

# Machine Learning Project (Regression Model and Clustering)

## Data Scientist

Presented by  
Imam Santoso

Batch September 2023



# Imam Santoso

## About You

Successful Computer Science graduate with expertise in data analytics. Skilled in Python, SQL, R, Tableau, and machine learning. Certified in Data Analysis, R Programming, and SQL. Experienced in Google Looker Studio. Committed to leveraging data for business growth and social impact.



## My Experiences

- **Alibaba Cloud Computing**  
Rakamin Academy Jun 2022 - August 2023
- **Short Class Data Analytics**  
Revou Mar 2023 - Apr 2023

# Challenge

- 1. Task 1 : Dbeaver Connection with PostgreSQL**
- 2. Task 2 : Tableau Public Create Dashboard**
- 3. Task 3 : Machine Learning Regression (Time Series)**
- 4. Task 4 : Machine Learning Clustering**

# Task 1 - Dbeaver

- query 1 : Berapa rata-rata umur customer jika dilihat dari marital statusnya ?

```
select
  "Marital Status" as marital_status,
  round(avg(age), 2) as avg_age
from
  "case study customer"
where
  "Marital Status" != ''
group by
  marital_status
order by
  avg_age asc
;
```

	asc marital_status ▼	123 avg_age ▼	
1	Single	29.38	
2	Married	43.04	

# Task 1 - Dbeaver

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

```
select
    gender,
    round(avg(age), 2) as avg_age
from
    customer
group by
    gender
order by
    avg_age asc
```

	<small>123</small> gender <small>↑↓</small>	<small>123</small> avg_age <small>↑↓</small>
1	1	39.14
2	0	40.33



# Task 1 - Dbeaver

- query 3 : Tentukan nama store dengan total quantity terbanyak!

```
select
    st.storename as store_name,
    sum(tr.qty) as quantity
from
    transaction as tr
join
    store as st
on
    tr.storeid = st.storeid
group by
    store_name
order by
    quantity desc
```

	asc store_name	123 quantity
1	Lingga	2,777
2	Sinar Harapan	2,588
3	Prestasi Utama	1,395
4	Prima Kota	1,358
5	Buana	1,320
6	Prima Tendean	1,310
7	Prima Kelapa Dua	1,296
8	Harapan Baru	1,286
9	Bonafid	1,283
10	Priangan	1,239
11	Gita Ginara	1,236
12	Buana Indah	1,208

# Task 1 - Dbeaver

- query 4 : Tentukan nama produk terlaris dengan total amount terbanyak!

```
select
    pr."Product Name" as product_name,
    sum(tr.totalamount) as total_amount
from
    product as pr
join
    transaction as tr
on
    pr.productid = tr.productid
group by
    product_name
order by
    total_amount desc
```

	asc product_name T	123 total_amount T
1	Cheese Stick	27,615,000
2	Choco Bar	21,190,400
3	Coffee Candy	19,711,800
4	Yoghurt	19,630,000
5	Oat	15,440,000
6	Crackers	13,680,000
7	Potato Chip	13,104,000
8	Thai Tea	11,982,600
9	Cashew	11,286,000
10	Ginger Candy	8,403,200

# Task 2 - Sales Dashboard

## Sales Dashboard by Imam Santoso

Select Month

January 2021 December 20

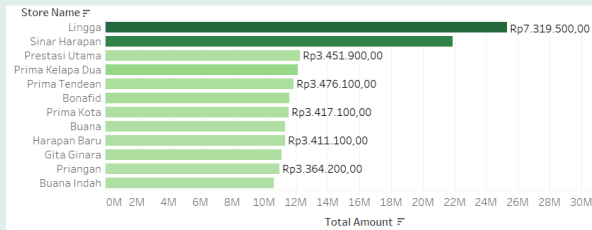
Select Day

(All)

Total Amount

Rp 162.043.000,00

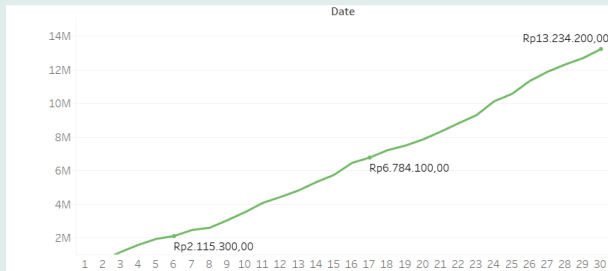
### Total Amount By Store Name



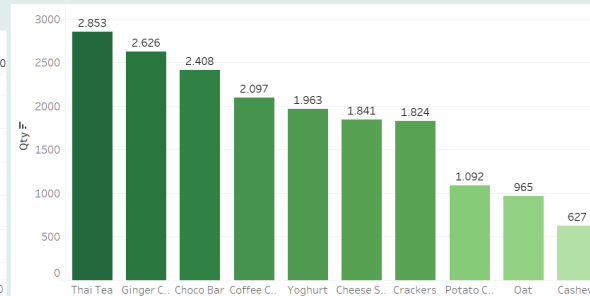
### Quantity Per Month



### Total Amount Per Day



### Quantity by Product







# Task 3 - Time Series

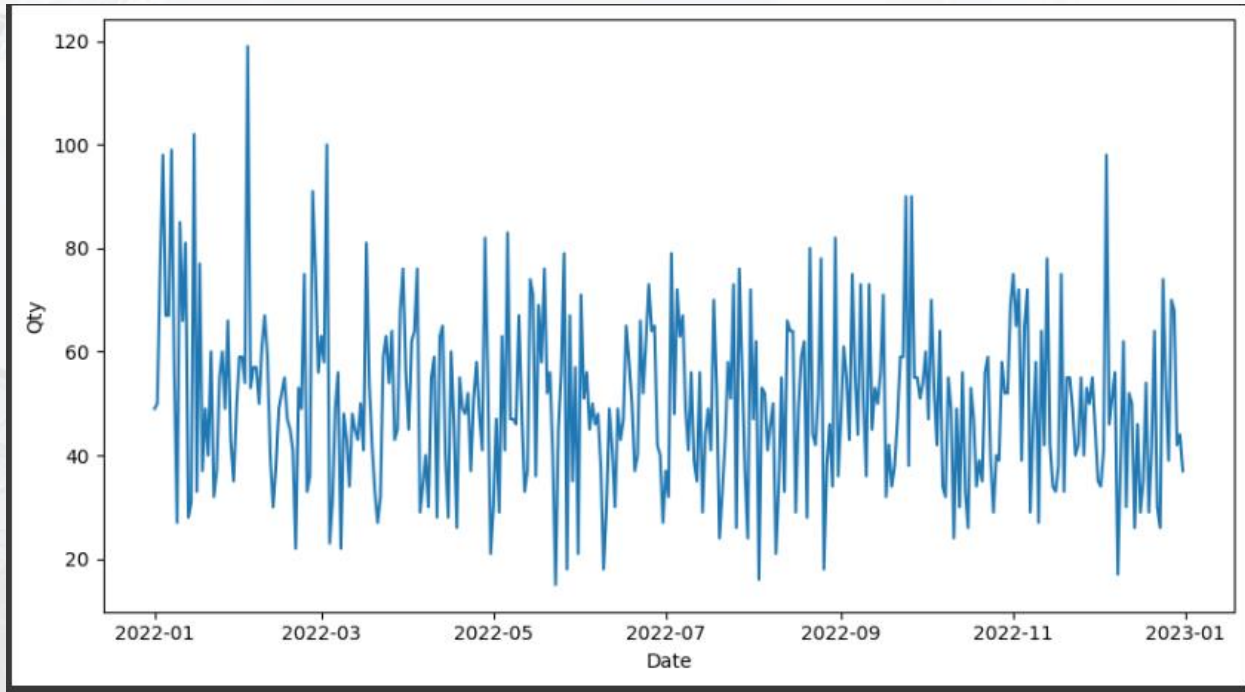
## Preview Data

```
# model regresi: time series
df_regresi = df_merge.groupby('Date').agg({'Qty': 'sum'}).reset_index()
df_regresi.head()
```

	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		

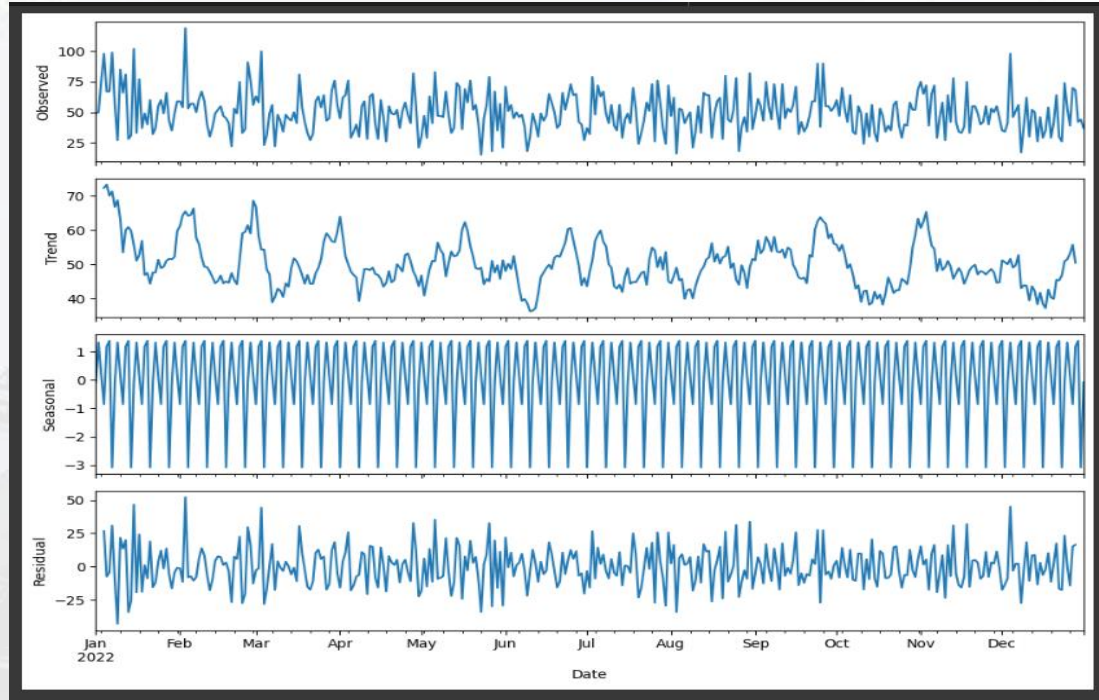
# Task 3 - Time Series

## Plot Data Time Series



# Task 3 - Time Series

## Plot Data Time Series





# Task 3 - Time Series



## Split Data Time Series

```
# Splitting the data into training and testing sets
train_size = int(len(df_regresi) * 0.8) # 80% data for training, 20% for testing
train_data, test_data = df_regresi.iloc[:train_size], df_regresi.iloc[train_size:]
```

train\_data.head()

	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	

test\_data.head()

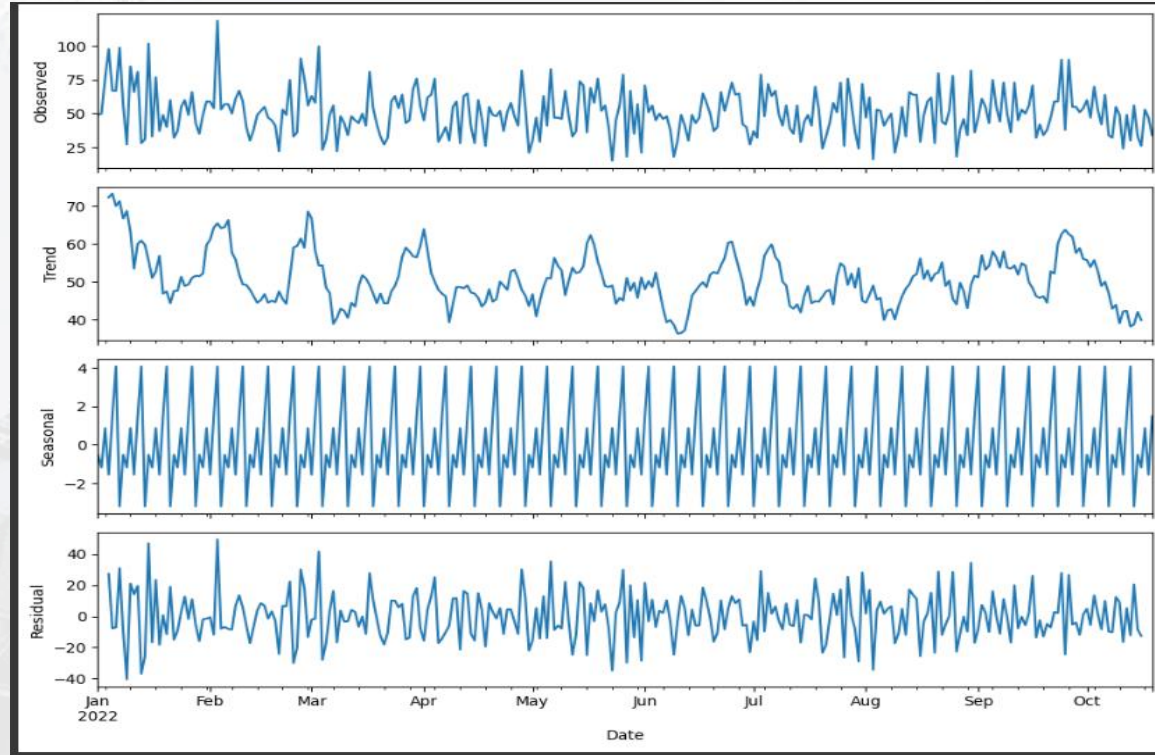
	Date	Qty	
292	2022-10-20	39	
293	2022-10-21	35	
294	2022-10-22	56	
295	2022-10-23	59	
296	2022-10-24	39	

Dilakukan split data train dan data test, data train menggunakan 80% dari keseluruhan data dan data test menggunakan 20% dari keseluruhan data



# Task 3 - Time Series

## Plot Data Train Time Series





# Task 3 - Time Series

## Uji Stasioner Data Train Time Series

```
# Statistical test to validate stationarity

# Ho = The data is not stationary
# Ha = The data is stationary

alpha = 0.05
adfuller_pvalue = adfuller(train_data['Qty'])[1]

if adfuller_pvalue <= alpha:
    print('Reject Ho. The data is stationary')
    print(adfuller_pvalue)
else:
    print('Fail to reject Ho. The data is not stationary')
    print(adfuller_pvalue)
```

```
Reject Ho. The data is stationary
8.939693654974982e-30
```

Uji ADF di atas menunjukkan bahwa data sudah stasioner. Akan tetapi, plot time series menunjukkan datanya belum stasioner secara musiman. Sehingga, akan dilakukan differencing tiap lag musimannya, yaitu lag 7.

# Task 3 - Time Series

## Differencing pada Lag Musiman Data Train Time Series

```
diff_mus7 = train_data['Qty'].diff(periods = 7)  
diff_mus7
```

```
0      NaN  
1      NaN  
2      NaN  
3      NaN  
4      NaN
```

```
...
```

```
287     1.0  
288    -29.0  
289     4.0  
290    23.0  
291   -15.0
```

```
Name: Qty, Length: 292, dtype: float64
```

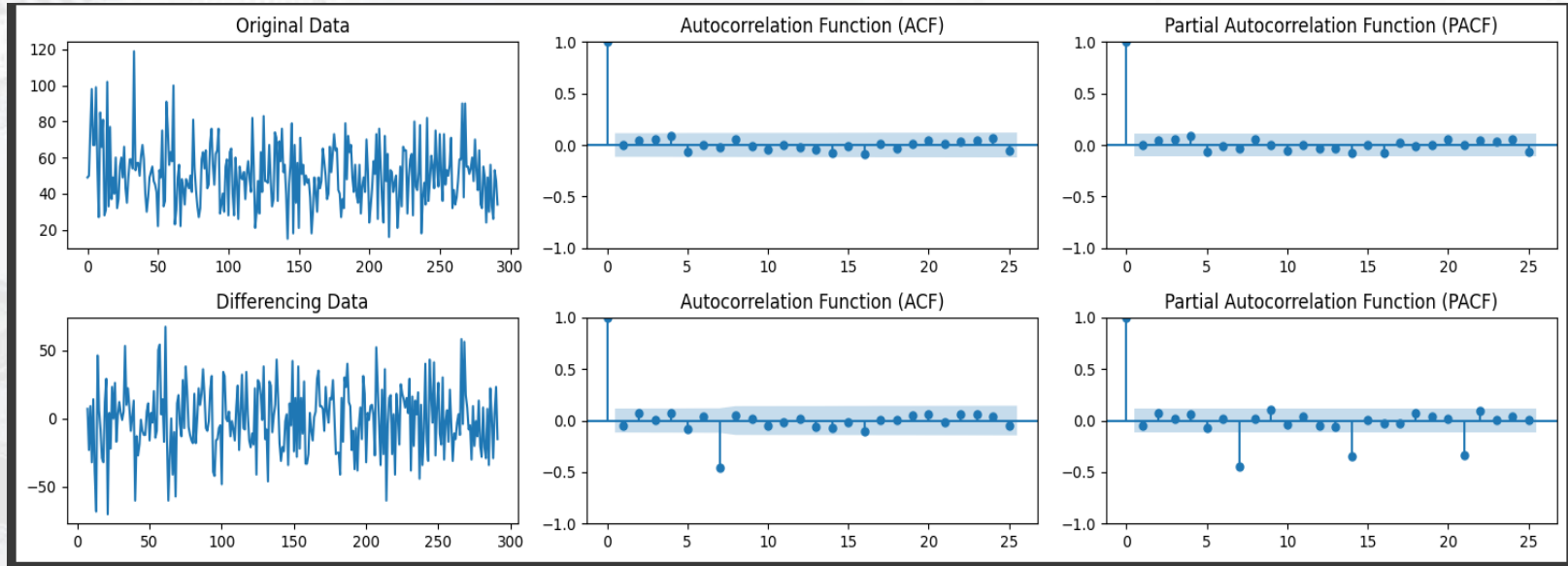
## Uji Stasioner

```
# Statistical test to validate stationarity  
  
# Ho = The data is not stationary  
# Ha = The data is stationary  
  
alpha = 0.05  
adfuller_pvalue = adfuller(diff_mus7.dropna())[1]  
  
if adfuller_pvalue < alpha:  
    print('Reject Ho. The data is stationary')  
    print(adfuller_pvalue)  
else:  
    print('Fail to reject Ho. The data is not stationary')  
    print(adfuller_pvalue)
```

```
Reject Ho. The data is stationary  
8.752600524789744e-13
```

# Task 3 – Time Series

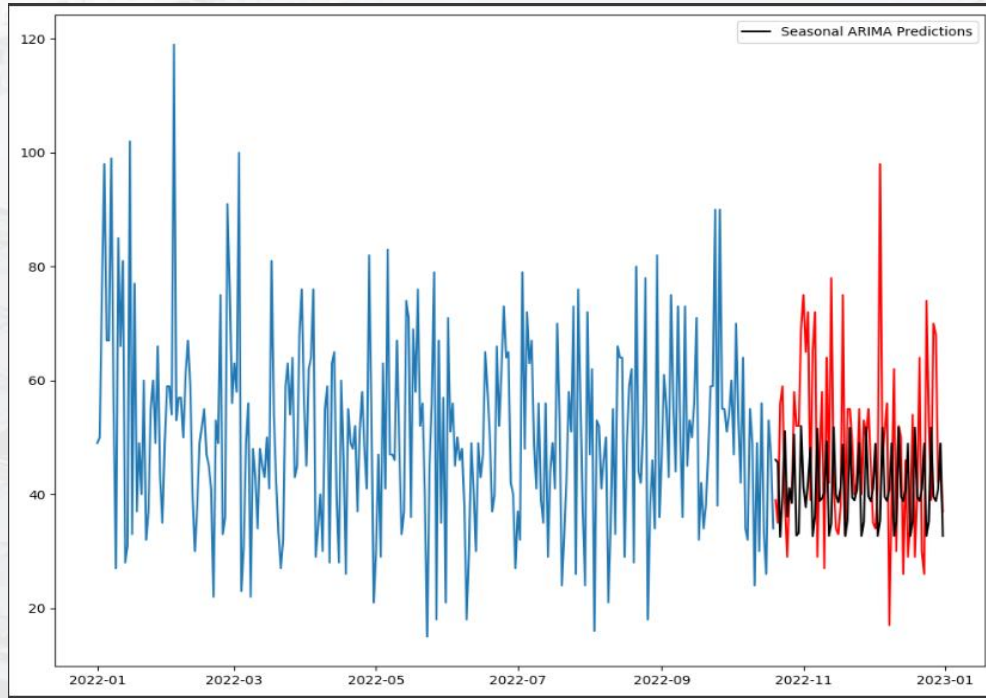
## Plot ACF dan PACF



Model Seasonal ARIMA yang terbentuk adalah ARIMA (0, 0, 0) dengan seasonal (0, 1, 1, 7). Akan tetapi, kita juga akan melakukan pengecekan untuk model gabungan MA = 1 dan/atau AR = 1, sehingga model yang mungkin terbentuk adalah model seasonal (0, 1, 1, 7), (1, 1, 0, 7), atau (1, 1, 1, 7).

# Task 3 - Time Series

## Plot Model Terbaik: Model ARIMA (0,0,0) Seasonal (1,1,0,7)



RMSE Value: 16.07272935224695

R-squared Value: -0.10396434594603399

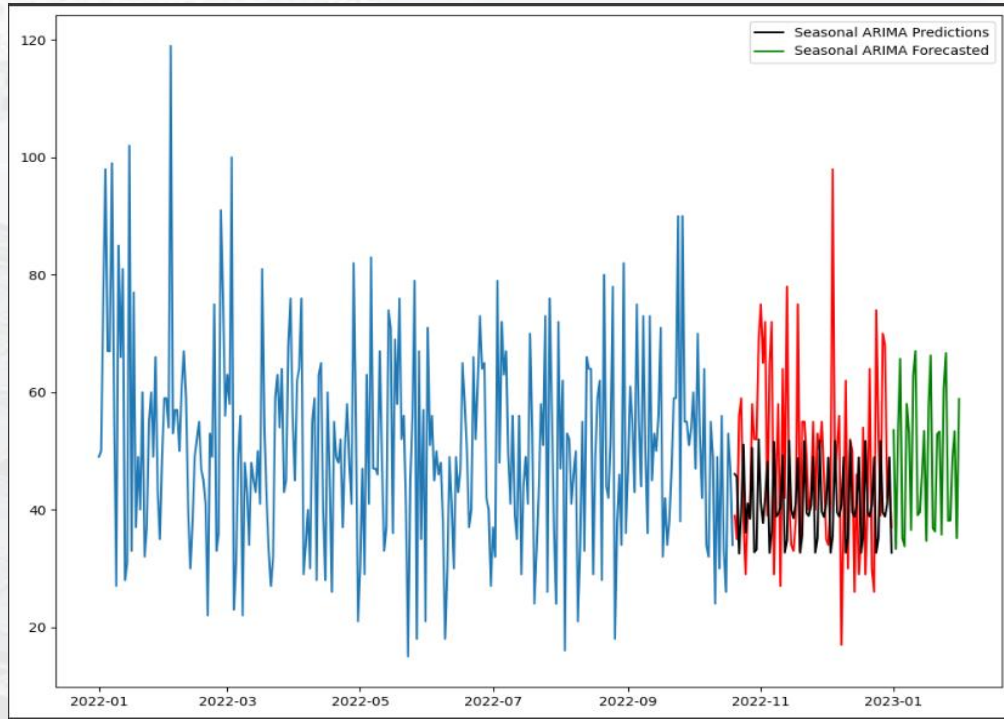
MAE Value: 13.358618452338368

Model ini memenuhi:

- (1) Parameter yang digunakan signifikan
- (2) Asumsi residual: white noise dan berdistribusi normal

# Task 3 - Time Series

**Forecast Model Terbaik: Model ARIMA (0,0,0) Seasonal (1,1,0,7)**



	lower Qty	upper Qty	forecasted Qty
Date			
2023-01-01	17.676142	89.460497	53.568319
2023-01-02	-2.575369	69.208986	33.316808
2023-01-03	17.626566	89.410921	53.518743
2023-01-04	29.834546	101.618901	65.726723
2023-01-05	-0.712008	71.072347	35.180170



# Task 4 - Kmeans Clustering

## Preview Data

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

df_cluster.head()
```

	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

# Task 4 - Kmeans Clustering

## Data Normalization

```
# normalization per feature
from sklearn.preprocessing import MinMaxScaler

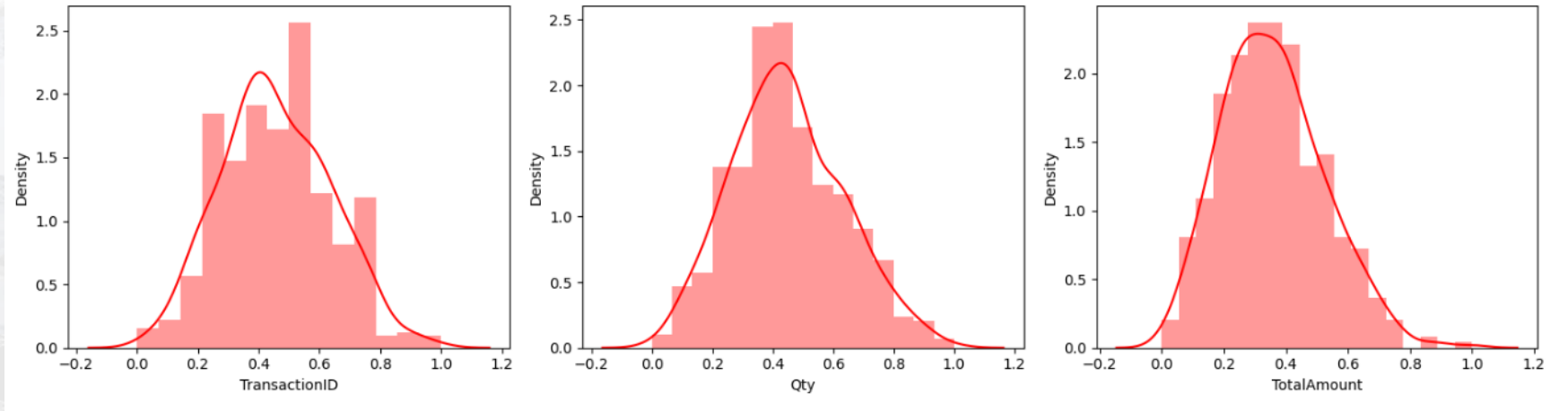
df_fix = df_cluster.drop('CustomerID', axis = 1)
num_fix = df_fix.columns

df_norm = MinMaxScaler().fit_transform(df_fix)
df_norm = pd.DataFrame(data = df_norm, columns = num_fix)
df_norm.head()
```

	TransactionID	Qty	TotalAmount
0	0.777778	0.724638	0.703949
1	0.555556	0.681159	0.397827
2	0.666667	0.666667	0.469255
3	0.388889	0.521739	0.278823
4	0.222222	0.246377	0.233899

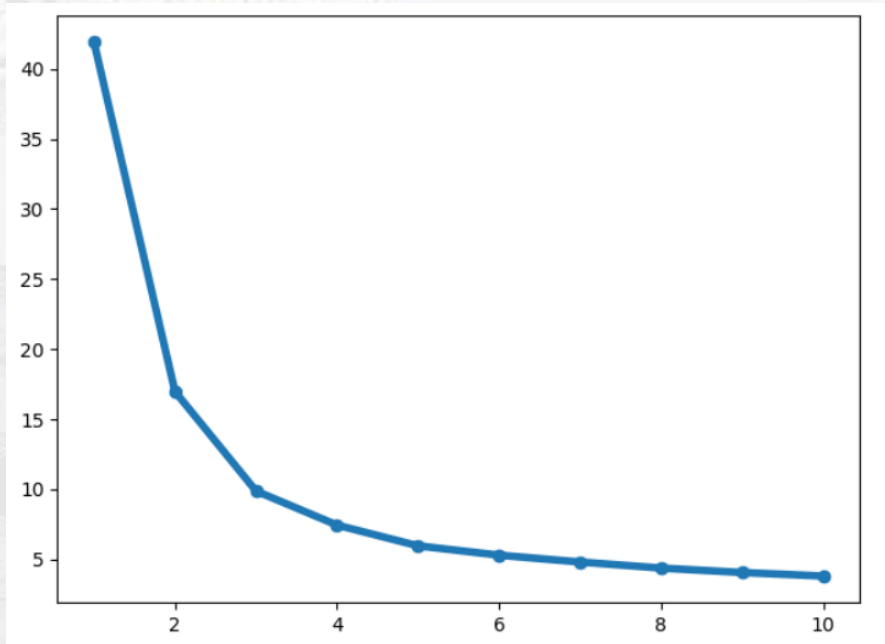
# Task 4 - Kmeans Clustering

## Data Normalization



# Task 4 - Kmeans Clustering

## Elbow Method



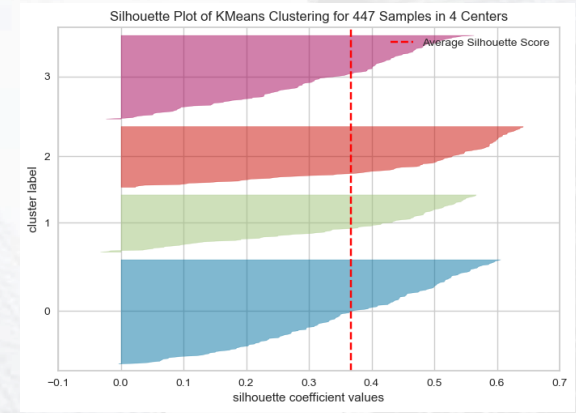
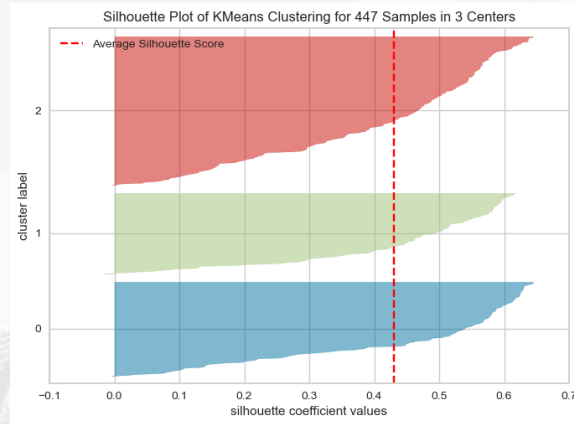
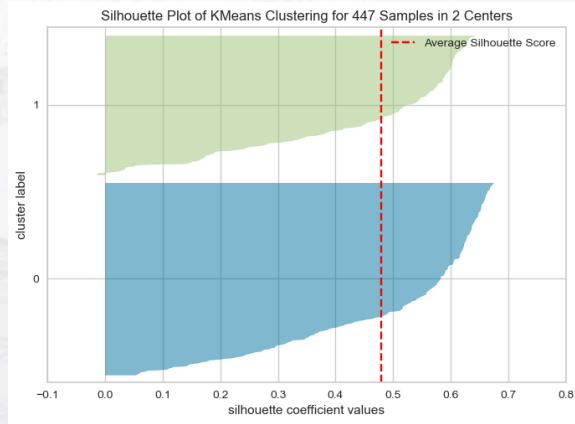
```
pd.Series(inertia) - pd.Series(inertia).shift(-1)
```

```
0    24.940793
1     7.095883
2     2.432542
3     1.481289
4     0.665881
5     0.489428
6     0.431677
7     0.316457
8     0.254853
9           NaN
dtype: float64
```

**Berdasarkan Elbow Method, jumlah cluster yang sesuai adalah 3 cluster.**

# Task 4 - Kmeans Clustering

## Silhouette Plot



For  $n\_clusters = 2$ , the silhouette score is 0.47981313905566353

For  $n\_clusters = 3$ , the silhouette score is 0.43028449435712024

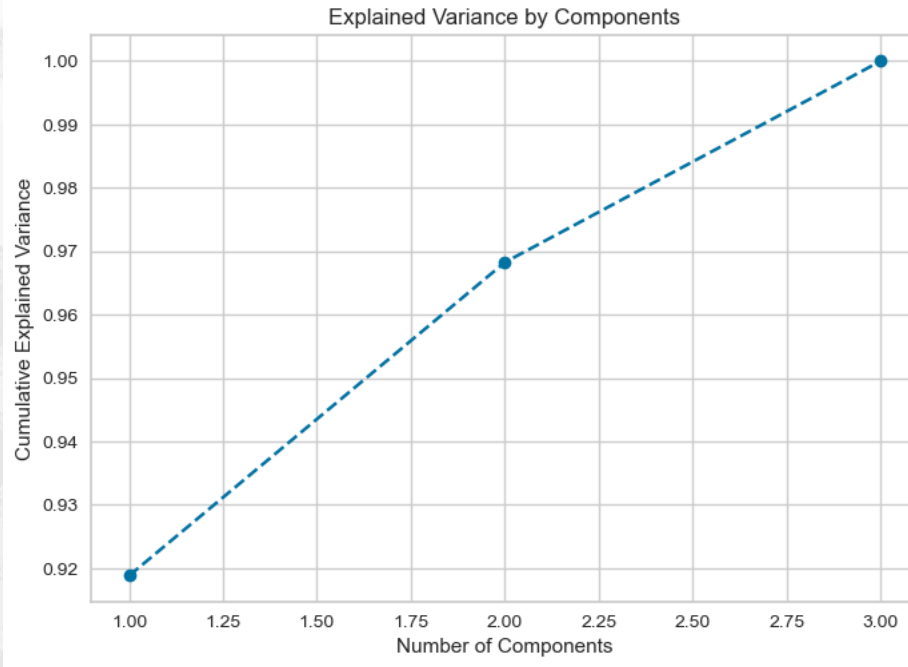
For  $n\_clusters = 4$ , the silhouette score is 0.36755972546792115

Berdasarkan keseimbangan tiap cluster dari silhouette plot, jumlah cluster yang optimal adalah 3 cluster. Meskipun rata-rata silhouette score untuk 2 cluster lebih tinggi, kita akan tetap menggunakan 3 cluster untuk pemodelan (dengan mempertimbangkan Elbow Method juga)



# Task 4 - Kmeans Clustering

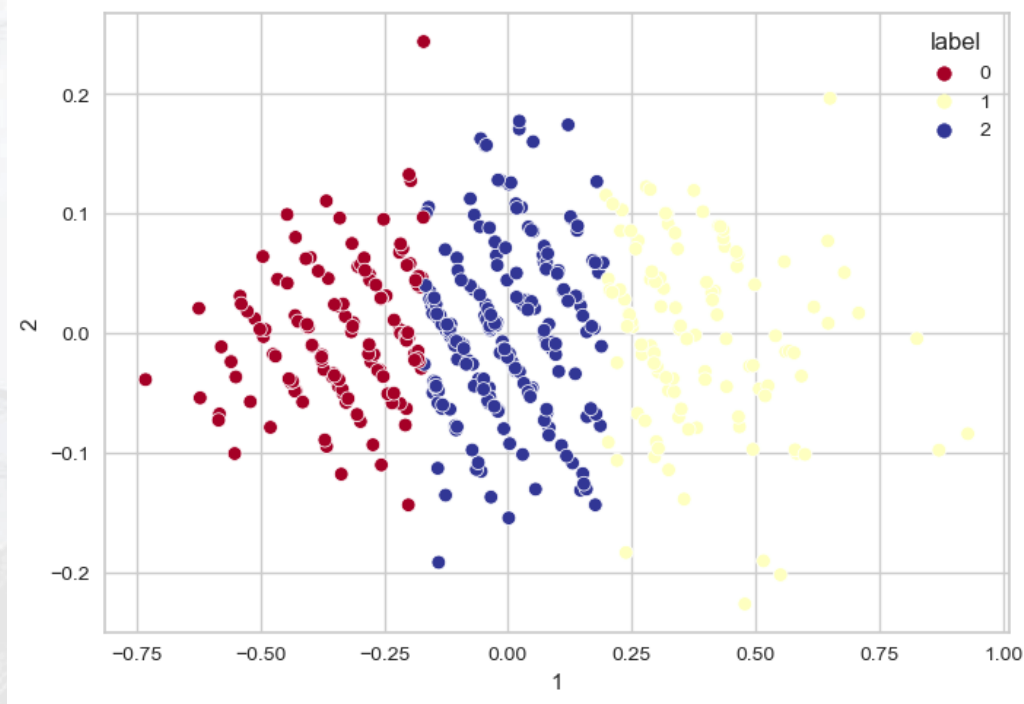
## PCA



n\_components yang dipilih adalah 2 components karena variasi data yang tercover sekitar 90%

# Task 4 - Kmeans Clustering

## Visualisasi Clustering



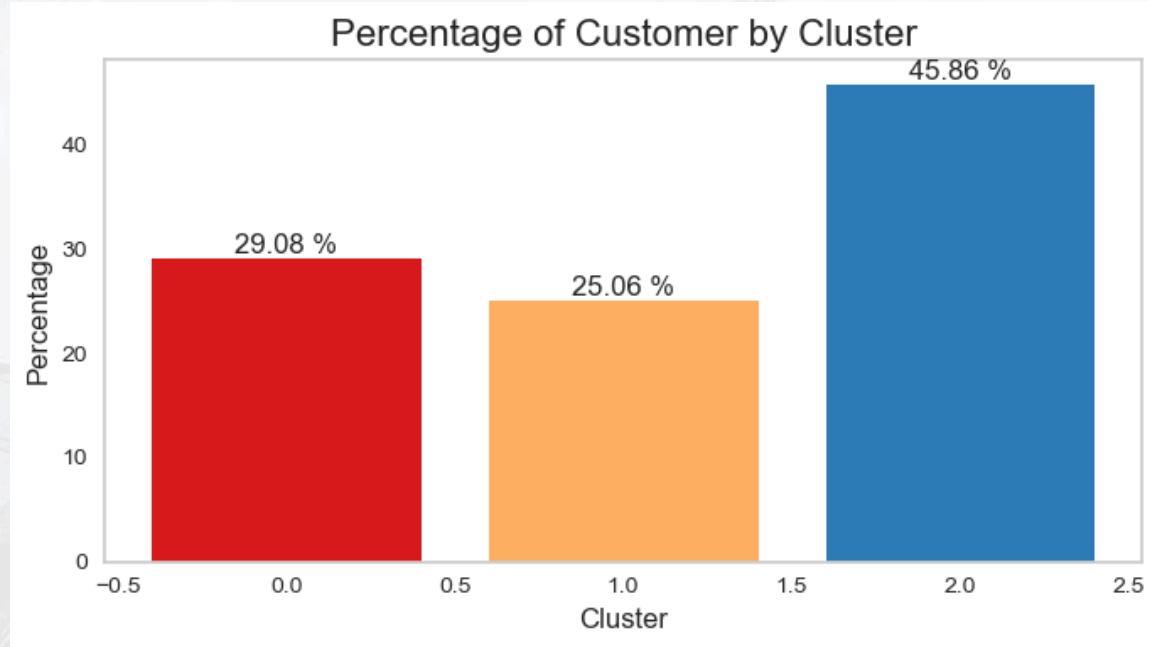
# Task 4 - Kmeans Clustering

## Customer Personality

cluster	TransactionID				Qty				TotalAmount			
	count	mean	median	std	count	mean	median	std	count	mean	median	std
0	130	7.692308	8.0	1.697450	130	26.646154	27.0	5.796098	130	228188.461538	233450.0	52952.510001
1	112	15.348214	15.0	1.769154	112	57.455357	57.0	7.124237	112	522525.892857	508200.0	81590.827900
2	205	11.224390	11.0	1.487988	205	40.960976	41.0	5.193174	205	360271.219512	360400.0	54821.139663

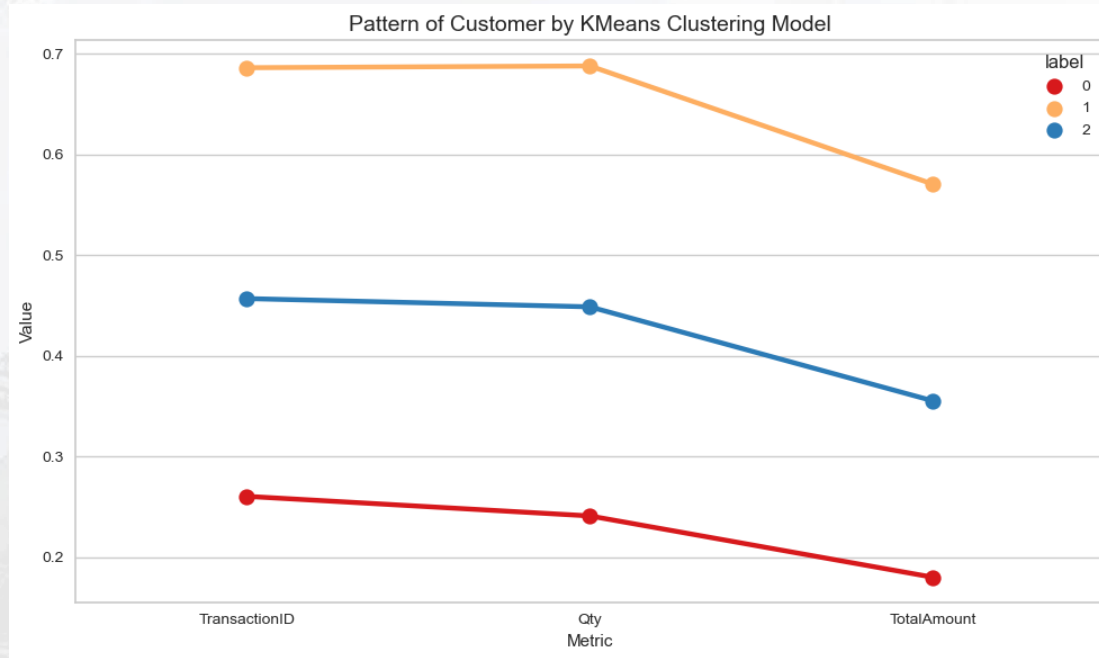
# Task 4 - Kmeans Clustering

## Customer Personality



# Task 4 - Kmeans Clustering

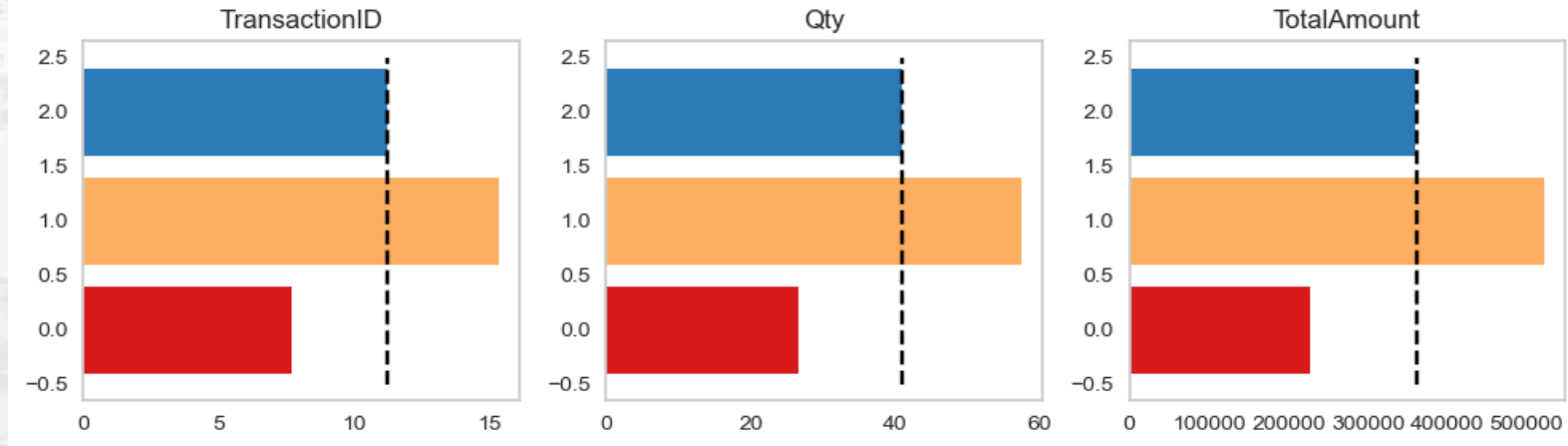
## Customer Personality





# Task 4 - Kmeans Clustering

## Customer Personality



Tabel - Akumulasi Hasil Nilai Fitur pada Setiap Cluster

Cluster	High Value	Average Value	Low Value
Cluster 0	TransactionID, Qty, TotalAmount		
Cluster 1	TransactionID, Qty, TotalAmount		
Cluster 2	TransactionID, Qty, TotalAmount		

# Task 4 - Kmeans Clustering

## Clustering Interpretation

- Cluster 0 - New Customer
  - ❑ Terdapat 130 customer (29.08%).
  - ❑ Customer di kelompok ini memiliki rata-rata transaksi yang rendah, yaitu sekitar 8 kali transaksi, transaksi, rata-rata jumlah atau kuantitas produk yang dibeli customer rendah yaitu sekitar 27 unit produk, dan rata-rata jumlah uang yang dikeluarkan oleh customer rendah yaitu sekitar 228K.
  
- Cluster 1 – Loyalty Customer
  - ❑ Terdapat 112 customer (25.06%)
  - ❑ Customer di kelompok ini memiliki rata-rata transaksi yang tinggi, yaitu sekitar 15 kali transaksi, rata-rata jumlah atau kuantitas produk yang dibeli customer tinggi yaitu sekitar 57 unit produk, dan rata-rata jumlah uang yang dikeluarkan oleh customer tinggi yaitu sekitar 522K.

# Task 4 - Kmeans Clustering

## Clustering Interpretation

- Cluster 2 - Potential Customer
- ❑ Terdapat 205 customer (45.86%)
- ❑ Customer di kelompok ini memiliki rata-rata transaksi yang sedang, yaitu sekitar 11 kali transaksi, rata-rata jumlah atau kuantitas produk yang dibeli customer sedang yaitu sekitar 41 unit produk, dan rata-rata jumlah uang yang dikeluarkan oleh customer sedang yaitu sekitar 360K.

## Business Recommendation

- ❖ Cluster 0 - New Customer: Untuk mendorong pembelian berulang dari pelanggan baru, bisnis dapat fokus pada program loyalitas, menawarkan kupon atau diskon untuk pembelian pertama, dan mengirimkan email atau notifikasi push untuk mempromosikan produk atau layanan baru.

# Task 4 - Kmeans Clustering

## Business Recommendation

- ❖ Cluster 1 - Loyalty Customer: Untuk mempertahankan pelanggan loyal dan meningkatkan nilai mereka, bisnis dapat menawarkan program loyalitas yang lebih menarik dan produk atau layanan eksklusif. Rekomendasi ini bertujuan untuk memberikan nilai tambah kepada pelanggan loyal, sehingga mereka lebih cenderung untuk tetap menjadi pelanggan Anda.
- ❖ Cluster 2 - Potential Customer: Pelanggan potensial adalah pelanggan yang telah membeli produk atau layanan Anda dalam beberapa waktu terakhir. Mereka adalah pelanggan yang memiliki potensi untuk menjadi pelanggan loyal, tetapi mereka belum mencapai tingkat pembelian yang sama seperti pelanggan loyal. Rekomendasi yang diberikan di atas bertujuan untuk meningkatkan jumlah produk yang dibeli pelanggan potensial per transaksi. Dengan demikian, Anda dapat mendorong mereka untuk menjadi pelanggan yang lebih loyal. Program diskon atau promosi dapat membantu Anda mencapai tujuan ini.



# Link Folder/Github/Video Presentation

Link Folder di Google Drive : [disini](#)

Link Github: [disini](#)

Link Video Presentation di Youtube: [disini](#)



# Thank You

