VIRTUL INTERNSHIP KALBE

PRESENTATION

DATA SCIENCE

IZRA NOOR ZAHARA ALIYA

ABOUT PROJECT

Terdapat 4 data csv yaitu data customer, store, product, dan transaction

Pertama akan dilakukan cleaning data selanjutnya akan dilakukan pemodelan memprediksi total quantity harian dari product yang terjual
dan cluster customer-customer yang mirip

DATA PREPRATION

Import data CSV

```
import numpy as np
import pandas as pd


# for operation 2
dfc = pd.read_csv('/content/Case Study - Customer.csv',sep=';')
dfp = pd.read_csv('/content/Case Study - Product.csv',sep=';')
dfs = pd.read_csv('/content/Case Study - Store.csv',sep=';')
dft = pd.read_csv('/content/Case Study - Transaction.csv',sep=';')
```

DATA CLEANING

Mengubah tipe data date

```
# Mengubah tipe data kolom 'date' menjadi datetime
dft['Date'] = pd.to_datetime(dft['Date'])

# Menampilkan DataFrame setelah mengubah tipe data kolom 'date'
print(dft)
```

```
     TransactionID  CustomerID       Date ProductID  Price  Qty  TotalAmount  \
0          TR11369         328 2022-01-01        P3   7500    4        30000
1          TR16356         165 2022-01-01        P9  10000    7        70000
2           TR1984         183 2022-01-01        P1   8800    4        35200
3          TR35256         160 2022-01-01        P1   8800    7        61600
4          TR41231         386 2022-01-01        P9  10000    1        10000
...            ...         ...        ...       ...    ...  ...          ...
5015       TR54423         243 2022-12-31       P10  15000    5        75000
5016        TR5604         271 2022-12-31        P2   3200    4        12800
5017       TR81224          52 2022-12-31        P7   9400    6        56400
5018       TR85016          18 2022-12-31        P8  16000    3        48000
5019       TR85684          55 2022-12-31        P8  16000    1        16000

      StoreID
0          12
1           1
2           4
3           4
4           4
...       ...
5015        3
5016        9
5017        9
5018       13
5019        6
```

DATA CLEANING

Melihat kolom yang memiliki missing value pada tabel customer

```
# Melihat kolom-kolom yang memiliki missing value (nilai null)
kolom_dengan_missing_value = dfc.columns[dfc.isna().any()].tolist()

# Menampilkan nama kolom yang memiliki missing value
print("Kolom dengan missing value:")
print(kolom_dengan_missing_value)
```

```
Kolom dengan missing value:
['Marital Status']
```

Cek data null

```
null_count_dfc = dfc.isnull().sum()
print(null_count_dfc)

null_count_dfs = dfs.isnull().sum()
print(null_count_dfs)

null_count_dfp = dfp.isnull().sum()
print(null_count_dfp)

null_count_dft = dft.isnull().sum()
print(null_count_dft)
```

Melihat kolom yang memiliki missing value pada tabel product

```
[20] # Melihat kolom-kolom yang memiliki missing value (nilai null)
     kolom_dengan_missing_value = dfp.columns[dfp.isna().any()].tolist()

     # Menampilkan nama kolom yang memiliki missing value
     print("Kolom dengan missing value:")
     print(kolom_dengan_missing_value)
```

```
Kolom dengan missing value:
['ProductID', 'Product Name']
```

DATA CLEANING

Mengisi nilai missing value pada customer

```
mode_marital_status = dfc['Marital Status'].mode().iloc[0]

# Mengganti nilai-nilai yang hilang di kolom 'Marital Status' dengan mode
dfc['Marital Status'].fillna(mode_marital_status, inplace=True)

# Menampilkan DataFrame setelah mengganti nilai-nilai yang hilang
print(dfc)
```

```
     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
..          ...  ...     ...            ...    ...
442         443   33       1        Married   9,28
443         444   53       0        Married  15,31
444         445   51       0        Married  14,48
445         446   57       0        Married   7,81
446         447   54       1        Married  20,37

[447 rows x 5 columns]
```

Mengisi nilai missing value pada customer product

```
[20] # Melihat kolom-kolom yang memiliki missing value (nilai null)
     kolom_dengan_missing_value = dfp.columns[dfp.isna().any()].tolist()

     # Menampilkan nama kolom yang memiliki missing value
     print("Kolom dengan missing value:")
     print(kolom_dengan_missing_value)
```

```
Kolom dengan missing value:
['ProductID', 'Product Name']
```

DATA PROCESSING

DATA PREDICTION WITH MECHINE LEARNING
memprediksi total quantity harian
dari product yang terjual

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Menggabungkan DataFrame dfp dan dft berdasarkan 'ProductID'
merged_data = dfp.merge(dft, on='ProductID')

# Mengagregasi data dengan mengelompokkan berdasarkan tanggal dan menjumlahkan kuantitas produk
daily_qty = merged_data.groupby('Date')['Qty'].sum()

# Membuat time series data yang memuat total kuantitas harian
ts_data = daily_qty.resample('D').sum()

# Mengisi nilai-nilai yang hilang dengan 0 jika diperlukan
ts_data = ts_data.fillna(0)

# Memisahkan data menjadi data pelatihan dan data uji
train_data = ts_data.iloc[:-7]  # Menggunakan data historis untuk pelatihan
test_data = ts_data.iloc[-7:]   # Menggunakan data terakhir sebagai data uji

# Memodelkan data dengan ARIMA
model = ARIMA(train_data, order=(1, 1, 1))
model_fit = model.fit()
```

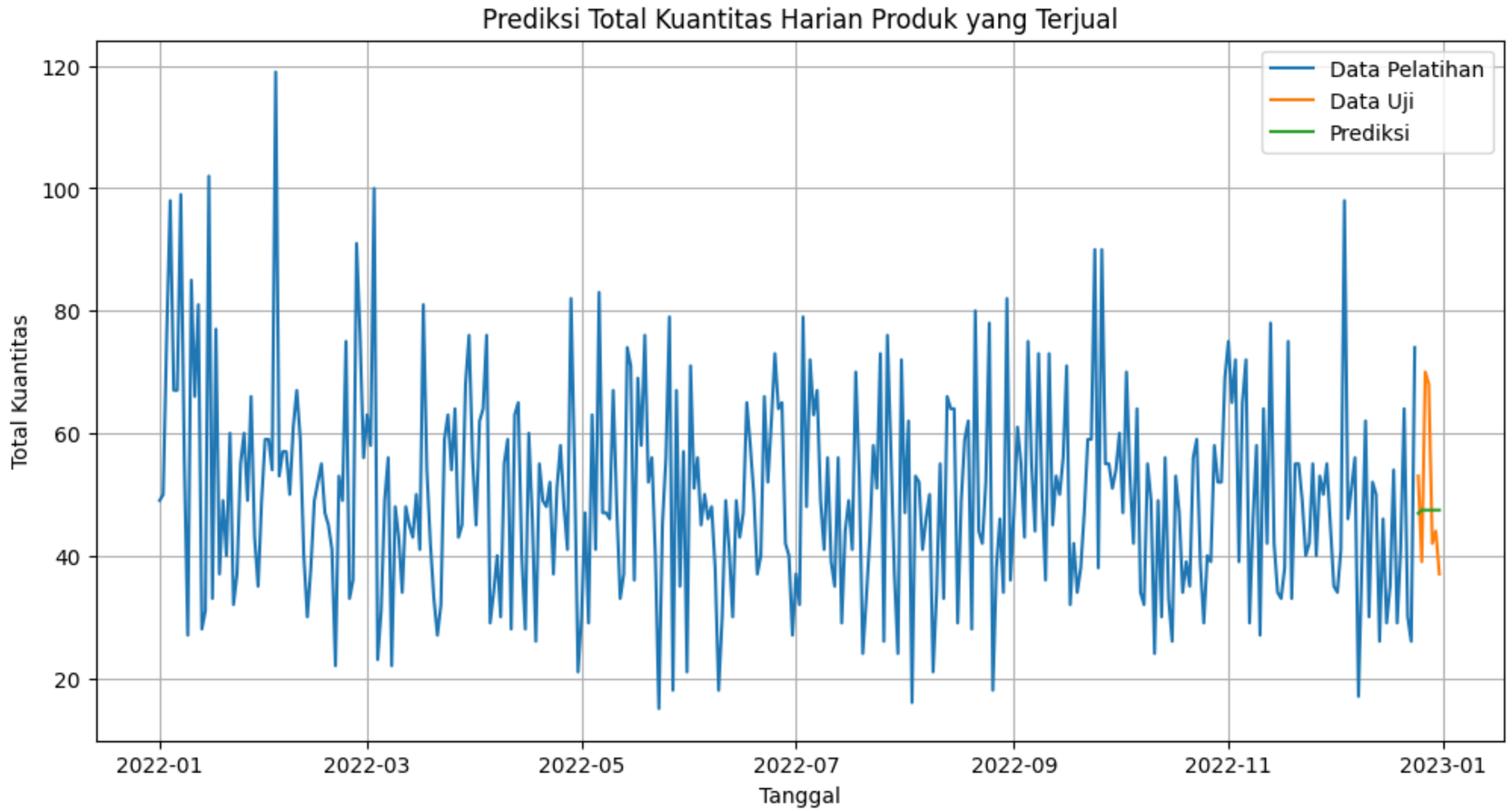
```python
# Membuat prediksi untuk 7 hari ke depan
forecast_steps = 7
forecast = model_fit.forecast(steps=forecast_steps)

# Menampilkan hasil prediksi
print("Prediksi 7 Hari ke Depan:")
print(forecast)

# Plot hasil prediksi
plt.figure(figsize=(12, 6))
plt.plot(train_data.index, train_data.values, label='Data Pelatihan')
plt.plot(test_data.index, test_data.values, label='Data Uji')
plt.plot(test_data.index, forecast, label='Prediksi')
plt.title('Prediksi Total Kuantitas Harian Produk yang Terjual')
plt.xlabel('Tanggal')
plt.ylabel('Total Kuantitas')
plt.legend()
plt.grid(True)
plt.show()
```

DATA PROCESSING

HASIL PEMODELAN

DATA PREDICTION WITH MECHINE LEARNING memprediksi total quantity harian dari product yang terjual

```
Prediksi 7 Hari ke Depan:
2022-12-25    46.919296
2022-12-26    47.453362
2022-12-27    47.442830
2022-12-28    47.443038
2022-12-29    47.443034
2022-12-30    47.443034
2022-12-31    47.443034
Freq: D, Name: predicted_mean, dtype: float64
```



DATA PROCESSING

HASIL CLUSTERING cluster customer-customer yang mirip

```
from sklearn.cluster import KMeans
merged_data = pd.merge(dfc[['CustomerID', 'Marital Status']], dft[['CustomerID', 'TransactionID', 'Qty', 'TotalAmount']], on='CustomerID')

# Melakukan agregasi
agg_data = merged_data.groupby('CustomerID').agg({
    'TransactionID': 'count',
    'Qty': 'sum',
    'TotalAmount': 'sum'
}).reset_index()


kmeans = KMeans(n_clusters=3)
agg_data['Cluster'] = kmeans.fit_predict(agg_data[['TransactionID', 'Qty', 'TotalAmount']])

# Menampilkan hasil clustering
print(agg_data)
```

```
     CustomerID  TransactionID  Qty  TotalAmount  Cluster
0             1             17   60       623300        1
1             2             13   57       392300        2
2             3             15   56       446200        2
3             4             10   46       302500        0
4             5              7   27       268600        0
..          ...            ...  ...          ...      ...
442         443             16   59       485100        1
443         444             18   62       577700        1
444         445             18   68       587200        1
445         446             11   42       423300        2
446         447             13   42       439300        2

[447 rows x 5 columns]
```

THANK YOU!