

FINAL PROJECT

Project-Based Intern : Data Scientist

Presented by
Maya Indah Nurrohmah



Hai!
Perkenalkan aku
**Maya Indah
Nurrohmah**

Deskripsi Diri

Saya merupakan seorang *fresh graduate* **Program Studi Matematika** di **Universitas Diponegoro**. Saya senang berbagi dan berkolaborasi dengan orang lain, memecahkan masalah, dan dapat menyampaikan ide dengan baik.

Pengalaman

Mentee

***Program Data, Business Analytics, and Operations
Di Ruangguru***
(Feb 2022 – Jul 2022)

Mempelajari bisnis *analytics*, bahasa pemrograman *Python* dan *SQL*, proses pengumpulan dan pemrosesan data, pembuatan visualisasi data, dan strategi bisnis yang baik bagi perusahaan.

Peserta

***Learn Basics & Data Analytics with Microsoft Excel for
Beginners di Pintar***
(Tahun 2021)

Mempelajari dasar-dasar penggunaan excel untuk melakukan pengolahan data mulai dari impor data, pembersihan data, analisis deskriptif, regresi, dan korelasi

Case Study

Case Study

(Business Understanding)

Misalkan Data Scientist di Kalbe Nutritionals mendapatkan **project baru dari tim inventory** dan **tim marketing** dengan permintaan dari setiap tim sebagai berikut:

A. Tim Inventory

Permintaan:

Membantu memprediksi jumlah penjualan dari total keseluruhan produk Kalbe

Tujuan:

Mengetahui perkiraan *quantity* produk yang terjual



Case Study

(Business Understanding)

Misalkan Data Scientist di Kalbe Nutritionals dan sedang mendapatkan **project baru** dari **tim inventory** dan **tim marketing** dengan permintaan dari setiap tim sebagai berikut:

B. Tim Marketing

Permintaan:

Membuat *customer segmentation*

Tujuan:

Memberikan *personalized promotion* dan *sales treatment* kepada *customer*



data_store

Column	Description	Column	Description
StoreID	Kode Unik Store	Type	Modern Trade, General Trade
StoreName	Nama Toko	Latitude	Kode Latitude
GroupStore	Nama group	Longitude	Kode Longitude

data_products

Column	Description
ProductID	Kode Unik Product
Product Name	Nama Product
Price	Harga dlm rupiah

data_transactions

Column	Description	Column	Description
TransactionID	Kode Unik Transaksi	Price	Harga Item (Rp)
CustomerID	No Unik Customer	Qty	Jumlah Item yang dibeli
Date	Tanggal transaksi	Total Amount	Price x Qty (Rp)
ProductID	Kode Unik Product	StoreID	Kode Unik Store

data_customer

Column	Description
CustomerID	No Unik Customer
Age	Usia Customer (Tahun)
Gender	Jenis Kelamin (0 Wanita, 1 Pria)
Marital Status	Married, Single (Blm menikah/Pernah menikah)
Income	Pendapatan per Bulan dalam Jutaan Rupiah

Data Pre-processing (Menyesuaikan Tipe Data dan Handle Missing Value)

data_store

```
# Melihat informasi data_store
```

```
data_store.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   StoreID      14 non-null    int64
1   StoreName    14 non-null    object
2   GroupStore   14 non-null    object
3   Type         14 non-null    object
4   Latitude     14 non-null    object
5   Longitude    14 non-null    object
dtypes: int64(1), object(5)
memory usage: 800.0+ bytes
```

```
# Mengubah tipe data kolom StoreID
```

```
data_store['StoreID'] =
data_store['StoreID'].astype("object")
data_store.info()
```

data_products

```
# Melihat informasi data_products
```

```
data_products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   ProductID    10 non-null    object
1   Product Name  10 non-null    object
2   Price        10 non-null    int64
dtypes: int64(1), object(2)
memory usage: 368.0+ bytes
```

data_transactions

```
# Melihat informasi data_transactions
```

```
data_transactions.info()
```

```
<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
```

```
# Mengubah Tipe Data
```

```
data_transactions['TransactionID'] =
data_transactions['TransactionID'].astype("object")
data_transactions['CustomerID'] =
data_transactions['CustomerID'].astype("object")
data_transactions['StoreID'] =
data_transactions['StoreID'].astype("object")
data_transactions.info()
```

Data Pre-processing (Menyesuaikan Tipe Data dan Handle Missing Value)

data_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	444 non-null	object
4	Income	447 non-null	object

```
dtypes: int64(3), object(2)  
memory usage: 17.6+ KB
```

```
# Mengubah tipe data 'CustomerID' dan 'Gender' menjadi object
```

```
data_customer['CustomerID'] = data_customer['CustomerID'].astype("object")  
data_customer['Gender'] = data_customer['Gender'].astype("object")  
data_customer.info()
```

```
# Menghapus nilai null
```

```
data_customer = data_customer.dropna()  
data_customer.count()
```

```
CustomerID      444  
Age             444  
Gender          444  
Marital Status  444  
Income          444  
dtype: int64
```


Data Pre-processing (Menyesuaikan Tipe Data dan Handle Missing Value)

data_customer

```
# Menampilkan umur customer termuda
```

```
data_customer[data_customer.Age == data_customer.Age.min()]
```

	CustomerID	Age	Gender	Marital Status	Income
127	128	0	1	Married	6,77

```
# Menampilkan umur customer yang berusia kurang dari 18 tahun
```

```
data_customer[data_customer.Age < 18]
```

	CustomerID	Age	Gender	Marital Status	Income
11	12	2	1	Married	4,94
73	74	3	1	Married	5,09
127	128	0	1	Married	6,77

```
# Mengecualikan customer berusia 2,3,dan 0 tahun
```

```
data_customer = data_customer[data_customer['Age'] >= 18]  
data_customer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 441 entries, 0 to 446
```

```
Data columns (total 5 columns):
```

#	Column	Non-Null Count	Dtype
0	CustomerID	441 non-null	object
1	Age	441 non-null	int64
2	Gender	441 non-null	object
3	Marital Status	441 non-null	object
4	Income	441 non-null	object

```
dtypes: int64(1), object(4)
```

```
memory usage: 20.7+ KB
```

Data Pre-processing (Memeriksa Data Duplikat)

data_store

```
# Memeriksa data duplikat dalam dataset
```

```
duplicate_rows_data_store = data_store[data_store.duplicated()]  
print('banyaknya baris yang terduplikat:', duplicate_rows_data_store.shape)
```

banyaknya baris yang terduplikat: (0, 6)

data_products

```
# Memeriksa data duplikat dalam dataset
```

```
duplicate_rows_data_products = data_products[data_products.duplicated()]  
print('banyaknya baris yang terduplikat:', duplicate_rows_data_products.shape)
```

banyaknya baris yang terduplikat: (0, 3)

data_transactions

```
# Memeriksa data duplikat dalam dataset
```

```
duplicate_rows_data_transactions = data_transactions[data_transactions.duplicated()]  
print('banyaknya baris yang terduplikat:', duplicate_rows_data_transactions.shape)
```

banyaknya baris yang terduplikat: (0, 8)

data_customer

```
# Memeriksa data duplikat dalam dataset
```

```
duplicate_rows_data_customer = data_customer[data_customer.duplicated()]  
print('banyaknya baris yang terduplikat:', duplicate_rows_data_customer.shape)
```

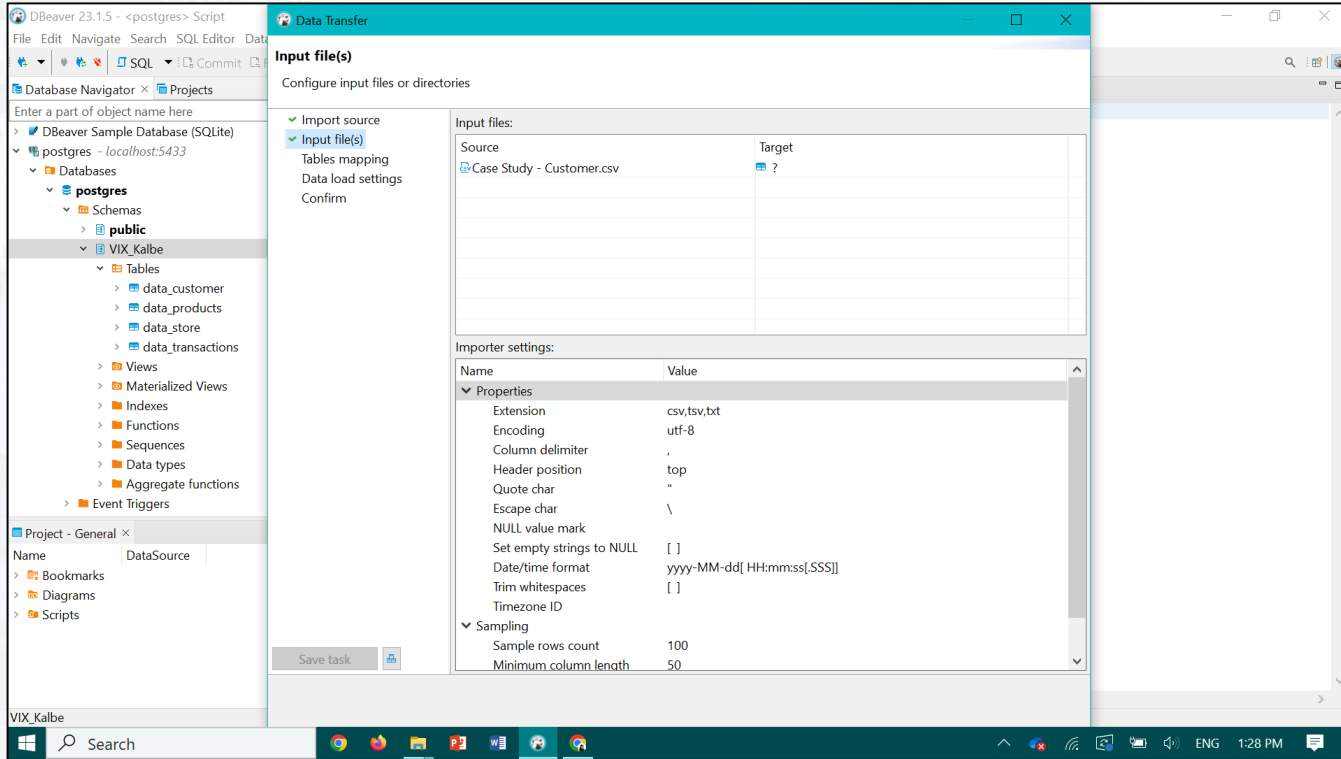
banyaknya baris yang terduplikat: (0, 5)

Dari masing-masing dataset, **tidak ada data yang terduplikat**

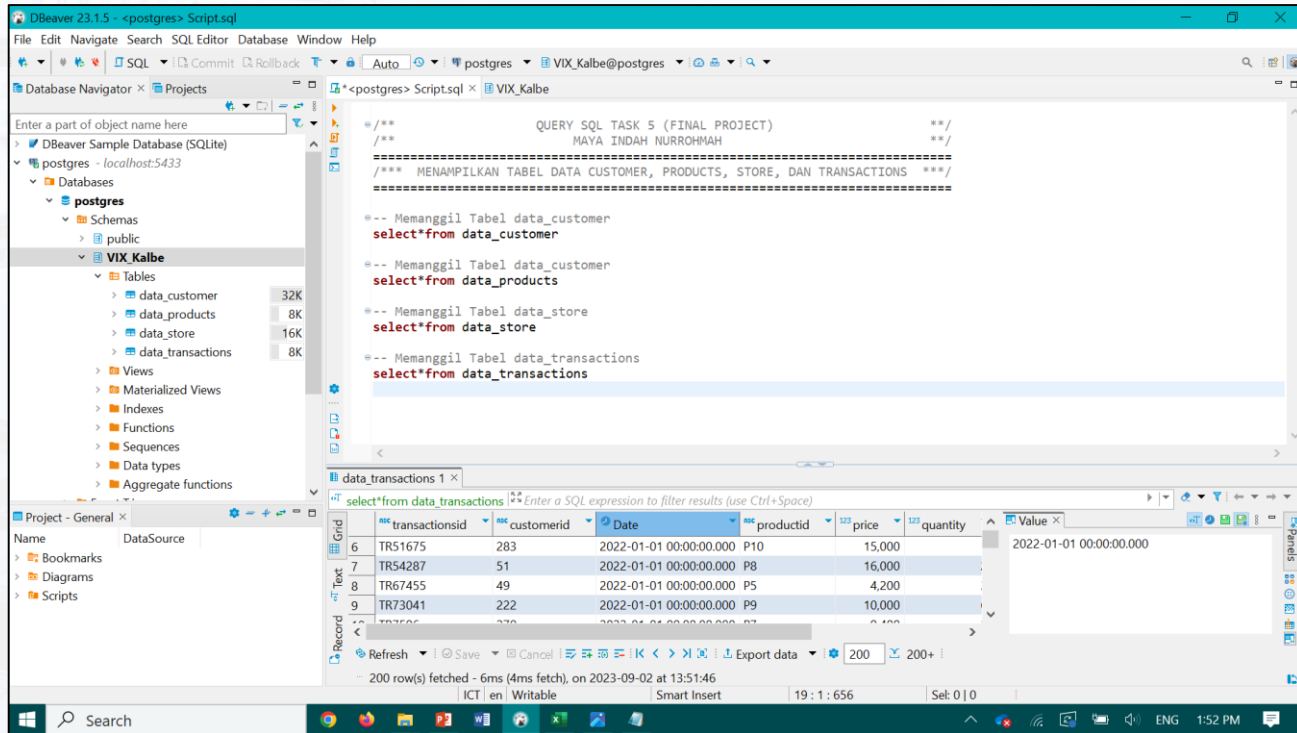
Challenge 1

Melakukan ***data ingestion*** ke dalam dbeaver

data ingestion adalah proses pemindahan data dari satu atau beberapa sumber
Ke suatu penyimpanan yang nantinya akan disimpan dan dianalisis lebih lanjut



Membuat schemas baru >> Import data dalam menu Tables >> Import Sources >> Configure Table and Configure Structure >>Next >> Proceed



The screenshot shows the DBeaver 23.1.5 interface. The left sidebar displays the database structure: PostgreSQL - localhost:5433 -> postgres -> public -> VIX_Kalbe. The main editor shows a SQL script with comments and queries for each table. The bottom pane displays the results of a query on the 'data_transactions' table.

```
/**      QUERY SQL TASK 5 (FINAL PROJECT)      **/
/**      MAYA INDAH NURROHMAH      **/
/**      MENAMPILKAN TABEL DATA CUSTOMER, PRODUCTS, STORE, DAN TRANSACTIONS      **/
=====

-- Mengambil Tabel data_customer
select*from data_customer

-- Mengambil Tabel data_customer
select*from data_products

-- Mengambil Tabel data_store
select*from data_store

-- Mengambil Tabel data_transactions
select*from data_transactions
```

	transactionsid	customerid	Date	productid	price	quantity
6	TR51675	283	2022-01-01 00:00:00.000	P10	15,000	
7	TR54287	51	2022-01-01 00:00:00.000	P8	16,000	
8	TR67455	49	2022-01-01 00:00:00.000	P5	4,200	
9	TR73041	222	2022-01-01 00:00:00.000	P9	10,000	

200 row(s) fetched - 6ms (4ms fetch), on 2023-09-02 at 13:51:46

Setelah memasukkan dan mengatur delimiter dari 4 dataset ke dalam dbeaver, tabel-table tersebut telah berhasil di import dalam dbeaver

Challenge 2

Melakukan **exploratory data analysis** melalui dbeaver

Exploratory Data Analysis

Rata-Rata Umur *Customer* (berdasarkan *Marital Status*)

Query:

```
select
  maritalstatus,
  avg(age) as average_age
from
  data_customer
group by
  maritalstatus;
```

Output:

ABC maritalstatus ▼	123 average_age ▼
Single	29.3846153846
Married	43.4065281899

Jadi, diperoleh informasi bahwa:

- **Rata-rata usia *customer*** yang masih **lajang** (*single*) yaitu **29,4 tahun**
- **Rata-rata usia *customer*** yang **sudah menikah** (*Married*) yaitu **43,4 tahun**

Exploratory Data Analysis

Rata-Rata Umur *Customer* (berdasarkan *Gender*)

Query:

```
select
  gender,
  avg(age) as average_age
from
  data_customer
group by
  gender;
```

Output:

	ABC gender ▾	123 average_age ▾
1	1	39.824120603
2	0	40.326446281

Jadi, diperoleh informasi bahwa:

- **Rata-rata usia *customer* laki-laki** (kode 1) yaitu **39,8 tahun**
- **Rata-rata usia *customer* perempuan** (kode 0) yaitu **40,3 tahun**

Exploratory Data Analysis

```
/* Melakukan left join data_transactions dengan data_store dan data_products */
```

```
select*from data_transactions;  
select*from data_store;  
select*from data_products;
```

```
select
```

```
  a.productid,  
  c.productname,  
  a.quantity,  
  a.totalamount,  
  a.storeid,  
  b.storenama
```

```
from
```

```
  data_transactions as a  
left join  data_store as b  
  on a.storeid = b.storeid  
left join  data_products as c  
  on a.productid = c.productid;
```

	productid	productname	quantity	totalamount	storeid	storenama
1	P3	Crackers	4	30,000	12	Prestasi Utama
2	P9	Yoghurt	7	70,000	1	Prima Tendean
3	P1	Choco Bar	4	35,200	4	Gita Ginara
4	P1	Choco Bar	7	61,600	4	Gita Ginara
5	P9	Yoghurt	1	10,000	4	Gita Ginara
6	P10	Cheese Stick	1	15,000	5	Bonafid
7	P8	Oat	2	32,000	2	Prima Kelapa Dua
8	P5	Thai Tea	3	12,600	13	Buana
9	P9	Yoghurt	6	60,000	4	Gita Ginara
10	P7	Coffee Candy	2	18,800	14	Priangan
11	P4	Potato Chip	4	48,000	12	Prestasi Utama

Exploratory Data Analysis

Nama **Store** dengan **Total Quantity** Terbanyak

Query:

```
select
  storenama,
  sum(quantity) as totalquantity
from
  data_question3_question4
group by
  storenama
order by
  totalquantity desc;
```



Output:

	storenama	totalquantity
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

Jadi, terdapat 12 *store* yang ada dalam data dan **store yang berhasil menjual item (total quantity) terbanyak** yaitu **Lingga** dengan total item yang berhasil terjual (*total quantity*) sebanyak 2777 buah.

Exploratory Data Analysis

Nama **Produk Terlaris** dengan *Total Amount* Terbanyak

Query:

```
select
  productname,
  sum(totalamount) as totalamountfinal
from
  data_question3_question4
group by
  productname
order by
  totalamountfinal desc;
```



Output:

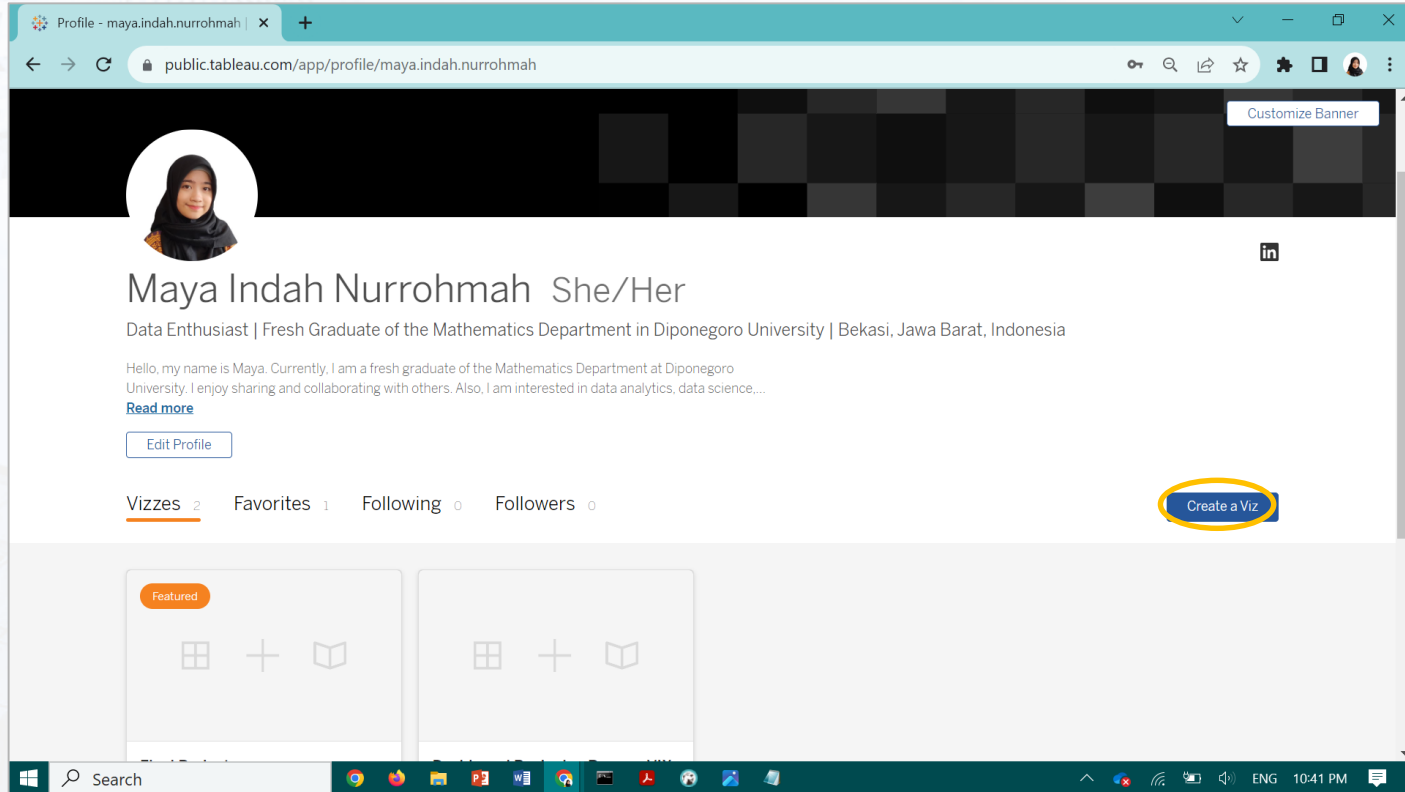
	productname	totalamountfinal
1	Cheese Stick	27,615,000
2	Choco Bar	21,190,400
3	Coffee Candy	19,711,800
4	Yoghurt	19,630,000
5	Oat	15,440,000
6	Crackers	13,680,000
7	Potato Chip	13,104,000
8	Thai Tea	11,982,600
9	Cashew	11,286,000
10	Ginger Candy	8,403,200

Jadi, diperoleh informasi bahwa terdapat 10 jenis produk yang terjual dan **produk dengan total amount terbanyak** yaitu **cheese stick** (Rp 27.615.000)

Challenge 3

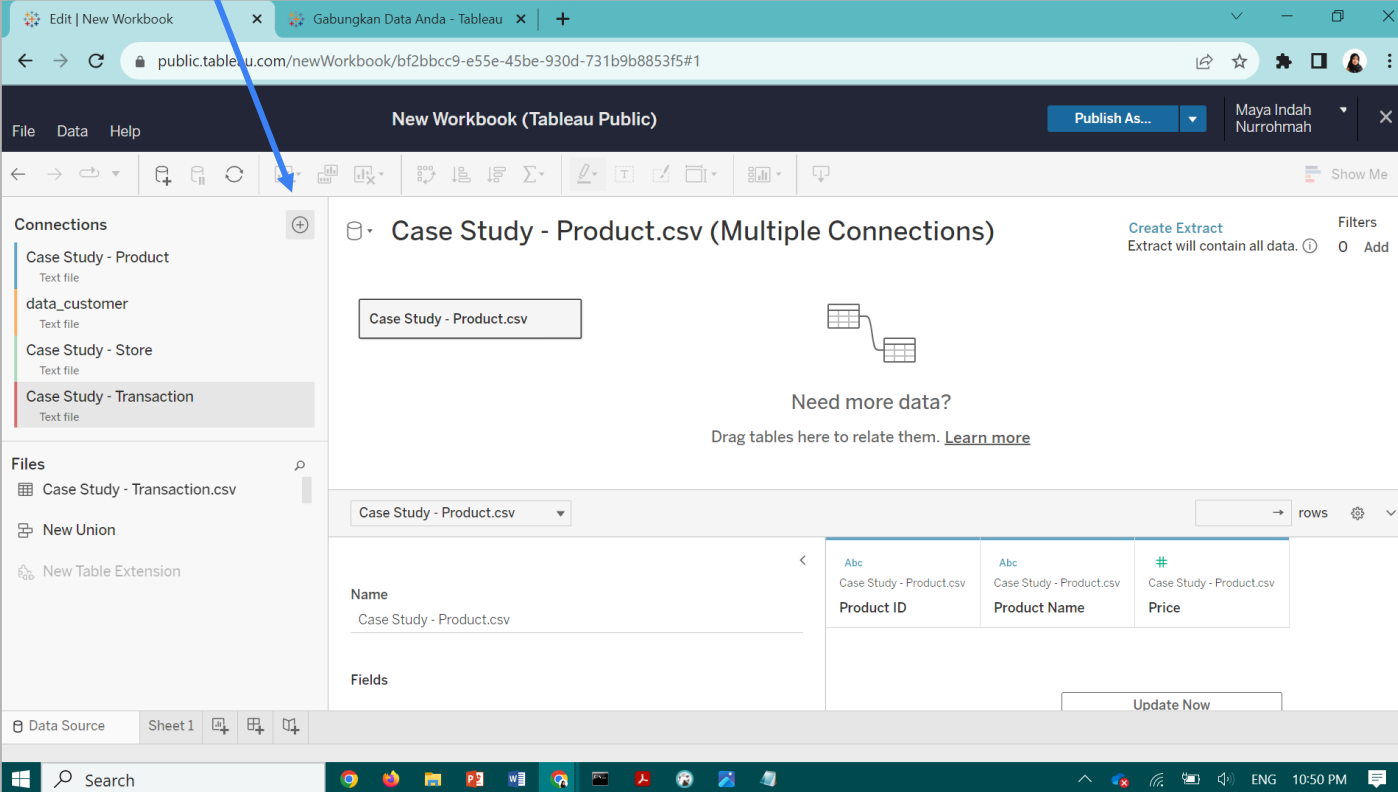
Melakukan **data ingestion** ke dalam **Tableau Public**

Melakukan **data ingestion** ke dalam **Tableau Public**



The screenshot shows a web browser window displaying the profile of Maya Indah Nurrohmah on Tableau Public. The browser's address bar shows the URL `public.tableau.com/app/profile/maya.indah.nurrohmah`. The profile header includes a circular profile picture of a woman wearing a hijab, a black and white checkered banner, and a "Customize Banner" button. Below the banner, the name "Maya Indah Nurrohmah" is displayed with the pronouns "She/Her" and a LinkedIn icon. The bio identifies her as a "Data Enthusiast | Fresh Graduate of the Mathematics Department in Diponegoro University | Bekasi, Jawa Barat, Indonesia". A short bio paragraph follows, mentioning her university and interests in data analytics and science, with a "Read more" link. An "Edit Profile" button is located below the bio. The navigation tabs at the bottom of the profile section are "Vizzes" (with a count of 2), "Favorites" (1), "Following" (0), and "Followers" (0). A blue "Create a Viz" button is highlighted with a yellow circle. The main content area shows two placeholder cards for visualizations, each with a "Featured" label and icons for a grid, a plus sign, and a document. The Windows taskbar at the bottom shows the search bar and various application icons, with the system clock indicating 10:41 PM on ENG.

Melakukan **data ingestion** ke dalam **Tableau Public**



The screenshot shows the Tableau Public web interface. A blue arrow points to the 'Get Data' button in the top toolbar. The 'Connections' pane on the left lists several text files: 'Case Study - Product', 'data_customer', 'Case Study - Store', and 'Case Study - Transaction'. The main workspace displays 'Case Study - Product.csv (Multiple Connections)'. Below this, a message asks 'Need more data?' with a link to 'Learn more'. At the bottom, a table preview shows columns: 'Name', 'Product ID', 'Product Name', and 'Price'. The 'Name' column is expanded, showing 'Case Study - Product.csv'. The 'Product ID' column shows 'Case Study - Product.csv'. The 'Product Name' column shows 'Case Study - Product.csv'. The 'Price' column shows 'Case Study - Product.csv'. The 'Update Now' button is visible at the bottom right of the table preview.

Connections

- Case Study - Product
Text file
- data_customer
Text file
- Case Study - Store
Text file
- Case Study - Transaction
Text file

Files

- Case Study - Transaction.csv
- New Union
- New Table Extension

Case Study - Product.csv (Multiple Connections)

Create Extract
Extract will contain all data. ⓘ 0 Add

Case Study - Product.csv

Need more data?
Drag tables here to relate them. [Learn more](#)

Case Study - Product.csv

Name	Product ID	Product Name	Price
Case Study - Product.csv	Case Study - Product.csv	Case Study - Product.csv	Case Study - Product.csv

Update Now

The screenshot displays the Tableau Public interface for a workbook titled "TASK 5 (Tableau Public)". The user is logged in as Maya Indah Nurrohman. The interface shows a "Connections" pane on the left with four text file connections: "Case Study - Product", "data_customer", "Case Study - Store", and "Case Study - Transaction". The main workspace shows a "Case Study - Transaction.csv+ (Multiple Connections)" view, which is a Union of these four files. The Union is visualized as a central box "Case Study - Transaction.c..." with lines connecting to four other boxes: "Case Study - Product.csv", "Case Study - Store.csv", "data_customer.csv", and "Case Study - Transaction.csv". Below the Union, a dropdown menu shows "Case Stud..." and "Case Stud...". The bottom of the interface shows a "Data Source" section with a table of data. The table has three columns: "ProductID (Case Study - ...)", "Product Name", and "Price (Case Study - Prod...)". The data rows show "Abc" for ProductID and "Case Study - Product.csv" for Product Name. The Price column shows a hash symbol "#". An "Update Now" button is visible at the bottom right.

The screenshot displays the Tableau Public interface for a workbook titled "TASK 5 (Tableau Public)". The user is logged in as Maya Indah Nurrohman. The interface shows a "Connections" pane on the left with four text file connections: "Case Study - Product", "data_customer", "Case Study - Store", and "Case Study - Transaction". The main workspace shows a "Case Study - Transaction.csv+ (Multiple Connections)" view, which is a Union of these four files. The Union is visualized as a central box "Case Study - Transaction.c..." with lines connecting to four other boxes: "Case Study - Product.csv", "Case Study - Store.csv", "data_customer.csv", and "Case Study - Transaction.csv". Below the Union, a dropdown menu shows "Case Stud..." and "Case Stud...". The bottom of the interface shows a "Data Source" section with a table of data. The table has three columns: "ProductID (Case Study - ...)", "Product Name", and "Price (Case Study - Prod...)". The data rows show "Abc" for ProductID and "Case Study - Product.csv" for Product Name. The Price column shows a hash symbol "#". An "Update Now" button is visible at the bottom right.

Challenge 4

Membuat **dashboard di Tableau Public**

Total Amount

Rp 162,043,000.00

Total Quantity

18,296.00 item

DASHBOARD PENJUALAN TAHUN 2022

Pilih Nama Produk:

All

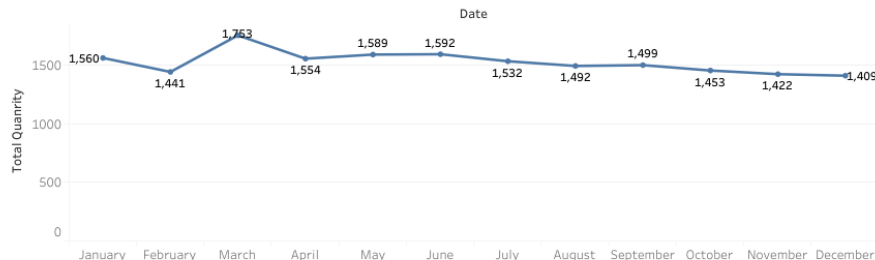
Pilih Nama Store:

All

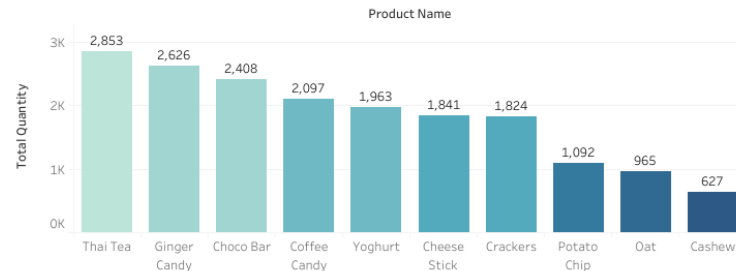
Pilih Bulan:

All

JUMLAH QUANTITY DARI BULAN KE BULAN



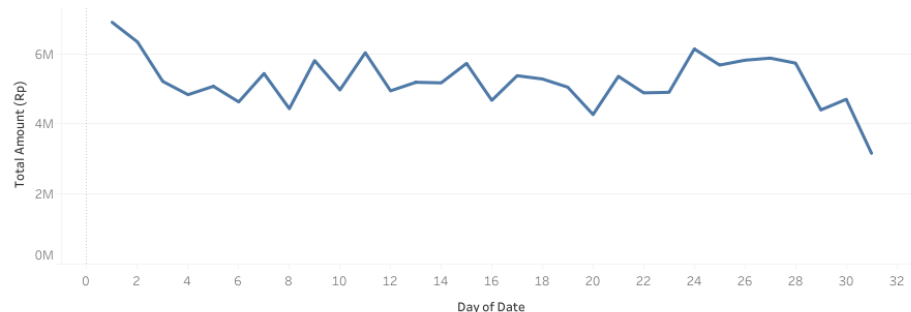
TOTAL QUANTITY BERDASARKAN PRODUK



TOTAL AMOUNT BERDASARKAN NAMA STORE

Lingga Rp 25,294,100	Prestasi Utama Rp 12,285,200	Bonafid Rp 11,595,600	Harapan Baru Rp 11,329,500	Gita Ginara Rp 11,116,100
	Prima Kelapa Dua Rp 12,136,300	Prima Kota Rp 11,551,100	Priangan Rp 10,995,100	
Sinar Harapan Rp 21,882,600	Prima Tendeau Rp 11,895,500	Buana Rp 11,332,000		
			Buana Indah Rp 10,629,900	

JUMLAH TOTAL AMOUNT DARI HARI KE HARI



Challenge 5

Membuat **model prediktif timeseries** dan **clustering**

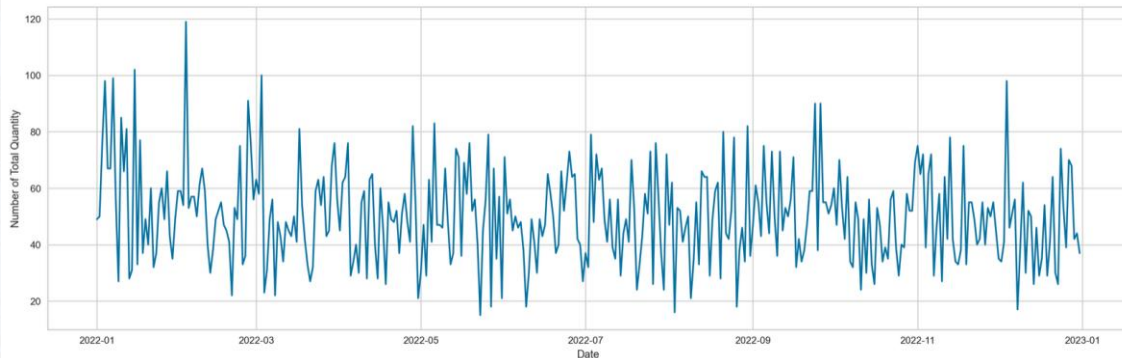
Model Prediktif *Time Series*

```
# Memanggil dataset
```

```
data_transactions_final_2 = pd.read_csv('data_transactions_final.csv', index_col=[0])  
data_transactions_final_2.head()
```

```
data_untuk_timeseries = data_transactions_final_2.groupby('Date').agg({'Qty': 'sum'})  
data_untuk_timeseries
```

[<matplotlib.lines.Line2D at 0x1a6ea9250cd>]



	Qty
Date	
2022-01-01	49
2022-01-02	50
2022-01-03	76
2022-01-04	98
2022-01-05	67
...	...

01

Import dan Grouping
Dataset

Model Prediktif *Time Series*

```
from statsmodels.tsa.stattools import adfuller
def adf_test(dataset):
    dfctest = adfuller(dataset, autolag = 'AIC')
    print("1. ADF : ",dfctest[0])
    print("2. P-Value : ", dfctest[1])
    print("3. Num Of Lags : ", dfctest[2])
    print("4. Num Of Observations Used For ADF Regression:",      dfctest[3])
    print("5. Critical Values :")
    for key, val in dfctest[4].items():
        print("\t",key, ": ", val)
adf_test(data_untuk_timeseries['Qty'])
```

02

Memeriksa Stationarity

```
1. ADF : -19.01878280229973
2. P-Value : 0.0
3. Num Of Lags : 0
4. Num Of Observations Used For ADF Regression: 364
5. Critical Values :
    1% : -3.4484434475193777
    5% : -2.869513170510808
    10% : -2.571017574266393
```

```
from pmdarima import auto_arima  
stepwise_fit = auto_arima(data_untuk_timeseries['Qty'], trace=True,  
suppress_warnings=True)
```

Performing stepwise search to minimize aic

```
ARIMA(2,0,2)(0,0,0)[0] intercept : AIC=3098.997, Time=0.51 sec  
ARIMA(0,0,0)(0,0,0)[0] intercept : AIC=3094.267, Time=0.01 sec  
ARIMA(1,0,0)(0,0,0)[0] intercept : AIC=3096.267, Time=0.05 sec  
ARIMA(0,0,1)(0,0,0)[0] intercept : AIC=3096.267, Time=0.12 sec  
ARIMA(0,0,0)(0,0,0)[0]          : AIC=3933.778, Time=0.02 sec  
ARIMA(1,0,1)(0,0,0)[0] intercept : AIC=3098.260, Time=0.09 sec
```

Best model: ARIMA(0,0,0)(0,0,0)[0] intercept

Total fit time: 0.797 seconds

Model Prediktif *Time Series*

```
print(data_untuk_timeseries.shape)
train=data_untuk_timeseries.iloc[:-30]
test=data_untuk_timeseries.iloc[-30:]
print(train.shape,test.shape)
```

```
(365, 1)
(335, 1) (30, 1)
```

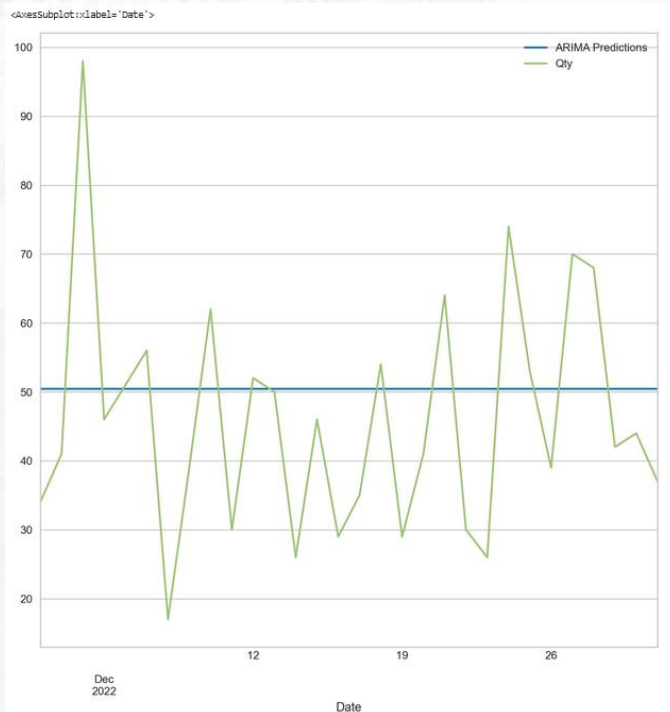
04

Split dan Train Data

```
# Train test split
```

```
from statsmodels.tsa.arima_model import ARIMA
model=sm.tsa.arima.ARIMA(train['Qty'],order=(0,0,0))
model=model.fit()
```

Model Prediktif *Time Series*



```
start=len(train)
end=len(train)+len(test)-1
pred=model.predict(start=start,end=end,typ='levels').rename('ARIMA Predictions')
pred.plot(legend=True)
test['Qty'].plot(legend=True)
```

```
from sklearn.metrics import mean_squared_error
from math import sqrt
test['Qty'].mean()
rmse=sqrt(mean_squared_error(pred,test['Qty']))
print(rmse)
```

17.549888906746727

Model Prediktif *Clustering*

01

Import Dataset



Melakukan **import dataset**
hasil **left join** dan **memilih**
fitur untuk dilakukan
clustering

```
# Memanggil dataset
```

```
data_transactions_final = pd.read_csv('data_transactions_final.csv', index_col=[0])  
data_transactions_final.head()
```

```
data_untuk_clustering = data_transactions_final[['CustomerID', 'TransactionID',  
                                                'Qty', 'TotalAmount']]
```

```
data_untuk_clustering
```



	CustomerID	TransactionID	Qty	TotalAmount
0	328	TR11369	4	30000
1	165	TR16356	7	70000
2	183	TR1984	4	35200
3	160	TR35256	7	61600
4	386	TR41231	1	10000
...

Menyesuaikan tipe data
dalam dataset

⋮

Informasi
dataset

02

```
data_untuk_clustering.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 5020 entries, 0 to 5019  
Data columns (total 4 columns):  
#   Column          Non-Null Count  Dtype  
---  ---  
0   CustomerID      5020 non-null   object  
1   TransactionID    5020 non-null   object  
2   Qty              5020 non-null   int64  
3   TotalAmount     5020 non-null   int64  
dtypes: int64(2), object(2)  
memory usage: 196.1+ KB
```

Model Prediktif **Clustering**

03

Grouping

...

Groupby berdasarkan **CustomerID** dan beberapa **aggregation**

```
data_untuk_clustering = data_untuk_clustering.groupby('CustomerID').agg({'TransactionID': 'count',  
                                                                           'Qty': 'sum',  
                                                                           'TotalAmount': 'sum'})  
data_untuk_clustering = data_untuk_clustering.rename(columns={'TransactionID': 'JumlahTransaksi'})  
data_untuk_clustering.head()
```

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



Model Prediktif **Clustering**

Melakukan **handling outliers** dengan metode **IQR**



Mengatasi
Outliers

04

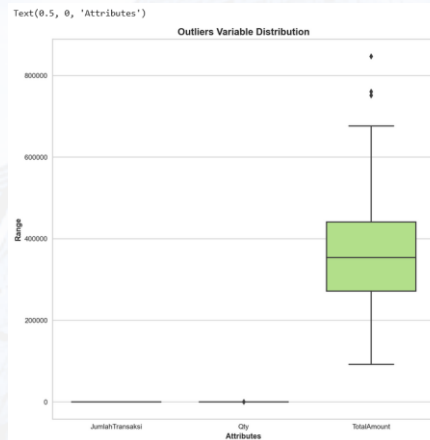
```
# Removing (statistical) outliers for TotalAmount
```

```
Q1 = data_untuk_clustering.TotalAmount.quantile(0.25)
```

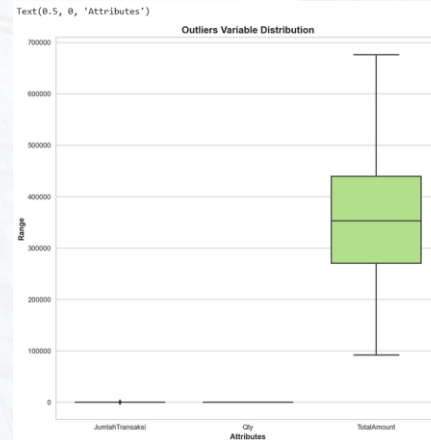
```
Q3 = data_untuk_clustering.TotalAmount.quantile(0.75)
```

```
IQR = Q3 - Q1
```

```
data_untuk_clustering = data_untuk_clustering[(data_untuk_clustering.TotalAmount >= Q1 - 1.5*IQR) &  
                                                (data_untuk_clustering.TotalAmount <= Q3 + 1.5*IQR)]
```



Sebelum **handling outliers**
dengan metode **IQR**



Sesudah **handling outliers**
dengan metode **IQR**

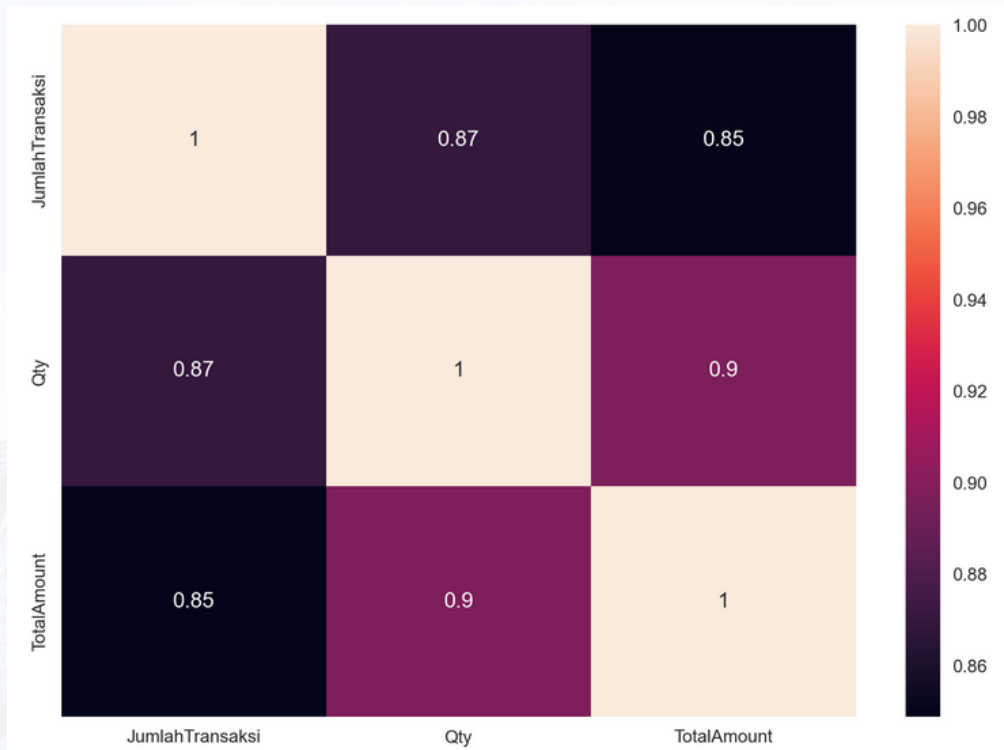
Model Prediktif **Clustering**

05

Korelasi antar
Variabel

⋮

Memeriksa
keterhubungan antar
variabel dalam dataset



Model Prediktif **Clustering**

Melakukan standarisasi
dengan **MinMaxScaler**

⋮

Standarisasi

06

```
# Mengubah Variabel Data Frame Menjadi Array
```

```
data_untuk_clustering_array = np.array(data_untuk_clustering)
print(data_untuk_clustering_array)
```

```
[[ 17    60 623300]
 [ 13    57 392300]
 [ 15    56 446200]
 ...
 [ 18    68 587200]
 [ 11    42 423300]
 [ 13    42 439300]]
```

```
# Menstandarkan Ukuran Variabel
```

```
scaler = MinMaxScaler()
data_untuk_clustering_scaled = scaler.fit_transform(data_untuk_clustering_array)
data_untuk_clustering_scaled
```

```
array([[0.77777778, 0.79365079, 0.90943332],
       [0.55555556, 0.74603175, 0.51395309],
       [0.66666667, 0.73015873, 0.60623181],
       ...,
       [0.83333333, 0.92063492, 0.84762883],
       [0.44444444, 0.50793651, 0.56702619],
       [0.55555556, 0.50793651, 0.59441876]])
```

Model Prediktif **Clustering**

07

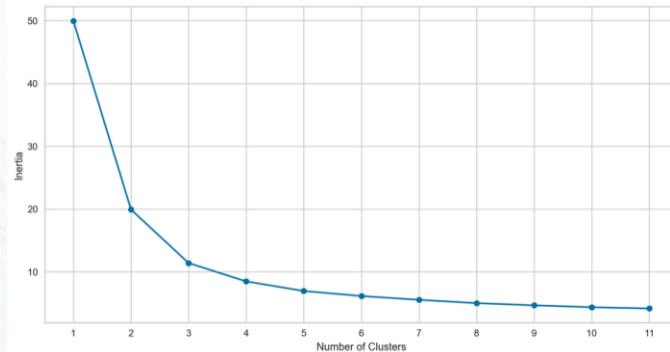
Menentukan
banyak cluster
(n)



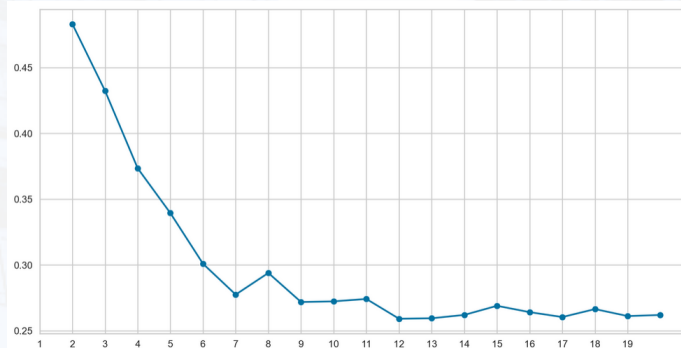
Menentukan n cluster
dengan **Elbow Method** dan
melakukan **evaluasi**
dengan **Silhouette Score**

Jadi untuk data yang diberikan, disimpulkan bahwa **jumlah cluster** optimal untuk data tersebut adalah **3 (n=3)** dengan **silhouette score** yaitu **0.43**.

Elbow Method:



Silhouette Score:



Model Prediktif **Clustering**

Mendeskripsikan
karakteristik tiap cluster
yang terbentuk



Karakteristik Tiap
Cluster (n=3)

08

Cluster 0

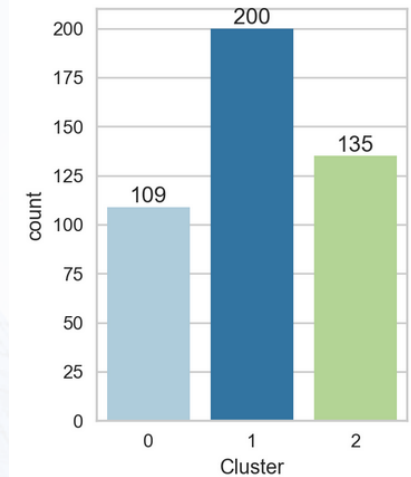
Cluster 0 merupakan kelompok yang memiliki customer paling sedikit yaitu 109 orang

Cluster 1

Cluster 1 merupakan kelompok yang memiliki customer paling banyak yaitu 200 orang

Cluster 2

Cluster 2 merupakan kelompok yang memiliki customer sebanyak 135 orang



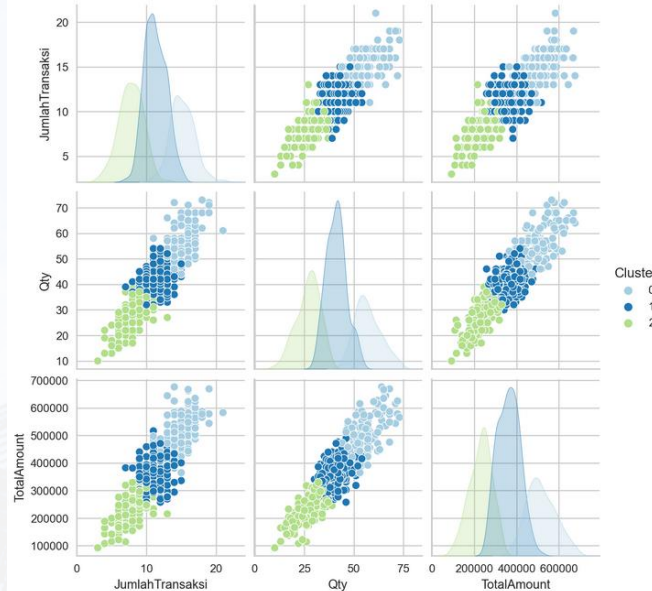
Model Prediktif **Clustering**

Mendeskripsikan
karakteristik tiap cluster
yang terbentuk

Karakteristik
Tiap Cluster

08

<Figure size 2000x1000 with 0 Axes>



Cluster 0

- Kelompok customer yang paling banyak melakukan transaksi
- Membeli item dengan jumlah banyak
- Menghasilkan total amount paling tinggi
- Kelompok ini akan mendatangkan keuntungan ke perusahaan
- Memberikan diskon dan penawaran lain yang ditargetkan untuk kelompok cluster 0 akan meningkatkan transaksi dan pembelian sehingga memaksimalkan keuntungan perusahaan.

Cluster 1

- Kelompok customer dengan jumlah transaksi rata-rata
- Membeli item dengan jumlah rata-rata
- Menghasilkan total amount sedang

Cluster 2

- Kelompok customer dengan jumlah transaksi sedikit
- Membeli item dengan jumlah sedikit
- Menghasilkan total amount rendah

Video Presentasi Dan File-File Pengerjaan



<http://bit.ly/DriveFPKalbeMaya>

<https://bit.ly/VideoPresentasiFPKalbeMaya>

Thank You



X



KALBE
Nutritional