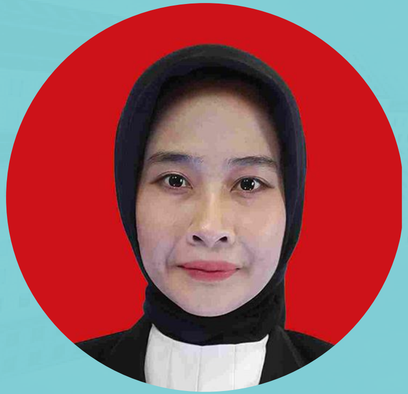
Rakamin Academy X KALBE Nutritionals

Machine Learning Project

Sales Prediction & Customer Segmentation Using ARIMA & K-Means Clustering

Kalbe Nutritionals Data Scientist Project Based Internship Program

Presented by
Nicken Shidqia Nurahman



Nicken Shidqia Nurahman

About Me

Civil engineer graduate with some experience in administration and project management, who is interested in data science.
Detail oriented, and time management person, and familiar with Microsoft Office, Python, HTML, CSS, SQL and Jupyter. Motivated to continue to learn and grow as a professional.

My Experience

Rakamin Academy

Data Science Bootcamp Student – RAKAMIN ACADEMY
Oct 2023 - Now

Project Management Masters Degree Student – UNIVERSITAS INDONESIA
Sep 2021 - Sep 2023

Engineering Administration and Project Control Staff - PT. ISTAKA KARYA
Aug 2019 - Sep 2021

Project Control Intern - PT. ISTAKA KARYA
Feb 2019 - Jul 2019

Surveying Laboratory Assistant – UNIVERSITAS TRISAKTI
Jul 2017- Agust 2019

Case Study

Rakamin Academy

Data scientist in Kalbe Nutritionals got a new project from :

- ***Inventory Team*** to predict sum of quantity from all products, so they could create sufficient daily inventory.
- ***Marketing Team*** to create cluster or segment of customer to get personalized promotion and sales treatment:

Tool & Library Used

The challenges in this project include :

- Perform **Data Ingestion** into dbeaver and PostgreSQL.
- Perform **Exploratory Data Analysis (EDA)** to know the average age of customer based on marital status and gender, also the best-selling store and product name.
- **Visualizing Data Dashboard** using Tableau
- Make **Machine Learning Regression** Model (Time Series) with ARIMA
- Make **Machine Learning Clustering** Model with KMeans Algorithm

Data Ingestion & Exploratory Data Analysis

Exploratory Data Analysis with PostgreSQL using DBeaver

Rakamin Academy

Query 1 :

Average customer age based on their marital status

```
select "Marital Status",
avg(age) as Age_Average
from customer c
group by "Marital Status";
```

Marital Status	age_average
	31.3333333333
Married	43.0382352941
Single	29.3846153846

Query 2 :

Average customer age based on their gender

```
select
    case
        when gender = 0 then 'Wanita'
        when gender = 1 then 'Pria'
        else '-'
    end as gender,
    avg(age) as Age_Average
    from customer c
    group by gender;
```

gender	age_average
Wanita	40.326446281
Pria	39.1414634146

Exploratory Data Analysis with PostgreSQL using DBeaver

Rakamin Academy

Query 3 :

Store name with the highest total quantity

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

storename	total_quantity
Lingga	2,777

Query 4 :

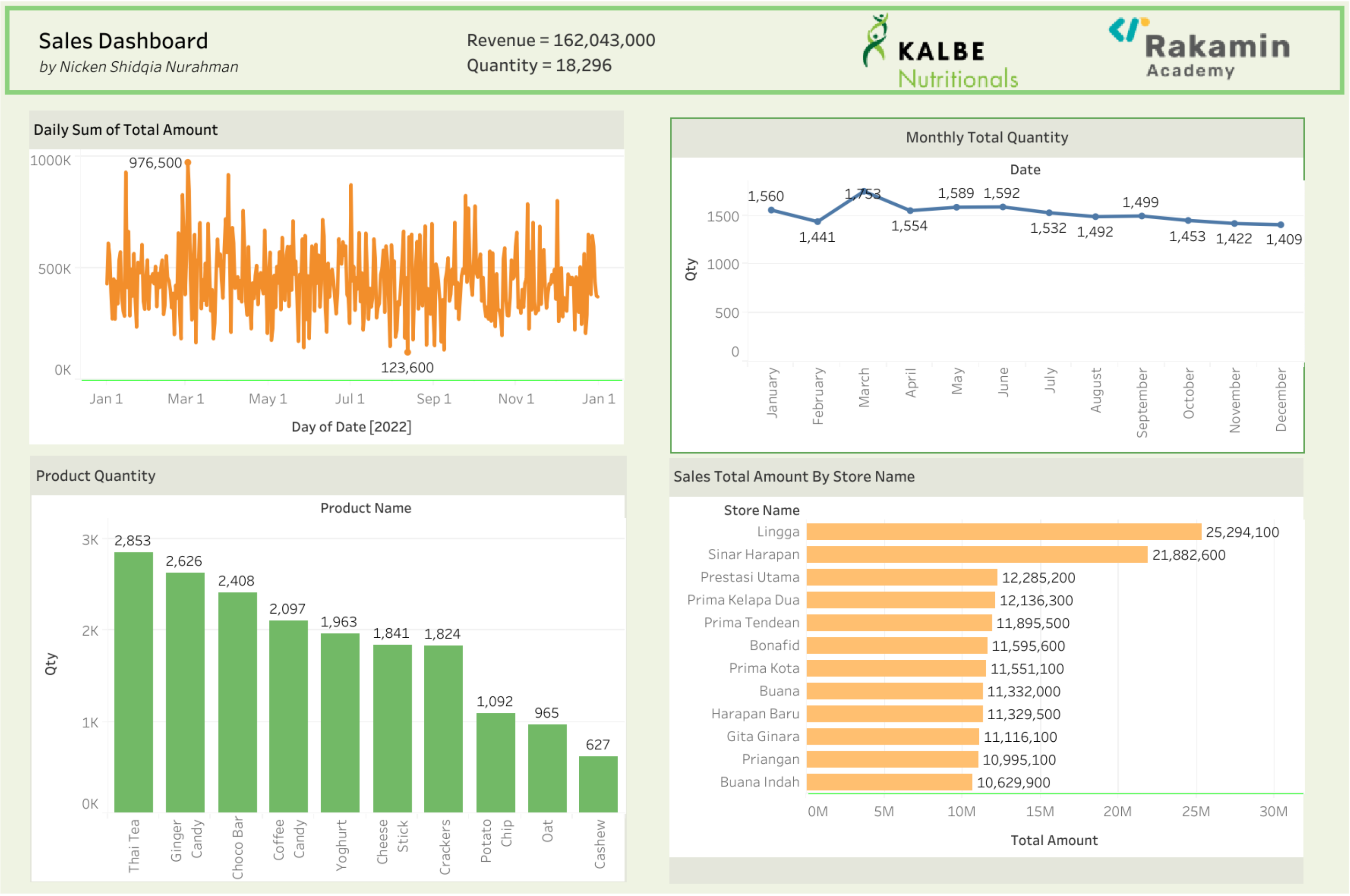
the best-selling product with the highest total amount

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

Product Name	total_amount
Cheese Stick	27,615,000

Data Visualization

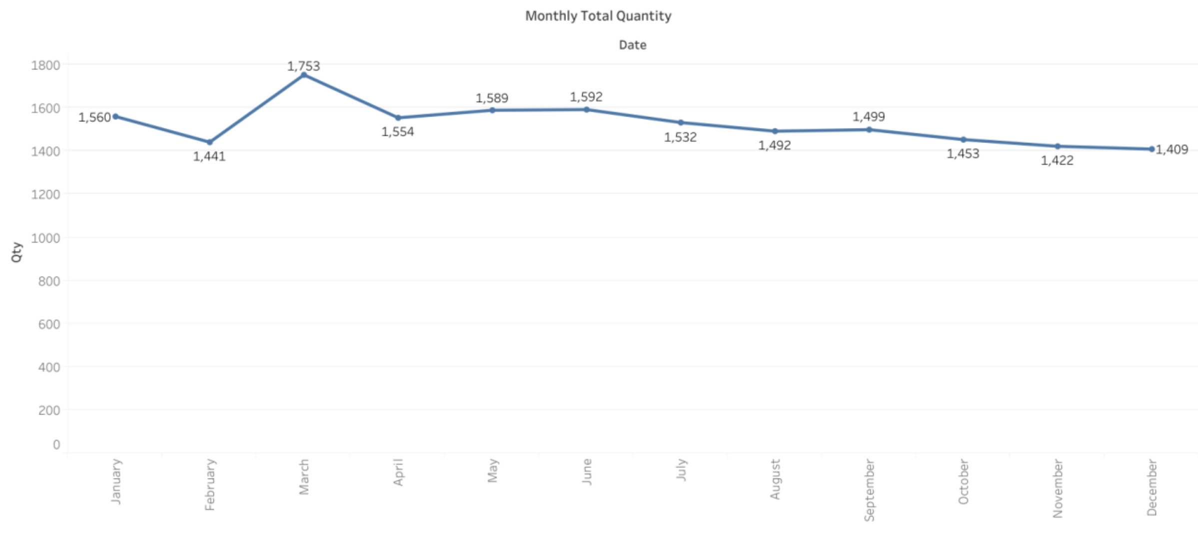
Dashboard in Tableau



Data Visualization in Tableau

Worksheet 1 :

Total quantity from month to month



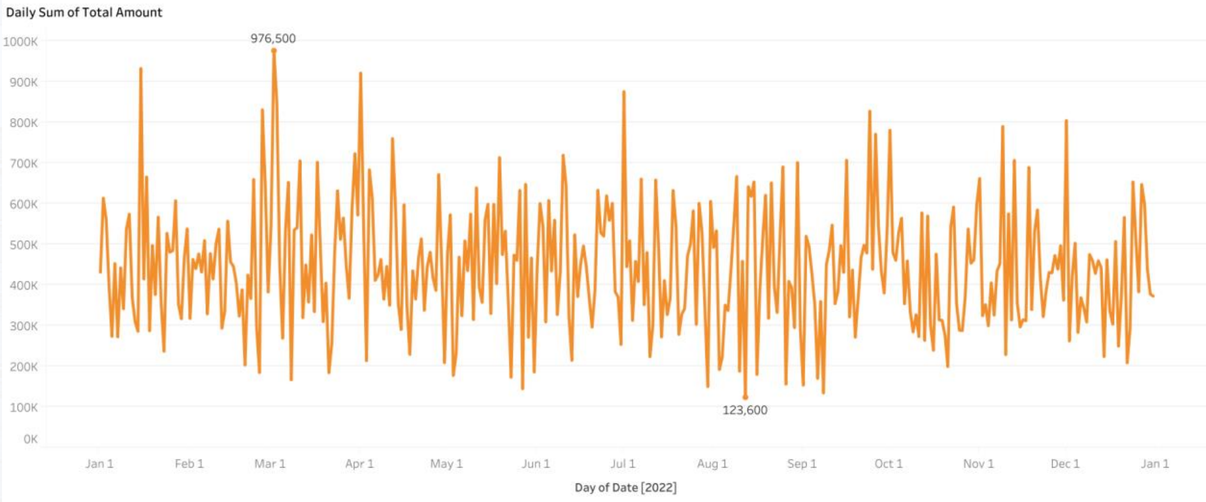
Key Insight :

- Sales trends fluctuate slightly, and show a gradual decline starting from June.
- The highest total quantity sold is in March 2022 with 1,753 items
- The lowest total quantity sold is in December 2022 with 1,409 items

Data Visualization in Tableau

Worksheet 2 :

Total amount from day to day



Key Insight :

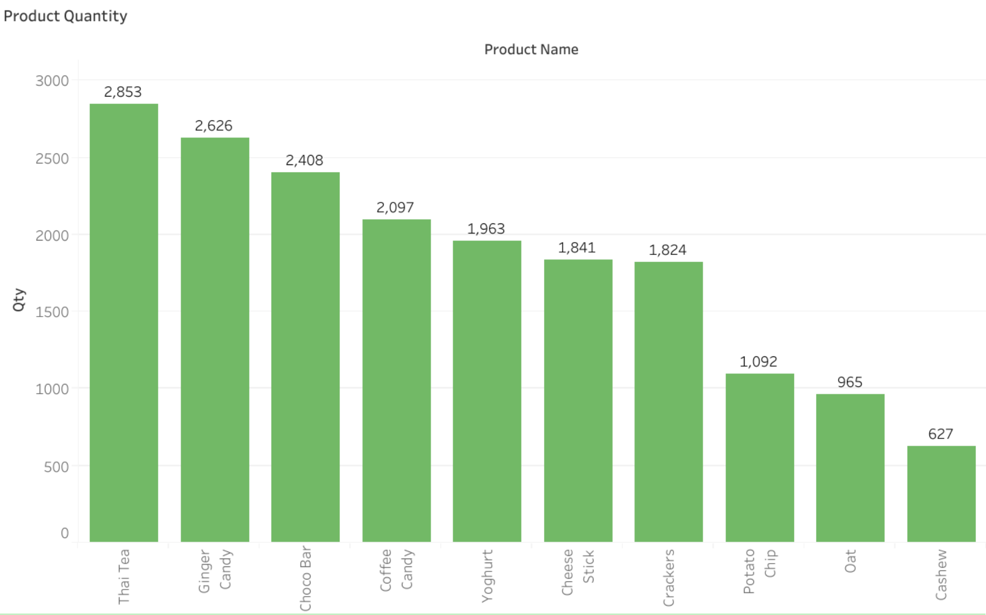
- Daily revenue trends fluctuate heavily
- The highest total amount is in March 2022 with Rp 976,500
- The lowest total amount is in August 2022 with Rp 123,600

Data Visualization in Tableau

Rakamin Academy

Worksheet 3 :

Total quantity by product



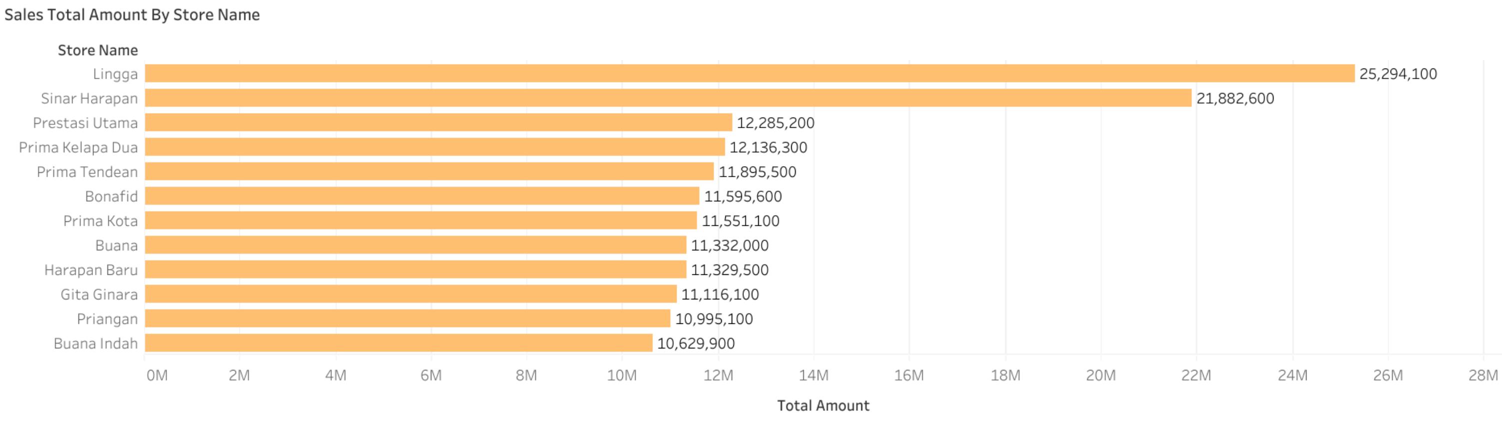
Key Insight :

- The highest selling product in 2022 is Thai Tea with 2,853 items sold.
- The lowest selling product in 2022 is Cashew with 627 items sold.

Data Visualization in Tableau

Worksheet 4 :

Total sales amount by store name



Key Insight :

- The best-selling store in 2022 is Lingga with sales revenue reached Rp 25,294,100.
- The lowest-selling store in 2022 is Buana Indah with sales revenue reached Rp 10,629,900.

Daily Product Quantity Prediction Using Time Series ARIMA

Data Cleaning

Data type error

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

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

Before :

	StoreID	StoreName	GroupStore	Type	Latitude	Longitude
0	1	Prima Tendean	Prima	Modern Trade	-6,2	106,816666

After :

	StoreID	StoreName	GroupStore	Type	Latitude	Longitude
0	1	Prima Tendean	Prima	Modern Trade	-6.200000	-6.200000
1	2	Prima Kelapa Dua	Prima	Modern Trade	-6.914864	-6.914864
2	3	Prima Kota	Prima	Modern Trade	-7.797068	-7.797068
3	4	Gita Ginara	Gita	General Trade	-6.966667	-6.966667

Before :

	CustomerID	Age	Gender	Marital Status	Income
0	1	55	1	Married	5,12

After :

	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

- Perbaikan data type error dari (,) menjadi (.) sebagai float pada Latitude, Langitude, dan Income

Data Cleaning

Data type error

Before :

```
df_transaction.dtypes
```

```
TransactionID     object
CustomerID         int64
Date              object
ProductID         object
Price              int64
Qty                int64
TotalAmount        int64
```

After :

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

```
TransactionID             object
CustomerID                 int64
Date              datetime64[ns]
```

Perbaikan data type error dari object menjadi datetime pada kolom Date

Missing Values

Before :

```
df_customer.isnull().sum()
```

```
CustomerID        0
Age               0
Gender            0
Marital Status    3
Income            0
```

After :

```
df_customer = df_customer.dropna()
```

```
df_customer.isnull().sum()
```

```
CustomerID        0
Age               0
Gender            0
Marital Status    0
Income            0
```

Drop missing values pada Marital Status sebanyak 3 row

Data Merging

Rakamin Academy

```
df_merge = pd.merge(df_customer, df_transaction, on = ['CustomerID'])
df_merge = pd.merge(df_merge, df_store, on = ['StoreID'])
df_merge = pd.merge(df_merge, df_product.drop(columns = ['Price']), on = ['ProductID'])
df_merge = df_merge.sort_values(by='Date').reset_index(drop = True)
df_merge.head()
```

	CustomerID	Age	Gender	Marital Status	Income	TransactionID	Date	ProductID	Price	Qty	TotalAmount	StoreID	StoreName	GroupStore
0	183	27	1	Single	0.18	TR1984	2022-01-01	P1	8800	4	35200	4	Gita Ginara	Gita
1	49	44	1	Married	13.48	TR67455	2022-01-01	P5	4200	3	12600	13	Buana	Buana
2	233	43	1	Married	5.69	TR97336	2022-01-01	P7	9400	2	18800	12	Prestasi Utama	Prestasi
3	287	36	0	Single	3.70	TR76340	2022-01-01	P4	12000	4	48000	12	Prestasi Utama	Prestasi
4	123	34	0	Married	4.36	TR99839	2022-01-01	P2	3200	6	19200	1	Prima Tendean	Prima

New Data Frame

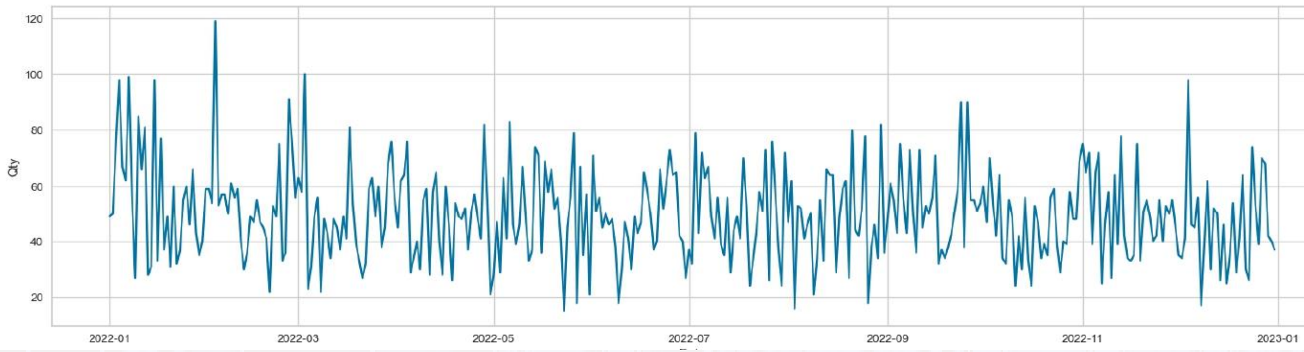
Rakamin Academy

```
df_regression = df_merge.groupby(['Date']).agg({'Qty': 'sum'}).reset_index()
df_regression
```

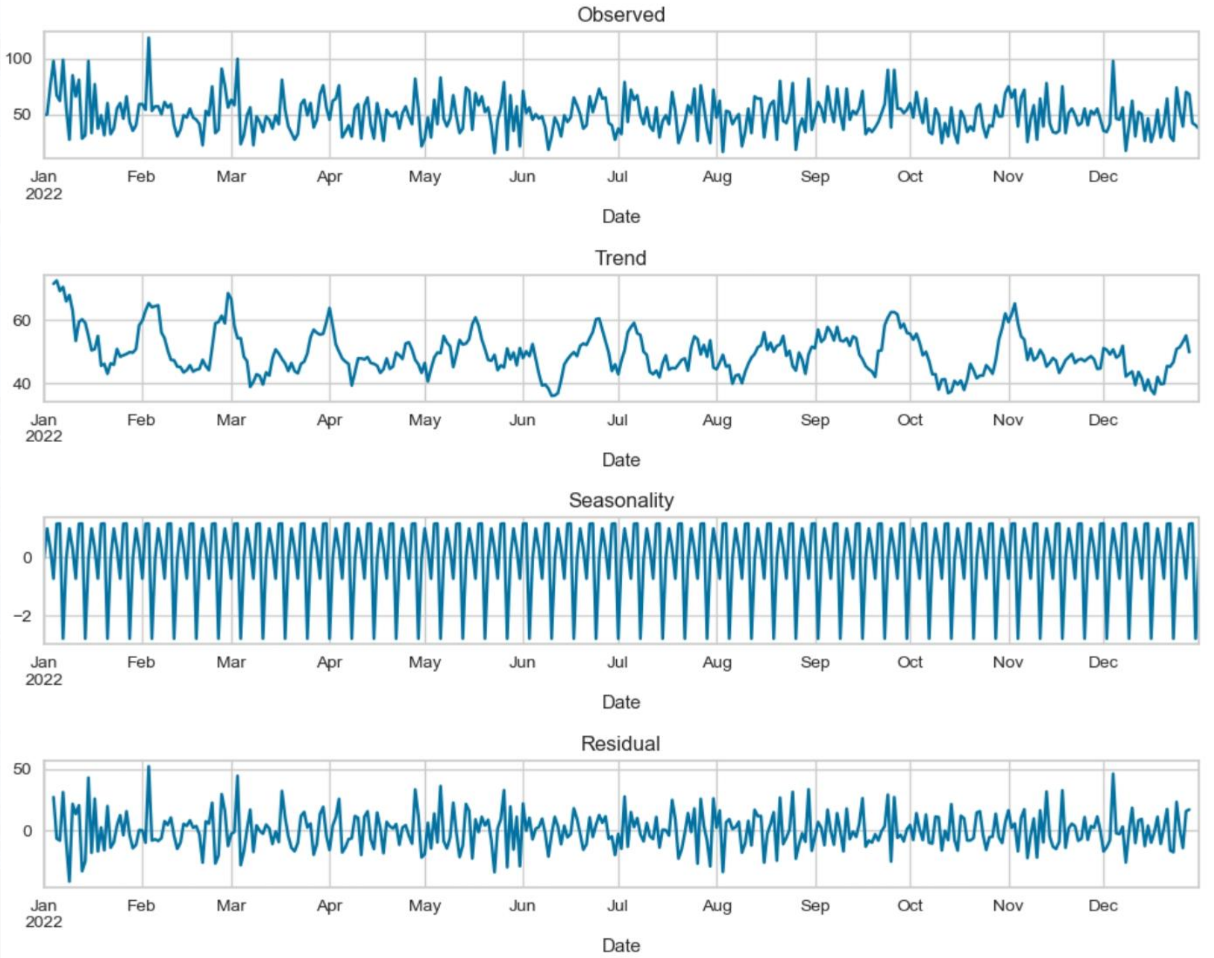
	Date	Qty
0	2022-01-01	49
1	2022-01-02	50
2	2022-01-03	76
3	2022-01-04	98
4	2022-01-05	67

Mengelompokkan data berdasarkan *Date* dengan *Total Quantity*

```
plt.figure(figsize=(20,5))
sns.lineplot(data=df_regression , x=df_regression['Date'] , y=df_regression['Qty'])
```



Seasonal Decomposition




```
reg_decompose = seasonal_decompose(df_regression.set_index('Date'))
plt.figure(figsize = (10,8))

plt.subplot(411)
reg_decompose.observed.plot(ax = plt.gca())
plt.title('Observed')

plt.subplot(412)
reg_decompose.trend.plot(ax = plt.gca())
plt.title('Trend')

plt.subplot(413)
reg_decompose.seasonal.plot(ax = plt.gca())
plt.title('Seasonality')

plt.subplot(414)
reg_decompose.resid.plot(ax = plt.gca())
plt.title('Residual')

plt.tight_layout()
```

Berdasarkan Trend, Seasonality, dan Residual: cukup fluktuatif dan mengindikasikan adanya downtrend dalam penjualan

Stationary Data Check

Rakamin Academy

Augmented Dicky-Fuller (ADF) Test

```python
def adf_test(dataset):
    dftest = adfuller(dataset, autolag = 'AIC')
    print("1. ADF Statistic: ", dftest[0])
    print("2. p-value: ", dftest[1])
    print("3. Num of Lags: ", dftest[2])
    print("4. Num of observation used for ADF Regression: ", dftest[3])
    print("5. Critical Values: ")
    for key, val in dftest[4].items():
        print("\t",key, ":", val)


adf_test(df_regression['Qty'])
```

```
1. ADF Statistic:  -19.09151387240814
2. p-value:  0.0
3. Num of Lags:  0
4. Num of observation used for ADF Regression:  364
5. Critical Values:
	 1% : -3.4484434475193777
	 5% : -2.869513170510808
	 10% : -2.571017574266393
```

H0 : Data is non-stationary
H1 : Data is stationary
If p-value < 0.05, then reject the H0 hypothesis

Based on ADF statistic, the p-value is $0.0 < 0.05$, then reject H0 and accept H1. Therefore data is stationary.

Data Training & Testing

Splitting data

80% Training, 20% Testing

```
split_size = round(df_regression.shape[0] * 0.8)

data_train = df_regression[:split_size]
data_test = df_regression[split_size:].reset_index(drop = True)
data_train.shape , data_test.shape
```

```
((292, 2), (73, 2))
```

Data Training

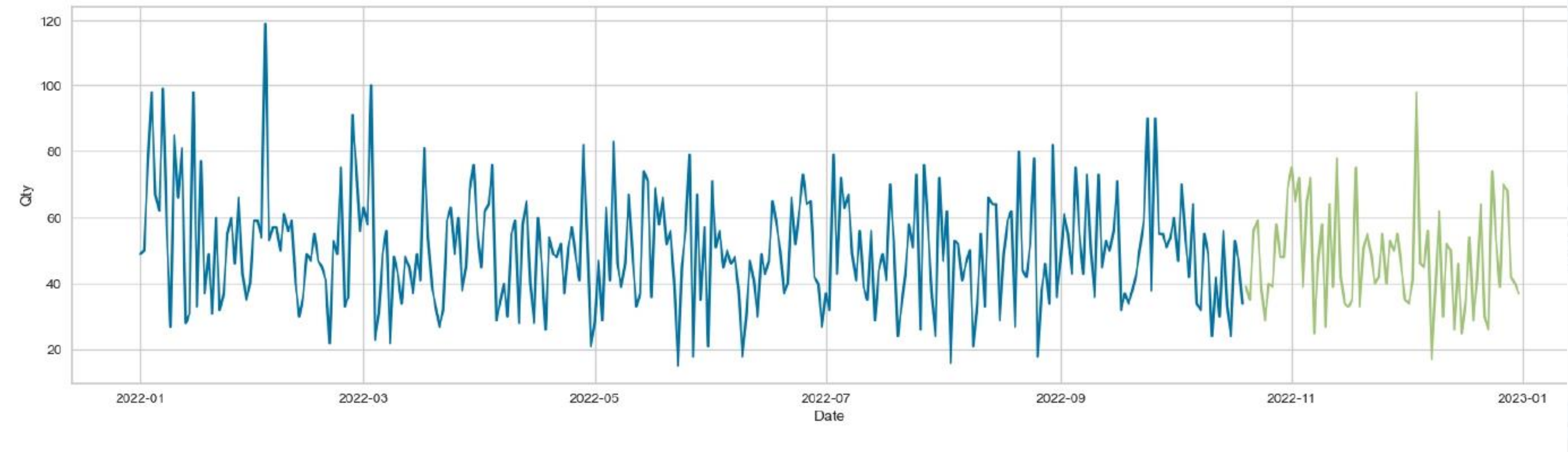
	Date	Qty
0	2022-01-01	49
1	2022-01-02	50
2	2022-01-03	76
3	2022-01-04	98
4	2022-01-05	67

Data Testing

	Date	Qty
0	2022-10-20	39
1	2022-10-21	35
2	2022-10-22	56
3	2022-10-23	59
4	2022-10-24	39

akamin
academy

```
plt.figure(figsize=(20,5))
sns.lineplot(data=data_train , x=data_train['Date'] , y=data_train['Qty'])
sns.lineplot(data=data_test, x=data_test['Date'], y=data_test['Qty'])
```



Find p,d,q & AIC for ARIMA Model

Rakamin Academy

Model 1 – Auto-fit ARIMA

```
auto_arima = pm.auto_arima(data_train, trace=True, seasonal=False, stepwise=False, suppress_warnings=True)
auto_arima.summary()
```

Dep. Variable:	y	No. Observations:	292
Model:	SARIMAX(1, 0, 1)	Log Likelihood	-1244.943
Date:	Mon, 23 Oct 2023	AIC	2495.886
Time:	06:09:33	BIC	2506.916
Sample:	01-01-2022	HQIC	2500.304
	- 10-19-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	1.0000	4.8e-05	2.08e+04	0.000	1.000	1.000
ma.L1	-0.9830	0.016	-60.722	0.000	-1.015	-0.951
sigma2	290.0520	24.022	12.074	0.000	242.969	337.135

Ljung-Box (L1) (Q):	0.11	Jarque-Bera (JB):	8.00
Prob(Q):	0.74	Prob(JB):	0.02
Heteroskedasticity (H):	0.70	Skew:	0.39
Prob(H) (two-sided):	0.08	Kurtosis:	3.24

ARIMA
2 models = Auto Regression (AR) & Moving Average (MA)
1 method = Differencing (I)

p = AR = The number of Autoregressive terms
d = I = The number of nonseasonal differences
q =MA =The number of lagged forecast errors

```
Best model:  ARIMA(1,0,1)(0,0,0)[0]
Total fit time: 9.813 seconds
```

AIC = 2495.88

ARIMA (1,0,1) artinya tidak ada Differencing (0) karena stasioner, dengan Autoregression pada rangkaian 1 lag dan 1 order Moving Average diterapkan.

Autocorrelation function (ACF) & Partial Autocorrelation Function (PACF)

Model 2 – ACF & PACF Plot

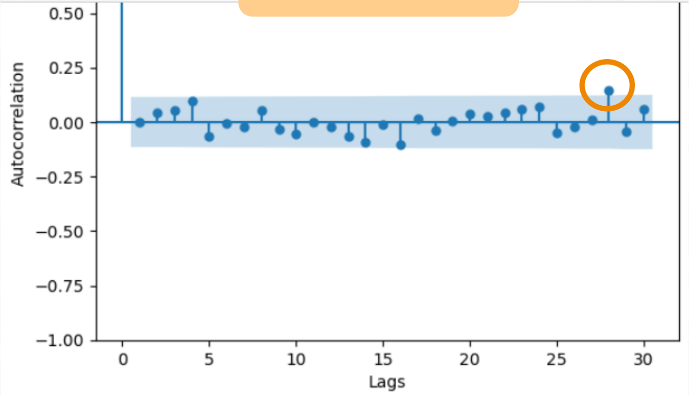
```
plot_acf(data_train, lags=30)
plt.xlabel('Lags')
plt.ylabel('Autocorrelation')
plt.title('Autocorrelation Function (ACF)')


plot_pacf(data_train, lags=30)
plt.xlabel('Lags')
plt.ylabel('Autocorrelation')
plt.title('Partial Autocorrelation (PACF)')
plt.show()
```

```
                                SARIMAX Results
==============================================================================
Dep. Variable:                    qty   No. Observations:                  292
Model:               ARIMA(28, 0, 28)   Log Likelihood               -1207.646
Date:                Mon, 23 Oct 2023   AIC                           2531.291
Time:                        06:10:25   BIC                           2744.543
Sample:                    01-01-2022   HQIC                          2616.711
                         - 10-19-2022
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
```

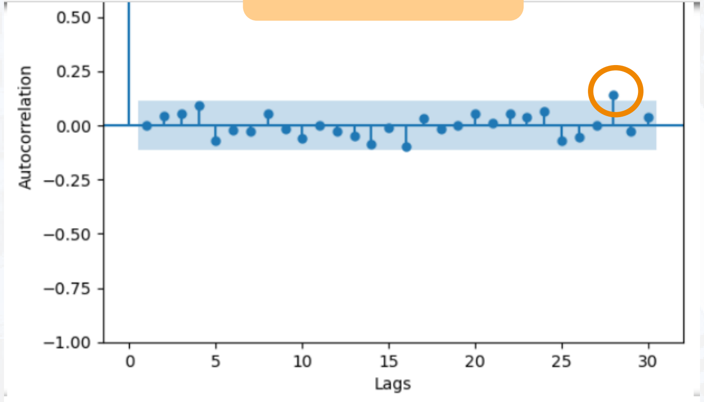
Tail off at pattern ACF --> AR Model = 28^{th} lag
Tail off at pattern PACF --> MA Model = 28^{th} lag

Conclusion :
p,d,q = 28,0,28
AIC = 2531.291

ACF



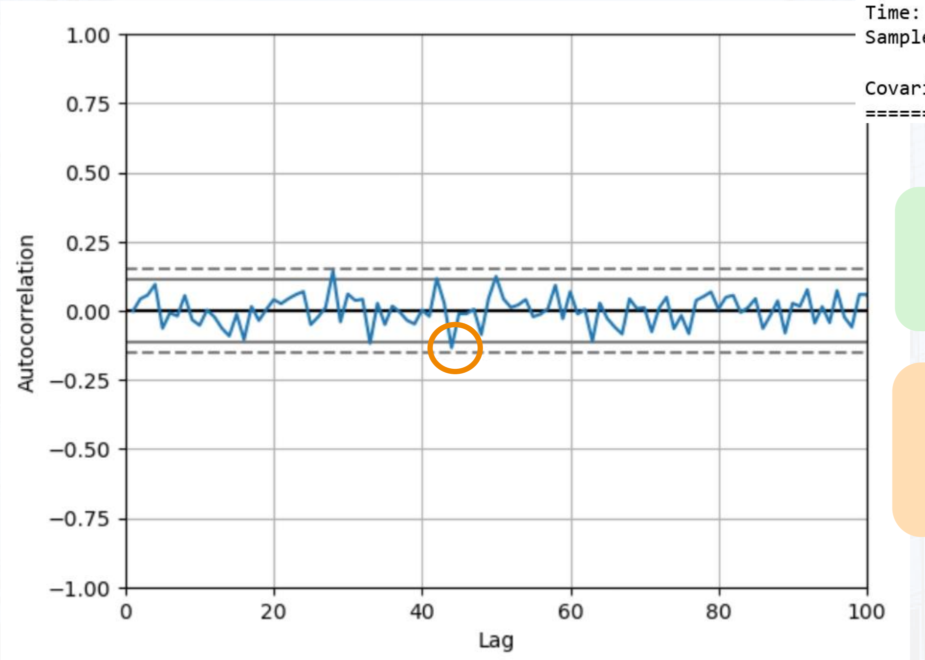
PACF



Find p,d,q & AIC for ARIMA Model

Model 3 – Autocorrelation Plot

```
autocorrelation_plot(data_train).set_xlim([0,100])
```




```
                               SARIMAX Results
==============================================================================
Dep. Variable:                    qty   No. Observations:                  292
Model:               ARIMA(44, 0, 44)   Log Likelihood               -1189.831
Date:                Mon, 23 Oct 2023   AIC                           2559.662
Time:                        06:13:01   BIC                           2890.570
Sample:                    01-01-2022   HQIC                          2692.211
                         - 10-19-2022
Covariance Type:                  opg
==============================================================================
```

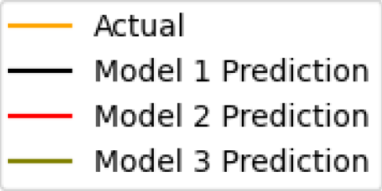
Garis keluar pada autocorrelation dari Threshold levels = 44th lag

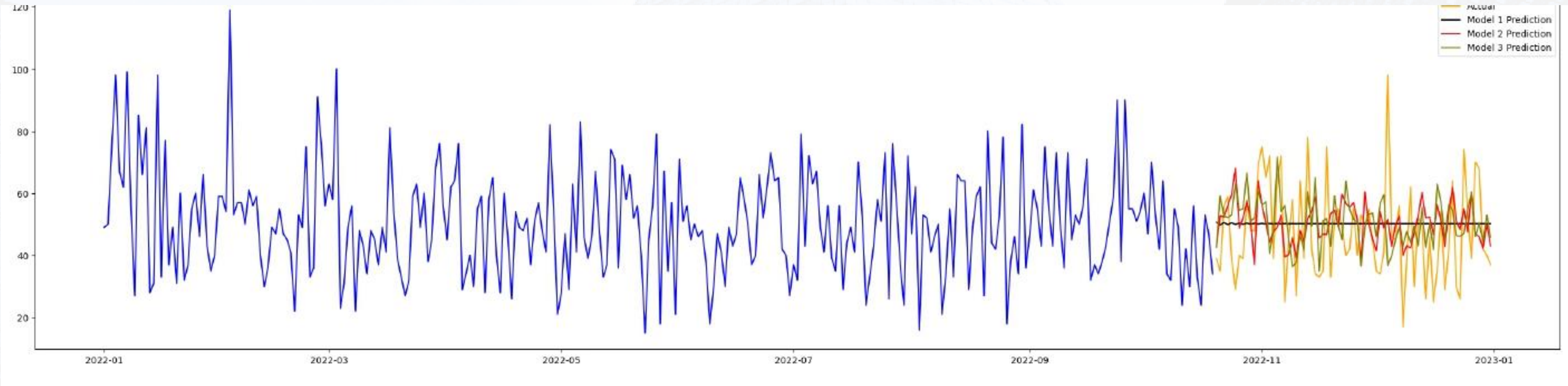
Conclusion :
p,d,q = 44,0,44
AIC = 2559.66

ARIMA Modelling

Plot Data Train, Test, and Model Prediction

```python
# Data train & Data test forecast
forecast1 = model1.get_forecast(steps = len(data_test))
forecast2 = model2.get_forecast(steps = len(data_test))
forecast3 = model3.get_forecast(steps = len(data_test))

df_forecast1 = forecast1.conf_int()
df_forecast1['Predictions'] = model1.predict(start=df_forecast1.index[0], end=df_forecast1.index[-1])
df_forecast1_out = df_forecast1['Predictions']
```

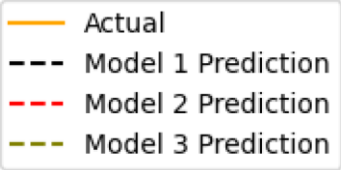


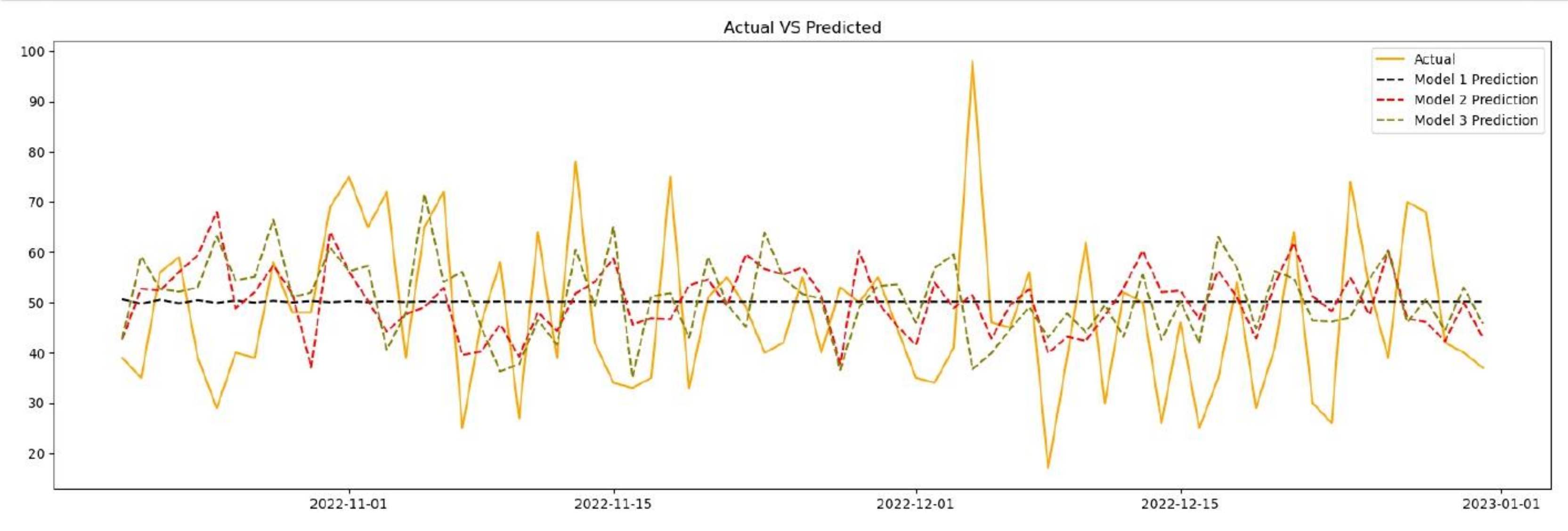


ARIMA Modelling

Plot Data Test, and Model Prediction

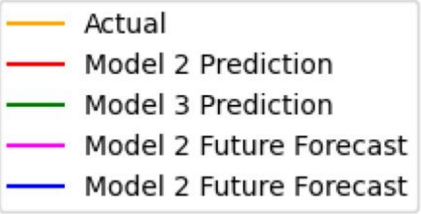
```
plt.figure(figsize=(20,6))
plt.plot(data_test.index, data_test['qty'], label='Actual', color='orange')
plt.plot(data_test.index, df_forecast1_out, label = 'Model 1 Prediction', color ='black', linestyle='--')
plt.plot(data_test.index, df_forecast2_out, label = 'Model 2 Prediction', color ='red', linestyle='--')
plt.plot(data_test.index, df_forecast3_out, label = 'Model 3 Prediction', color ='olive', linestyle='--')
plt.legend()
plt.title('Actual VS Predicted')
plt.show()
```



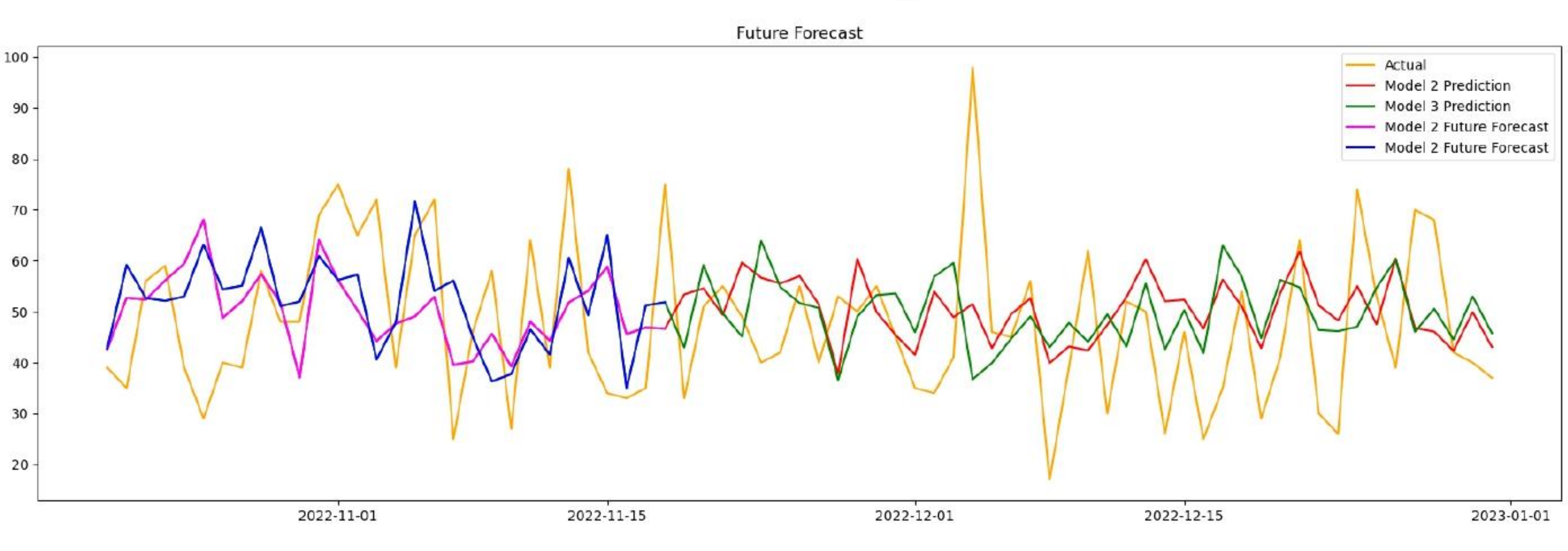


ARIMA Modelling

Plot Future Forecasting




```python
forecast_period = 30

future_model2 = model2.get_forecast(steps=forecast_period)
df_future_model2 =future_model2.conf_int()
df_future_model2['Predictions'] = future_model2.predicted_mean
df_future_model2.index = pd.date_range(start = data_train.index[-1], periods = forecast_period + 1, closed = 'right')
df_future_model2_out = df_future_model2['Predictions']
```



MAE, MSE, RMSE, MAPE

Rakamin Academy

```
mae1 = mean_absolute_error(data_test,df_forecast1_out)
mse1 = mean_squared_error(data_test,df_forecast1_out)
rmse1 = np.sqrt(mse1)
mape1 = mean_absolute_percentage_error(data_test,df_forecast1_out)*100
```

```
Model 1
Mean Absolute Error (MAE) = 13.01
Mean Squared Error (MSE) = 246.15
Root Mean Squared Error (RMSE) = 15.69
Mean Absolute Percentage Error (MAPE) = 32.45%
```

```
Model 2
Mean Absolute Error (MAE) = 13.10
Mean Squared Error (MSE) = 255.45
Root Mean Squared Error (RMSE) = 15.98
Mean Absolute Percentage Error (MAPE) = 31.88%
```

```
Model 3
Mean Absolute Error (MAE) = 13.61
Mean Squared Error (MSE) = 290.53
Root Mean Squared Error (RMSE) = 17.04
Mean Absolute Percentage Error (MAPE) = 33.02%
```

- MAE menghitung berapa rata-rata kesalahan absolut dalam prediksi
- MSE menghitung berapa rata-rata kesalahan kuadrat dalam prediksi
- RMSE merupakan akar kuadrat dari MSE
- MAPE menghitung berapa rata-rata kesalahan dalam prediksi sebagai persentase dari nilai aktual
- *Semakin kecil nilai MAE, MSE, RMSE, dan MAPE, maka semakin baik kualitas model tersebut*

Kesimpulan :
Model 2 dengan p,d,q (28,0,28) menunjukkan metrik evaluasi terbaik

Forecast Quantity Sales with The Best Parameter

```
df_future_model2_out.describe()
```

```
count    30.000000
mean     50.116869
std       7.303344
min      37.041028
25%      45.600319
50%      49.702099
75%      53.784914
max      68.038147
Name: Predictions, dtype: float64
```

Prediksi untuk quantity pada January 2023 adalah 50 pcs/hari

Customer Segmentations Using Kmeans Clustering

Standarisasi & Normalisasi Dataset

Create New Dataset

```
clust = preclust.drop(columns = 'customerid')
clust.head()
```

	transactionid	qty	totalamount
0	17	60	623300
1	13	57	392300
2	15	56	446200
3	10	46	302500
4	7	27	268600

Standarisasi Dataset

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler
x = clust.values
x_std = StandardScaler().fit_transform(x)
df_std=pd.DataFrame(data=x_std, columns=clust.columns)
df_std.isna().sum()
```

```
transactionid    0
qty              0
totalamount      0
dtype: int64
```

Normalisasi Dataset

```
x_norm = MinMaxScaler().fit_transform(x)
x_norm
```

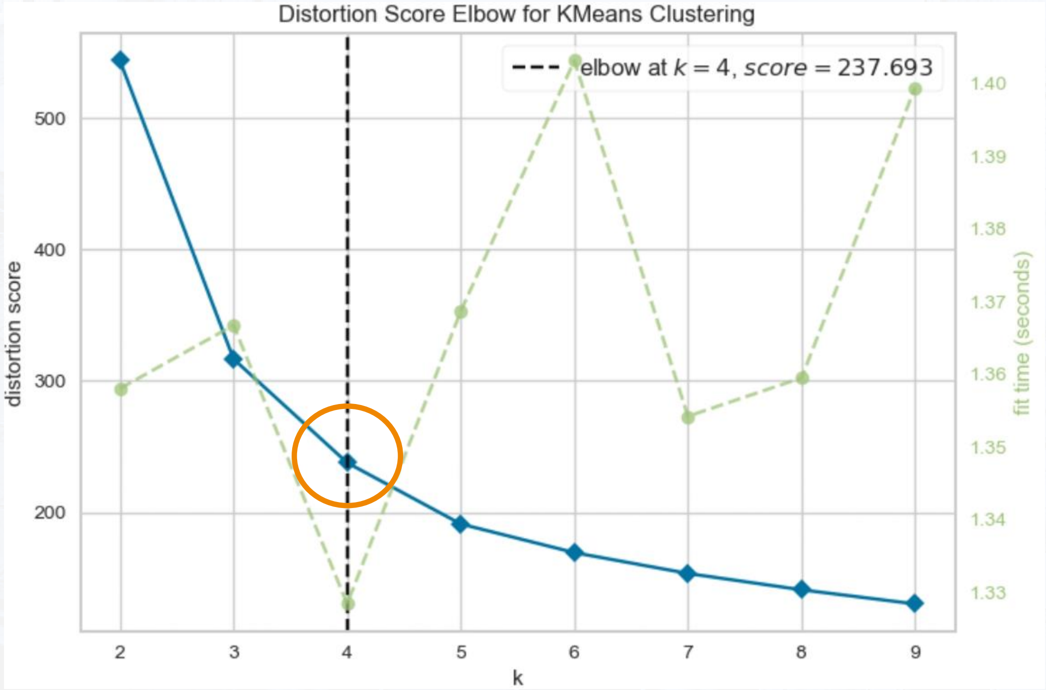
```
array([[0.77777778, 0.72463768, 0.70394911],
       [0.55555556, 0.68115942, 0.39782666],
       [0.66666667, 0.66666667, 0.46925523],
       ...,
       [0.83333333, 0.84057971, 0.6561092 ],
       [0.44444444, 0.46376812, 0.43890803],
       [0.55555556, 0.46376812, 0.46011132]])
```

Elbow Method

Rakamin Academy

Find k

```
from yellowbrick.cluster import KElbowVisualizer
visualizer = KElbowVisualizer(model1, k=(2,10))
visualizer.fit(x_std)
visualizer.show()
```



Input Cluster to dataset

```
clust['cluster']=kmeans_4.labels_
clust.head()
```

	transactionid	qty	totalamount	cluster
0	17	60	623300	2
1	13	57	392300	3
2	15	56	446200	3
3	10	46	302500	0
4	7	27	268600	1

KMeans Clustering Customer Segmentation


```
clust['customerID']=preclust['customerid'].astype('category')

#create scatter plot
plt.figure(figsize=(10,6))
sns.scatterplot(data=clust ,x='qty',y='totalamount', hue='cluster', palette='Spectral', sizes=70)
plt.xlabel('Quantity')
plt.ylabel('Total Amount')
plt.title('KMeans Clustering Customer Segmentation')
plt.legend(title='Cluster')
plt.show()
```

	customerID	qty	totalamount
cluster			
0	180	37.350000	325663.333333
3	115	49.121739	437241.739130
1	93	24.505376	208283.870968
2	56	62.035714	576716.071429

KMeans Clustering Customer Segmentation

Rakamin Academy

Cluster 0

Insight :

- Cluster 0 is the cluster with the most largest number of customers
- Cluster 0 has the second lowest average of quantity and total amount

Conclusion : Cluster 0 needs to get promotion

Strategy :

1. Give special offering and discount
2. Offer bundling product

Cluster 1

Insight :

- Cluster 1 is the cluster with the second fewest number of customers
- Cluster 1 has the lowest average of quantity and total amount

Conclusion : Cluster 1 is customer that need to give more brand awareness

Strategy :

1. Give special offering and discount for new member
2. Engage current customers to help attract new customers with referral codes
3. Collaborate with influencers to promote products

Cluster 2

Insight :

- Cluster 2 is the cluster with the smallest number of customers
- Cluster 2 has the largest average of quantity and total amount

Conclusion : Cluster 2 is the customer that valuable to the business

Strategy :

1. Offer bundling product
2. Offer loyalty membership

Cluster 3

Insight :

- Cluster 3 is the cluster with the second largest number of customers
- Cluster 3 has the second largest average of quantity and total amount

Conclusion : Cluster 3 is the customer that has potential of upselling

Strategy :

1. Offer bundling product
2. Offer loyalty membership

Thank You

Rakamin Academy X KALBE Nutritionals