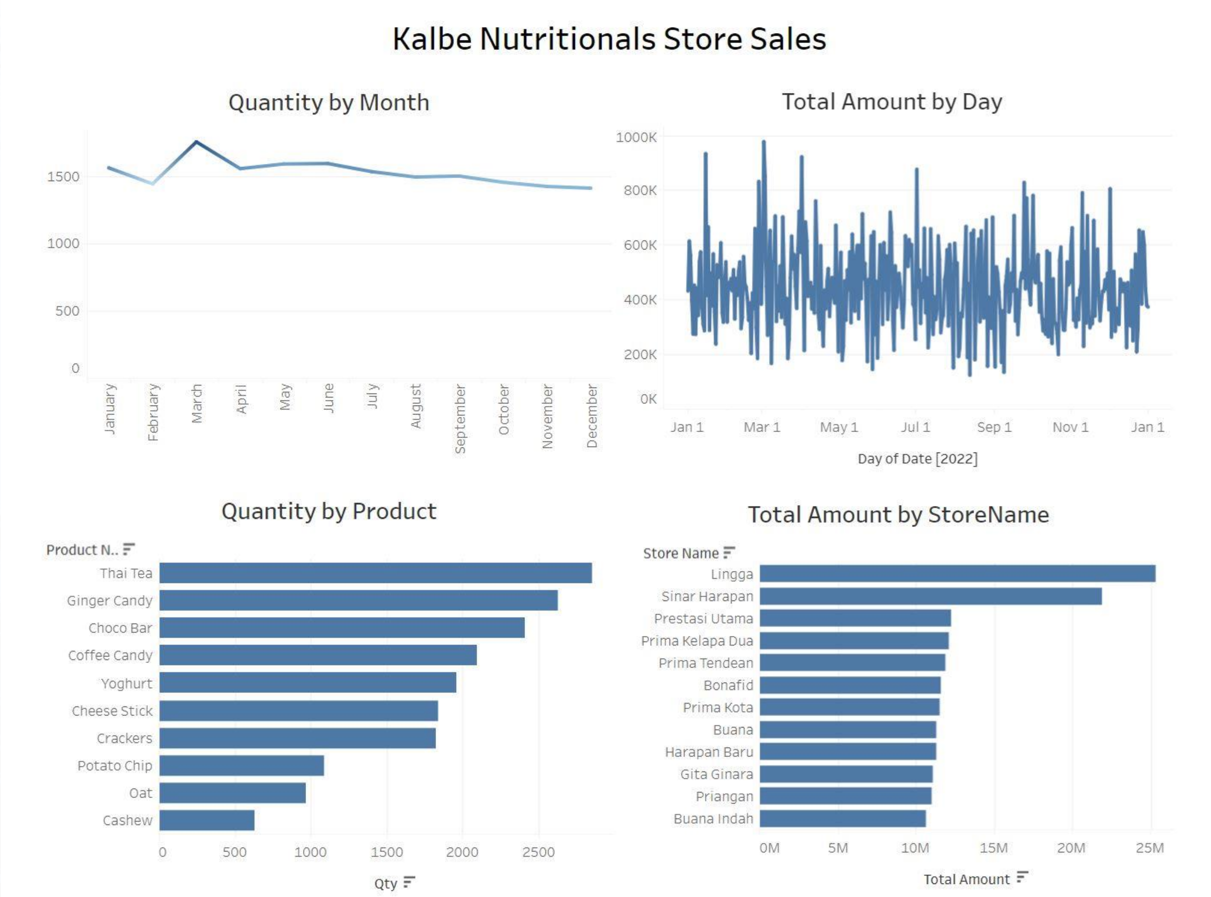
Middle Customers

- General Characteristic (medium quantity, medium frequency & monetary)
  - Quantity : 43 days
  - Frequency : 12 transactions
  - Monetary : \Rp. 348.003
- General marketing initiatives:
  - Encourage people to sample your goods or services by providing them with introductory discounts.

Highly Potential Churn

- General Characteristic (low quantity, low frequency & monetary)
  - Quantity : 29 days
  - Frequency : 9 transactions
  - Monetary : \Rp. 241.425
- General Marketing initiative:
  - Implement a proactive customer outreach program to answer any complaints or issues they may have.
  - Conduct surveys or feedback campaigns to better understand their wants and preferences.
  - Use discount promotional campaigns to entice them to make a purchase.

Kalbe Sales Dashboard



You Can Reach Me On

LinkedIn:
https://www.linkedin.com/in/axel-ivanda-tanjung/

GitHub :
https://github.com/axeltanjung/kalbe_forecast_cluster

Tableau :
https://public.tableau.com/app/profile/axel.ivanda.tanjung/viz/KalbeSalesDashboard_16960416448250/Dashboard1

Medium :
https://medium.com/@axelivandatanjung

Linktree :
https://linktr.ee/axeltanjung

Thank You

Rakamin Academy X KALBE Nutritionals