

Praktikum Data Mining

Minggu Ke-9

Faris Saifullah
3124640034

```
#no.1
import pandas as pd
df = pd.read_csv('transaction.csv')
df.head()
```

	InvoiceNo	StockCode	Qty	InvoiceDate	CustomerID	Country
0	537626	22725	830	12/7/2010 14:57	12347	Iceland
1	537626	22729	948	12/7/2010 14:57	12347	Iceland
2	537626	22195	695	12/7/2010 14:57	12347	Iceland
3	542237	22725	636	1/26/2011 14:30	12347	Iceland
4	542237	22729	536	1/26/2011 14:30	12347	Iceland

Analisa :
menampilkan data yang ada pada file transaction.csv



```
#no.2  
country_counts = df['Country'].value_counts()  
country_counts.head()
```

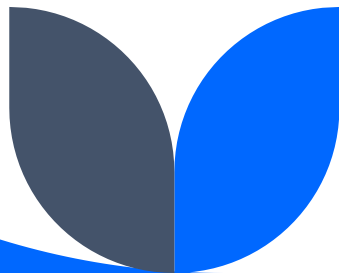
count	
Country	
Germany	2269
France	2109
EIRE	1620
Netherlands	634
Spain	539

Analisa :
menampilkan data kemunculan tiap negara pada dataset
file transaction.csv

```
#no.3
transaction_counts = dataset.groupby(['Country', 'InvoiceNo'])['InvoiceNo'].nunique()
transaction_counts_by_country = transaction_counts.groupby('Country').sum()
transaction_counts_by_country.head()
```

InvoiceNo	
Country	
Australia	44
Austria	12
Bahrain	1
Belgium	84
Brazil	1

Analisa :
menampilkan data banyaknya transaksi pada tiap negara



```
#no.4
from sklearn.cluster import KMeans
import numpy as np

cluster_i = []
cluster_val = []

for i in range(10):
    kmeans = KMeans(n_clusters=3, random_state=i)
    kmeans.fit(np.array(transaksi).reshape(-1, 1))
    sse = kmeans.inertia_
    cluster_i.append(i+1)
    cluster_val.append(sse)

cluster_i
cluster_val

[13229107.198522324,
 16420994.708162924,
 13229107.198522324,
 16420994.708162924,
 13229107.198522324,
 13229107.198522324,
 13229107.198522324,
 13229107.198522324,
 13229107.198522324,
 16420994.708162924]
```

Analisa :
menampilkan clustering pada transaksi dengan K-Means, dengan k=3, sebanyak 10 kali. Setiap kali selesai clustering, lakukan cluster analysis dengan SSE

```
#no.5  
min_index = cluster_val.index(min(cluster_val))  
cluster = cluster_i[min_index]  
cluster
```

```
1
```

Analisa :
mengambil cluster_i yang mempunyai cluster_val terkecil



```
#no.6
kmeans = KMeans(n_clusters=3, random_state=cluster - 1)
kmeans.fit(np.array(transaksi).reshape(-1, 1))
centroid = kmeans.cluster_centers_
centroid

array([[3542.21952146],
       [6173.85081301],
       [1400.13034188]])
```

Analisa :
menentukan posisi centroid dari setiap cluster



```
#no.7
sorted_centroid = np.sort(centroid, axis=0)
sorted_centroid

array([[1400.13034188],
       [3542.21952146],
       [6173.85081301]])
```

Analisa :
melakukan pengurutan posisi centroid secara ascending




```
#no.8
labels = kmeans.labels_
country_labels = pd.DataFrame({'Country': transaksi.index, 'Label': labels})
low_transaction_countries = country_labels[country_labels['Label'] == np.argmin(sorted_centroid)].Country.tolist()
high_transaction_countries = country_labels[country_labels['Label'] == np.argmax(sorted_centroid)].Country.tolist()
medium_transaction_countries = country_labels[
    (country_labels['Label'] != np.argmin(sorted_centroid)) &
    (country_labels['Label'] != np.argmax(sorted_centroid))
].Country.tolist()
print("Negara dengan transaksi rendah:")
print(low_transaction_countries)
print("\nNegara dengan transaksi sedang:")
print(medium_transaction_countries)
print("\nNegara dengan transaksi tinggi:")
print(high_transaction_countries)
```

```
Negara dengan transaksi rendah:
['Australia', 'Austria', 'Belgium', 'Brazil', 'Channel Islands', 'Cyprus', 'Denmark', 'EIRE', 'Finland', 'France', 'Germany', 'Greece', 'Iceland', 'Italy', 'Japan']

Negara dengan transaksi sedang:
['Canada', 'Israel', 'RSA', 'Singapore', 'Switzerland', 'USA', 'United Arab Emirates']

Negara dengan transaksi tinggi:
['Bahrain', 'Czech Republic', 'European Community', 'Lithuania', 'Saudi Arabia', 'Sweden']
```

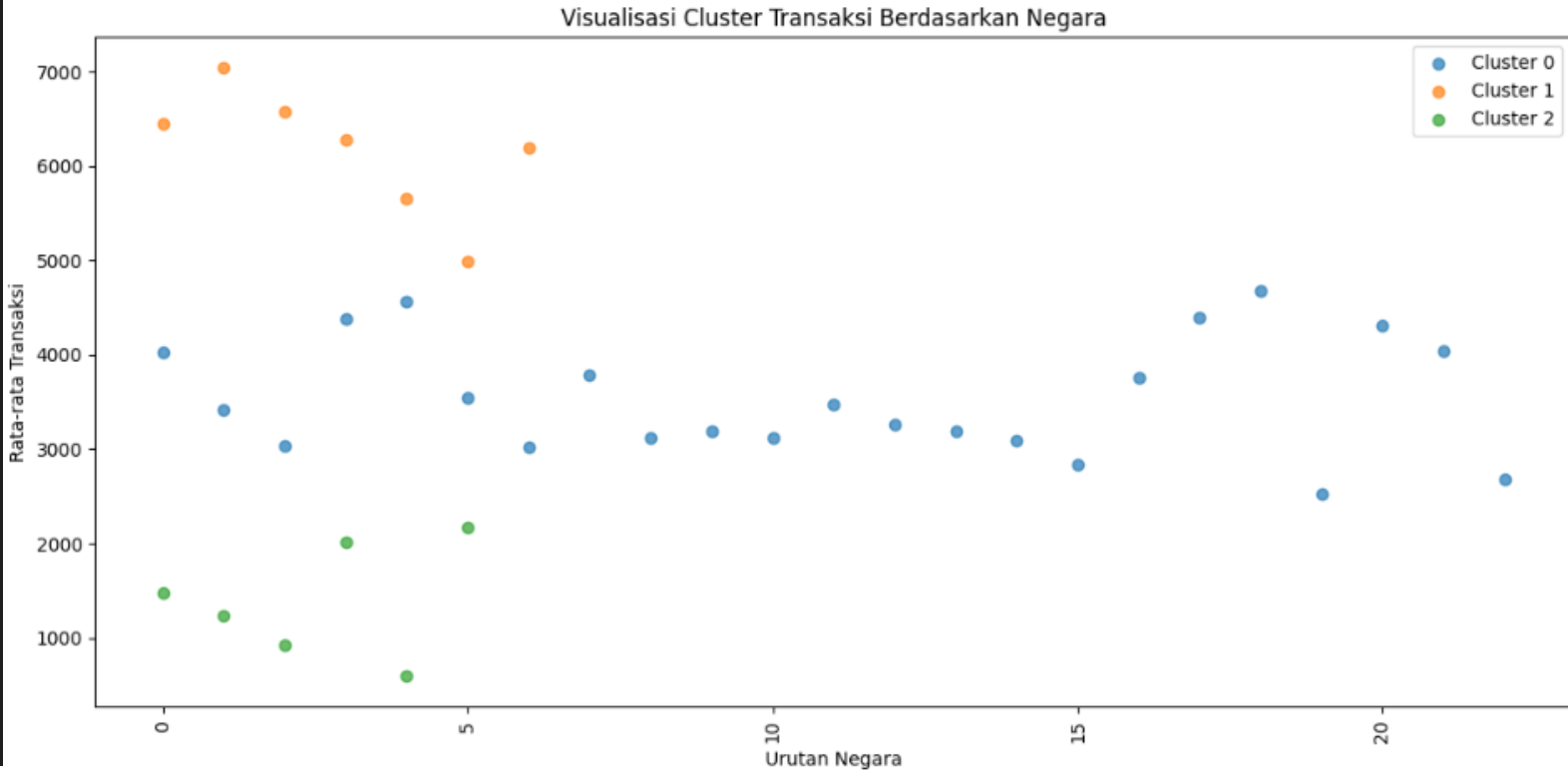
Analisa :

kode ini menggunakan pengelompokan k-means untuk mengelompokkan negara berdasarkan volume transaksinya dan memberikan kategorisasi negara yang jelas ke dalam tiga kelompok: negara dengan transaksi rendah, menengah, dan tinggi.

```
#no.9
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
for label in np.unique(labels):
    subset = country_labels[country_labels['label'] == label]
    plt.scatter(
        range(len(subset)),
        transaksi[subset['Country']],
        label=f'Cluster {label}',
        alpha=0.7
    )

plt.xlabel('Urutan Negara')
plt.ylabel('Rata-rata Transaksi')
plt.title('Visualisasi Cluster Transaksi Berdasarkan Negara')
plt.xticks(rotation=90)
plt.legend()
plt.tight_layout()
plt.show()
```



Analisa :

grafik ini memberikan gambaran visual yang jelas mengenai hasil clustering yang telah dilakukan.