

# Tugas Materi 10

Zul Fauzi Oktaviansyah

2110181056

3 – D4 IT - B

# Code 1

```
In [2]: dataset = pd.read_csv('transaction.csv')
dataset
```

Out[2]:

|       | 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

Membaca data csv transaction

# Code 2

```
In [3]: country = pd.DataFrame(dataset['Country'].value_counts())  
country = country.sort_index()  
country
```

Out[3]:

| Country            |      |
|--------------------|------|
| Australia          | 356  |
| Austria            | 88   |
| Bahrain            | 3    |
| Belgium            | 486  |
| Brazil             | 8    |
| Canada             | 36   |
| Channel Islands    | 184  |
| Cyprus             | 113  |
| Czech Republic     | 4    |
| Denmark            | 98   |
| EIRE               | 1620 |
| European Community | 5    |
| Finland            | 152  |

Menampilkan total kemunculan tiap2 negara

# Code 3

```
In [4]: totalTransaction = pd.DataFrame(dataset['Country'].value_counts())  
totalTransaction = totalTransaction.sort_index()  
totalTransaction
```

Out[4]:

| Country            |      |
|--------------------|------|
| Australia          | 356  |
| Austria            | 88   |
| Bahrain            | 3    |
| Belgium            | 486  |
| Brazil             | 8    |
| Canada             | 36   |
| Channel Islands    | 184  |
| Cyprus             | 113  |
| Czech Republic     | 4    |
| Denmark            | 98   |
| EIRE               | 1620 |
| European Community | 5    |
| Finland            | 152  |
| France             | 2109 |

Menampilkan total seluruh transaksi yang dilakukan

# Code 4

```
In [5]: quantityEachCountry = pd.DataFrame(dataset.groupby(['Country'])['Qty'].sum())
quantityEachCountry
```

Out[5]:

|                    | Qty    |
|--------------------|--------|
| Country            |        |
| Australia          | 177157 |
| Austria            | 41043  |
| Bahrain            | 1470   |
| Belgium            | 254748 |
| Brazil             | 4389   |
| Canada             | 19349  |
| Channel Islands    | 95964  |
| Cyprus             | 56814  |
| Czech Republic     | 2479   |
| Denmark            | 54340  |
| EIRE               | 848706 |
| European Community | 2765   |
| Finland            | 81068  |

Menampilkan total quntity tiap2 negara

# Code 5

```
In [6]: averageQuantity = quantityEachCountry.values / totalTransaction.values  
averageQuantity
```

```
Out[6]: array([[497.63202247],  
               [466.39772727],  
               [490.        ],  
               [524.17283951],  
               [548.625     ],  
               [537.47222222],  
               [521.54347826],  
               [502.77876106],  
               [619.75      ],  
               [554.48979592],  
               [523.89259259],  
               [553.        ],  
               [533.34210526],  
               [521.55381697],  
               [518.35698546],  
               [526.21212121],  
               [560.17142857],  
               [462.06557377],  
               [519.83684211],  
               [470.19565217],  
               [567.2       ],  
               [503.25      ],  
               [501.13333333],  
               [527.70657007]])
```

Menghitung average quantity tiap Negara = total quantity / total transaction

# Code 6

```
In [7]: quantityEachCountry['Qty'] = averageQuantity  
quantityEachCountry
```

Out[7]:

|                    | Qty        |
|--------------------|------------|
| Country            |            |
| Australia          | 497.632022 |
| Austria            | 466.397727 |
| Bahrain            | 490.000000 |
| Belgium            | 524.172840 |
| Brazil             | 548.625000 |
| Canada             | 537.472222 |
| Channel Islands    | 521.543478 |
| Cyprus             | 502.778761 |
| Czech Republic     | 619.750000 |
| Denmark            | 554.489796 |
| EIRE               | 523.892593 |
| European Community | 553.000000 |
| Finland            | 533.342105 |

Mereplace value total quantity dengan average quantity

# Code 7

```
In [8]: from sklearn.cluster import KMeans
```

```
In [9]: cluster_i = []
        cluster_val = []

        for i in range(10):
            clustering = KMeans(n_clusters = 3, init = 'random', n_init = 1, max_iter = 5).fit(quantityEachCountry)
            cluster_i.append(clustering)
            cluster_val.append(clustering.inertia_)
            print(clustering, clustering.inertia_)
```

```
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 9849.347473638736
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8650.598826148962
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8053.464374542196
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8257.151454676508
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8257.151454676508
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8053.464374542196
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 9849.347473638736
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8053.464374542196
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8257.151454676508
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8257.151454676508
```

Menghitung clustering kmean 10x tanpa bantuan optimasi dari library dengan  
Menset init = 1



# Code 8

```
In [10]: indexCluster = cluster_val.index(min(cluster_val))
         clusterOfMinVal = cluster_i[indexCluster]

         print(clusterOfMinVal, clusterOfMinVal.inertia_)

KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8053.464374542196

In [11]: centroid = clusterOfMinVal.cluster_centers_
         centroid

Out[11]: array([[486.74185073],
                [536.64580463],
                [605.875      ]])
```

Mencari cluster dengan val minimal dan menampilkan centroidnya

# Code 9

```
In [12]: sortedIndex = np.argsort(centroid.sum(axis=1))
         zeroArray = np.zeros_like(sortedIndex)
         zeroArray[sortedIndex] = np.arange(3)

         sortedCentroid = centroid[zeroArray]
         sortedCentroid
```

```
Out[12]: array([[486.74185073],
               [536.64580463],
               [605.875      ]])
```

---

```
In [13]: sortedLabel = zeroArray[clusterOfMinVal.labels_]
         sortedLabel
```

```
Out[13]: array([0, 0, 0, 1, 1, 1, 1, 0, 2, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0,
               0, 1, 1, 1, 0, 0, 2, 1, 1, 1, 1, 1, 1, 1, 0], dtype=int64)
```

Mengurutkan centroid dan label

# Code 10

```
In [14]: label_2 = (sortedLabel == 2).nonzero()  
countryHigh = quantityEachCountry.index[label_2]  
countryHigh
```

```
Out[14]: Index(['Czech Republic', 'Saudi Arabia'], dtype='object', name='Country')
```

```
In [15]: label_1 = (sortedLabel == 1).nonzero()  
countryMid = quantityEachCountry.index[label_1]  
countryMid
```

```
Out[15]: Index(['Belgium', 'Brazil', 'Canada', 'Channel Islands', 'Denmark', 'EIRE',  
               'European Community', 'Finland', 'France', 'Germany', 'Greece',  
               'Iceland', 'Italy', 'Lebanon', 'Netherlands', 'Norway', 'Poland',  
               'Singapore', 'Spain', 'Sweden', 'Switzerland', 'USA',  
               'United Arab Emirates'],  
              dtype='object', name='Country')
```

```
In [16]: label_0 = (sortedLabel == 0).nonzero()  
countryLow = quantityEachCountry.index[label_0]  
countryLow
```

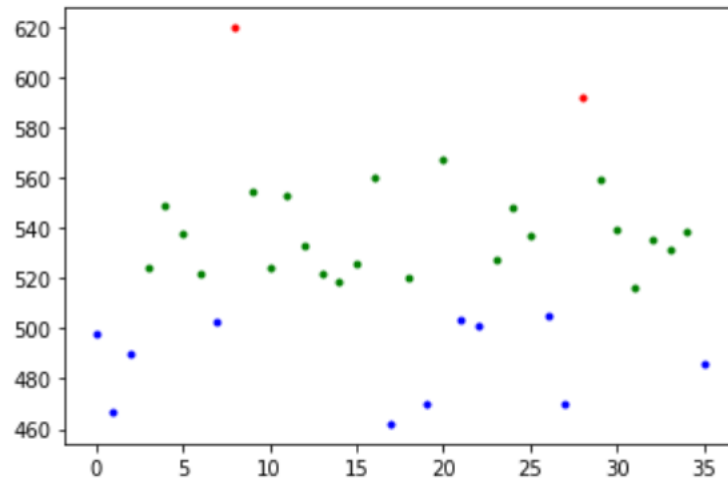
```
Out[16]: Index(['Australia', 'Austria', 'Bahrain', 'Cyprus', 'Israel', 'Japan',  
               'Lithuania', 'Malta', 'Portugal', 'RSA', 'Unspecified'],  
              dtype='object', name='Country')
```

Mengkelompokkan berdasarkan transaksi tinggi, sedang, rendah

# Code 11

```
In [17]: import matplotlib.pyplot as plt
```

```
In [18]: plt.plot(label_2, quantityEachCountry.iloc[label_2].to_numpy().reshape((1, -1)), 'r.')  
plt.plot(label_1, quantityEachCountry.iloc[label_1].to_numpy().reshape((1, -1)), 'g.')  
plt.plot(label_0, quantityEachCountry.iloc[label_0].to_numpy().reshape((1, -1)), 'b.')  
plt.show()
```



Menampilkan hasil pengelompokkan transaksi tinggi, sedang, rendah