CLUSTERING

NADILA IMAARAH

D4 SAINS DATA TERAPAN A

3323600015

1. DATASET: TRANSACTION.CSV

MENGIMPOR LIBRARY

```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import linkage, fcluster
from scipy.spatial.distance import pdist
import numpy as np
import matplotlib.pyplot as plt
```

LOAD DATASET

```
dataset = pd.read_csv('transaction.csv')
dataset
```

	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
...	...	...	...	...	...	...
10541	543911	21700	455	2/14/2011 12:46	17829	United Arab Emirates
10542	543911	22111	578	2/14/2011 12:46	17829	United Arab Emirates
10543	543911	22112	163	2/14/2011 12:46	17829	United Arab Emirates
10544	564428	23296	545	8/25/2011 11:27	17844	Canada
10545	564428	23294	643	8/25/2011 11:27	17844	Canada

10546 rows × 6 columns

MENGECEK MISSING VALUE

```
print(dataset.isna().sum())
```

```
InvoiceNo      0
StockCode      0
Qty            0
InvoiceDate    0
CustomerID     0
Country        0
dtype: int64
```

Dataset yang digunakan adalah dataset transaction yang terdiri dari beberapa fitur yakni InvoiceNO, StockCode, Qty, InvoiceDate, CustomerID dan country. Setelah itu saya mengecek missing value terlebih dahulu, didapatkan bahwa tidak ada missing value

2. COUNTRY = BERAPA KEMUNCULAN TIAP NEGARA PADA DATASET, DAN TAMPILKAN

```
country_counts = dataset["Country"].value_counts()
country_counts
```

Country	
Germany	2269
France	2109
EIRE	1620
Netherlands	634
Spain	539
Belgium	486
Switzerland	434
Portugal	367
Australia	356
Norway	239
Italy	190
Channel Islands	184
Finland	152
Cyprus	113
Sweden	109
Denmark	98
Japan	92
Austria	88
Poland	80
Israel	61
USA	47
Singapore	45
Unspecified	44
Canada	36
Iceland	35
Greece	33
United Arab Emirates	23
Malta	15
RSA	14

```
RSA                      14
Brazil                    8
Lithuania                 8
Lebanon                   5
European Community        5
Czech Republic            4
Bahrain                   3
Saudi Arabia              1
Name: count, dtype: int64
```

disini saya menhitung jumlah kemunculan nama tiap negara. Disini saya akan menggunakan kolom country saja lalu menggunakan fungsi value_counts() yang akan secara otomatis menghitung banyak kemunculan masing-masing nama negara di kolom country

3. TRANSAKSI = HITUNGLAH BANYAKNYA TRANSAKSI PADA TIAP NEGARA (1 KODE INVOICENO =1 TRANSAKSI)

```
trans_by_country = dataset.groupby('Country')['InvoiceNo'].nunique().reset_index()
trans_by_country.columns = ['Country', 'TransactionCount']
print("\nJumlah transaksi per negara:")
print(trans_by_country)
```

Disini saya akan mengelompokkan data berdarakan negara di kolom country dengan menggunakan fungsi groupby, setelah dikelompokkkan per negara akan dilakukan menghitung jumlah Invoice No setiap negara dengan menggunakan fungsi nunique(). InvoiceNo ini yang dihitung yang nilainya unik tidak duplikat

```
Jumlah transaksi per negara:
                 Country  TransactionCount
0              Australia                44
1                Austria                12
2                Bahrain                 1
3                Belgium                84
4                 Brazil                 1
5                 Canada                 3
6         Channel Islands                21
7                 Cyprus                16
8          Czech Republic                 2
9                Denmark                18
10                  EIRE               224
11    European Community                 3
12               Finland                26
13                France               344
14               Germany               377
15                Greece                 5
16               Iceland                 6
17                Israel                 4
18                 Italy                31
19                 Japan                14
20               Lebanon                 1
21             Lithuania                 2
22                 Malta                 2
23           Netherlands                76
24                Norway                28
25                Poland                17
26              Portugal                43
27                   RSA                 1
28          Saudi Arabia                 1
29             Singapore                 4
30                 Spain                72
31                Sweden                26
32           Switzerland                41
33                   USA                 5
34  United Arab Emirates                 2
35           Unspecified                 8
```

4. CLUSTER = LAKUKAN CLUSTERING PADA TRANSAKSI DENGAN AVERAGE LINKAGE, DENGAN K=3

```
Z = linkage(trans_by_country[['TransactionCount']], method='average')
clusters = fcluster(Z, 3, criterion='maxclust')
trans_by_country['Cluster'] = clusters
```

Selanjutnya dilakukan pengihitungan jarak atau kemiripan antar negara yang berdasarkan kolom TransactionCount dengan menggunakan metode average. Setelah itu akan dilakukan clusterisas, di sini saya akan membagi data menjadi 3 cluster. Dan label cluster ini akan saya tambahkan ke kolom baru pada dataframe

5. CENTROID = TENTUKAN POSISI CENTROID DARI SETIAP CLUSTER

```
centroids = trans_by_country.groupby('Cluster')['TransactionCount'].mean().reset_index()
centroids.columns = ['Cluster', 'Centroid']
centroids
```

	Cluster	Centroid
0	1	18.787879
1	2	360.500000
2	3	224.000000

Selanjutnya data akan di kelompokkan berdasarkan cluster akan dihitung nilai centroid dari setiap cluster. Berdasarkan output cluster 1 menunjukan centroid yang paling kecil hanya sebesar 18.79 kemudian cluster 2 memiliki nilai rata-rata sebesar 360.500 dan cluster 3 memiliki nilai rata-rata sebesar 224.000

6. SORTED = LAKUKAN PENGURUTAN POSISI CENTROID SECARA ASCENDING

```
sorted_centroids = centroids.sort_values('Centroid').reset_index(drop=True)
print(sorted_centroids)
```

✓ 0.0s Py

	Cluster	Centroid
0	1	18.787879
1	3	224.000000
2	2	360.500000

Hasil centroid sebelumnya akan diurutkan pada tahap ini. Pengurutan didasarkan pada value dari transaksi dengan metode pengurutan ascending (dari kecil ke besar).

7. INDEKS TERDEPAN DARI CENTROID SETELAH PENGURUTAN, MENGINDIKASIKAN CLUSTER TRANSANKSI RENDAH HINGGA TERTINGGI.TAMPILKAN NEGARA MANA SAJA YANG TRANSAKSINYA RENDAH, SEDANG DAN TINGGI

```
level_labels = ['Rendah', 'Sedang', 'Tinggi']
cluster_level_map = {
    cluster_id: level
    for cluster_id, level in zip(sorted_centroids['Cluster'], level_labels)
}
trans_by_country['Level'] = trans_by_country['Cluster'].map(cluster_level_map)
remap_cluster_id = {old: new for old, new in zip(sorted_centroids['Cluster'], [1, 2, 3])}
trans_by_country['Cluster'] = trans_by_country['Cluster'].map(remap_cluster_id)
```

```
print("\nHasil clustering semua negara:")
print(trans_by_country.sort_values('TransactionCount',
                                   ascending=True)[['Country', 'TransactionCount',
                                                    'Cluster', 'Level']])
```

✓ 0.0s

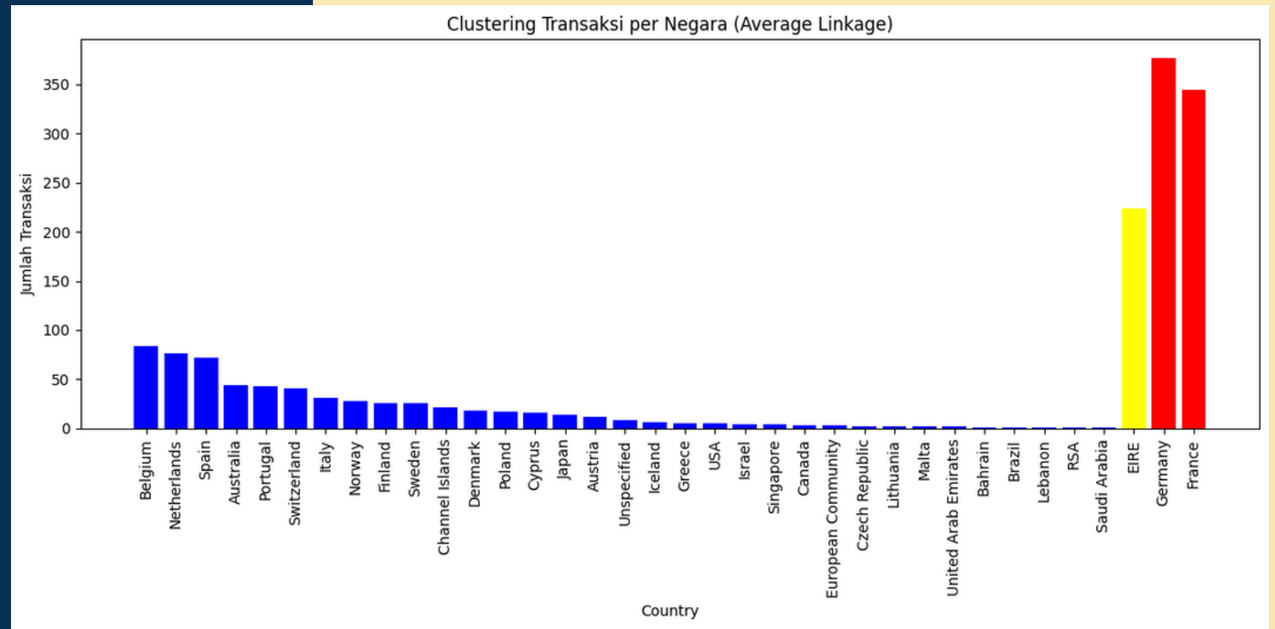
```
Hasil clustering semua negara:
                 Country  TransactionCount  Cluster   Level
2                Bahrain                 1        1  Rendah
4                 Brazil                 1        1  Rendah
28          Saudi Arabia                 1        1  Rendah
27                   RSA                 1        1  Rendah
20               Lebanon                 1        1  Rendah
8          Czech Republic                2        1  Rendah
22                 Malta                 2        1  Rendah
21             Lithuania                 2        1  Rendah
34  United Arab Emirates                 2        1  Rendah
5                Canada                  3        1  Rendah
11    European Community                 3        1  Rendah
17                Israel                 4        1  Rendah
29             Singapore                 4        1  Rendah
33                   USA                 5        1  Rendah
15                Greece                 5        1  Rendah
16               Iceland                 6        1  Rendah
35           Unspecified                 8        1  Rendah
1                Austria                12        1  Rendah
19                 Japan                14        1  Rendah
7                 Cyprus                16        1  Rendah
25                Poland                17        1  Rendah
9                 Denmark               18        1  Rendah
...
3                Belgium                84        1  Rendah
10                 EIRE                224        2  Sedang
13               France                344        3  Tinggi
14              Germany                377        3  Tinggi
Output is truncated. View as a scrollable element or open in a text editor. Adju
```

Pada tahap ini akan dilakukan labeling berdasarkan nilai centroid yang telah di dapatkan di tahap sebelumnya. Dan membuat rules jika cluster 1 menunjukkan negara tersebut memiliki jumlah transaksi yang sedikit dan masuk ke kategori rendah, cluster 2 menunjuukan negara tersebut memiliki jumlah transaksi yang sedang dan cluster 3 menunjukkan negara tersebut memiliki jumlah cluster yang banyak dan masuk ke kategori tinggi. Berdasarkan output negara yang termasuk cluster sedang yakni EIRE dan negara yang termasuk cluster tinggi yakni France dan Germany

8. Visualisasi dengan warna yang berbeda untuk hasil cluster (no. 7), dimana sumbu X=urutan country dan sumbu Y=transaksi

```
level_order = ['Rendah', 'Sedang', 'Tinggi']
trans_by_country['Level'] = pd.Categorical(trans_by_country['Level'], categories=level_order, ordered=True)
trans_by_country_sorted = trans_by_country.sort_values(['Level', 'TransactionCount'], ascending=[True, False])

color_map = {'Rendah': 'blue', 'Sedang': 'yellow', 'Tinggi': 'red'}
colors = trans_by_country_sorted['Level'].map(color_map)
plt.figure(figsize=(12, 6))
plt.bar(trans_by_country_sorted['Country'], trans_by_country_sorted['TransactionCount'], color=colors)
plt.xticks(rotation=90)
plt.xlabel('Country')
plt.ylabel('Jumlah Transaksi')
plt.title('Clustering Transaksi per Negara (Average Linkage)')
plt.tight_layout()
plt.show()
```



Setelah menambahkan label cluster dan melakuakn sorting cluster maka disini saya juga menampilkan visualisasi. Dapat disimpulkan bahwa Sebagian besar negara berada pada cluster 1 yang menunjukan cluster rendah. Hanya ada satu negara yang masuk cluster 2 yang menunjukkan kategori sedang dan hanya ada dua negara yang masuk ke cluster 3 yang menunjuukan kategori tinggi

TERIMA KASIH