CASE BASED - 2

COUNTRY DATA

WITH K-MEANS ALGORITHM

JOVIDIA LAVIOSA - 1301204142
IF4401
Kode Dosen : BDP

HELP International adalah LSM kemanusiaan internasional yang berkomitmen untuk memerangi kemiskinan dan menyediakan fasilitas dasar dan bantuan kepada orang-orang di negara-negara terbelakang selama masa bencana dan bencana alam.

HELP International telah berhasil mengumpulkan uang sekitar 10 juta dolar. Langkah selanjutnya adalah CEO LSM perlu memutuskan bagaimana menggunakan uang secara strategis dan efektif. Oleh karena itu, perlu adanya pengkategorian negara menggunakan beberapa faktor sosial-ekonomi dan kesehatan yang menentukan pembangunan negara secara keseluruhan.

Tujuan pengelompokan negara ini adalah untuk membantu LSM internasional menentukan negara mana saja yang membutuhkan bantuan. Dalam kasus ini negara-negara dikelompokan berdasarkan yang membutuhkan bantuan, mungkin membutuhkan bantuan, dan tidak membutuhkan bantuan berdasarkan maju, berkembang atau kurang berkembangnya suatu negara. Pemrosesan data ini menggunakan cara analisis pengelompokan dengan algoritma unsupervised learning dengan metode **K-Means Clustering.**

Ada 10 klasifikasi data di dalamnya diantaranya,

1. **Country** berisikan nama negara
2. **Child_mort** yaitu kematian anak di bawah usia 5 tahun per 1000 kelahiran hidup.
3. **Exports** yaitu hasil barang ekspor per kapita. Diberikan sebagai persentase usia dari PDB per kapita.
4. **Health** yaitu Total pengeluaran untuk pelayanan kesehatan per kapita. Diberikan sebagai persentase usia dari PDB per kapita.
5. **Imports** yaitu hasil barang impor per kapita diberikan sebagai persentase usia dari PDB per kapita.
6. **Income** yaitu penghasilan per individu.
7. **Inflation** yaitu Pengukuran tingkat pertumbuhan tahunan Total PDB.
8. **Life_expec** yaitu Rata-rata jumlah tahun seorang anak yang baru lahir akan hidup jika pola kematian saat ini dihapus.
9. **Total_fer** yaitu Jumlah anak yang akan dilahirkan oleh setiap wanita jika tingkat kesuburan usia saat ini tetap
10. **Gdpp** atau PDB per kapita. Dihitung sebagai Total PDB dibagi dengan total populasi.

IKHTISAR KUMPULAN DATA COUNTRY

Import Library dan Data

```python
# Mengimpor library untuk membaca dateset
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

1

- Library numpy memberi kemudahan dalam komputasi khususnya tipe data numerik.
- Library pandas berfungsi untuk membuat tabel, mengubah dimensi data dan mengecek data.
- Library mathplotlib.pyplot berfungsi untuk membuat sebuah plot
- Library seaborn untuk membentuk grafik
- Library plotly.express berfungsi untuk keperluan visualisasi data

```python
# Mengimpor data
df_country = pd.read_csv("Country-data.csv")
df_country
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200
...	...	...	...	...	...	...	...	...	...	...
162	Vanuatu	29.2	46.6	5.25	52.7	2950	2.62	63.0	3.50	2970
163	Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.47	13500
164	Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.95	1310
165	Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.67	1310
166	Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.40	1460

167 rows × 10 columns

2

```python
# Menampilkan Baris dan Kolom
df_country.shape
```

```
(167, 10)
```

3

```python
# Mengapus Keterangan Kolom Country
list_Country = list(df_country.columns)
list_Country.remove('country')
```

4

```python
# Menampilkan Info
df_country.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   country     167 non-null    object
 1   child_mort  167 non-null    float64
 2   exports     167 non-null    float64
 3   health      167 non-null    float64
 4   imports     167 non-null    float64
 5   income      167 non-null    int64
 6   inflation   167 non-null    float64
 7   life_expec  167 non-null    float64
 8   total_fer   167 non-null    float64
 9   gdpp        167 non-null    int64
dtypes: float64(7), int64(2), object(1)
memory usage: 13.2+ KB
```

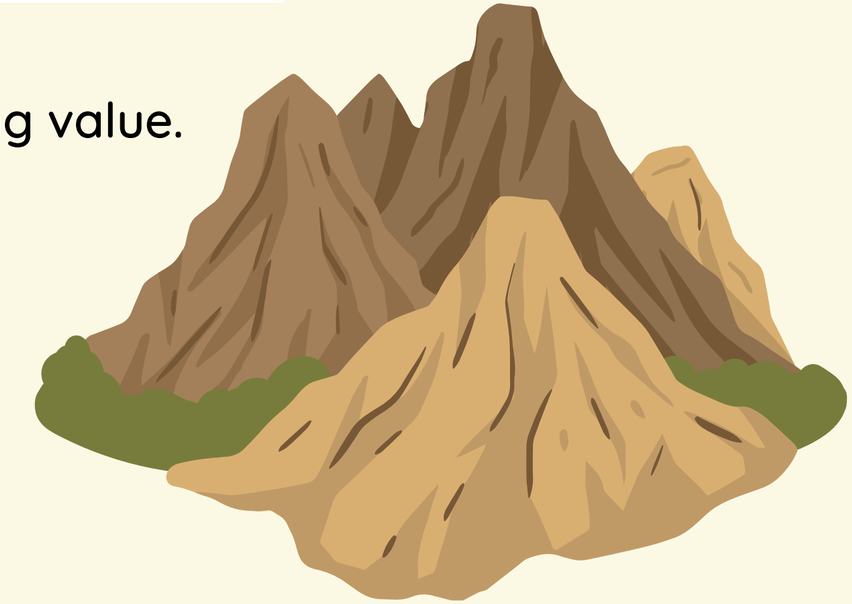
5

```python
# Cek missing value

df_country.isnull().sum()
```

```
country       0
child_mort    0
exports       0
health        0
imports       0
income        0
inflation     0
life_expec    0
total_fer     0
gdpp          0
dtype: int64
```

6

Tidak terdapat missing value.



Rangkuman Data

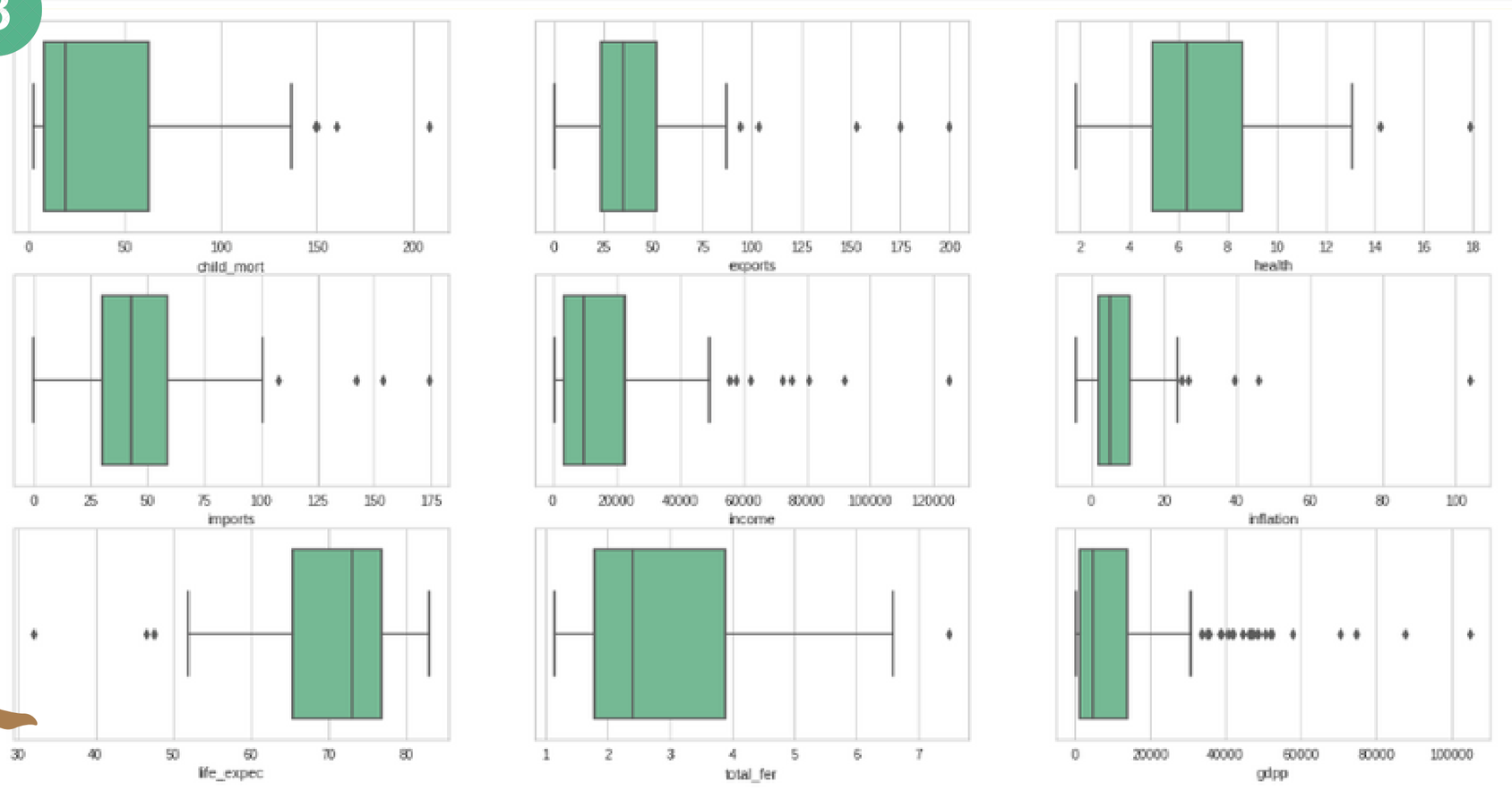
```
# Rangkuman data

df_country.describe()
```

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
count	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000
mean	38.270060	41.108976	6.815689	46.890215	17144.688623	7.781832	70.555689	2.947964	12964.155689
std	40.328931	27.412010	2.746837	24.209589	19278.067698	10.570704	8.893172	1.513848	18328.704809
min	2.600000	0.109000	1.810000	0.065900	609.000000	-4.210000	32.100000	1.150000	231.000000
25%	8.250000	23.800000	4.920000	30.200000	3355.000000	1.810000	65.300000	1.795000	1330.000000
50%	19.300000	35.000000	6.320000	43.300000	9960.000000	5.390000	73.100000	2.410000	4660.000000
75%	62.100000	51.350000	8.600000	58.750000	22800.000000	10.750000	76.800000	3.880000	14050.000000
max	208.000000	200.000000	17.900000	174.000000	125000.000000	104.000000	82.800000	7.490000	105000.000000

Menampilkan Boxplot

8

```
# Menampilkan Boxplot
fig, ax = plt.subplots(nrows = 3, ncols = 3, figsize=(20, 10))
ax=ax.flatten()
int_cols= df_country.select_dtypes(exclude='object').columns
for x, i in enumerate(int_cols):
    sns.boxplot(x=df_country[i], ax=ax[x], color='#6ac493')
```



Menampilkan Corrplot

9

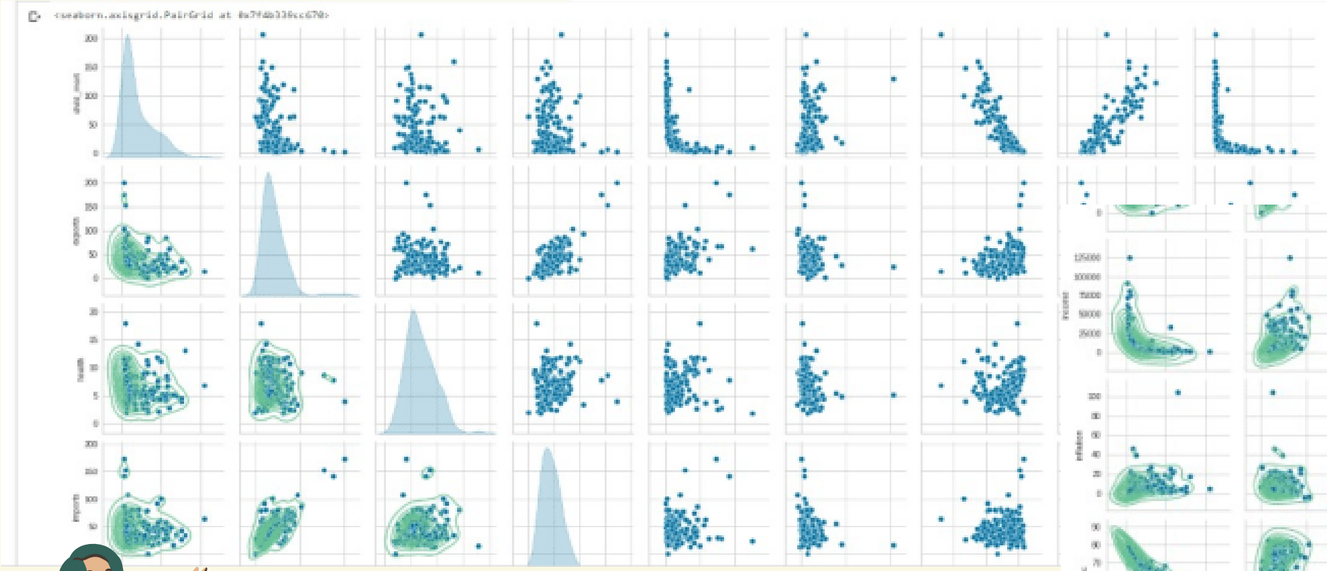
```
# Menampilkan Corrplot

corr = df_country.corr()
mask = np.triu(np.ones_like(corr, dtype=np.bool))
f, ax = plt.subplots(figsize=(10, 10))
cmap = sns.light_palette('#6ac493', as_cmap=True)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=None,center=0,square=True, annot=True, linewidths=.5, cbar_kws={"shrink": .9})
plt.show()
```

```
`np.bool` is a deprecated alias for the builtin `bool`. To silence this warning, use `bool` by itself. Doing this will no
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
```

Menampilkan Pairplot

```
# Menampilkan Pairplot

g = sns.pairplot(data=df_country,diag_kind='kde')
g.map_lower(sns.kdeplot, color="#6ac493")
```

10



PRA-PEMPROSESAN DATA

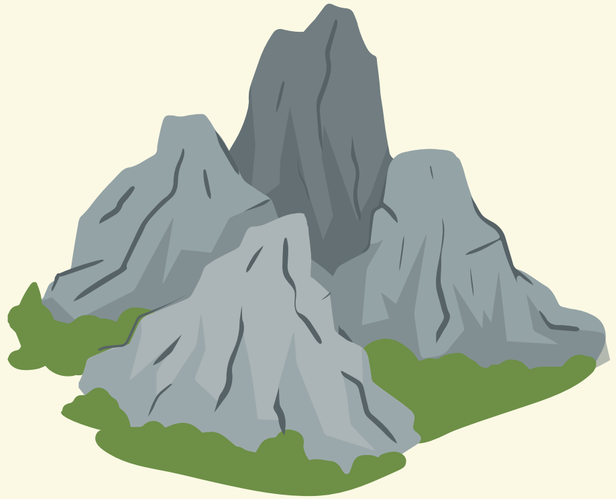
Import Library

1

```
# Import Library
from sklearn.preprocessing import StandardScaler
df_country_scaled = StandardScaler().fit_transform(df_country.drop(['country'], axis=1))
```

- Library sklearn.preprocessing berfungsi untuk melakukan normalisasi data agar data yang digunakan tidak memiliki penyimpangan yang besar.

Menampilkan Baris dan kolom. Karena country tadi sudah dihapus jadi kolomnya 9.



2

```
# Menampilkan Keterangan Baris dan Kolom
df_country_scaled.shape
```

```
(167, 9)
```

Data Scaled

3

```
# Data Scaled

df_country_scaled = pd.DataFrame(df_country_scaled)
df_country_scaled.columns = list_Country
df_country_scaled
```

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	1.291532	-1.138280	0.279088	-0.082455	-0.808245	0.157336	-1.619092	1.902882	-0.679180
1	-0.538949	-0.479658	-0.097016	0.070837	-0.375369	-0.312347	0.647866	-0.859973	-0.485623
2	-0.272833	-0.099122	-0.966073	-0.641762	-0.220844	0.789274	0.670423	-0.038404	-0.465376
3	2.007808	0.775381	-1.448071	-0.165315	-0.585043	1.387054	-1.179234	2.128151	-0.516268
4	-0.695634	0.160668	-0.286894	0.497568	0.101732	-0.601749	0.704258	-0.541946	-0.041817
...	...	...	...	...	...	...	...	...	...
162	-0.225578	0.200917	-0.571711	0.240700	-0.738527	-0.489784	-0.852161	0.365754	-0.546913
163	-0.526514	-0.461363	-0.695862	-1.213499	-0.033542	3.616865	0.546361	-0.316678	0.029323
164	-0.372315	1.130305	0.008877	1.380030	-0.658404	0.409732	0.286958	-0.661206	-0.637754
165	0.448417	-0.406478	-0.597272	-0.517472	-0.658924	1.500916	-0.344633	1.140944	-0.637754
166	1.114951	-0.150348	-0.338015	-0.662477	-0.721358	0.590015	-2.092785	1.624609	-0.629546

167 rows × 9 columns

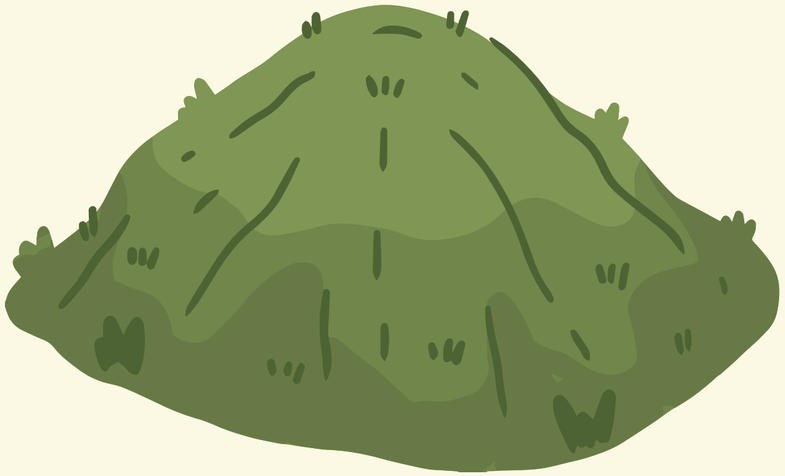
METODE K-MEANS CLUSTERING

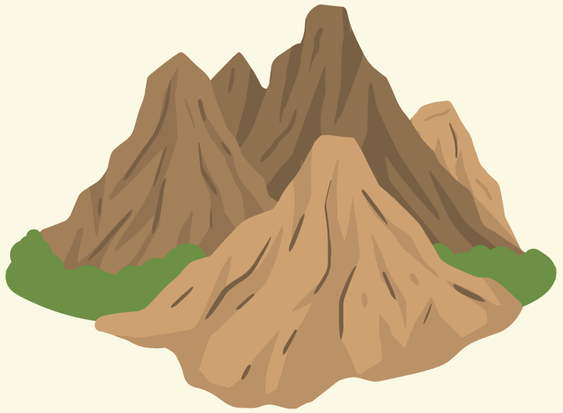
Import Library

```
#Import Library

from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer
```

1

- Import Library untuk melakukan metode K-Means dan pembuatan visualisasi Elbow

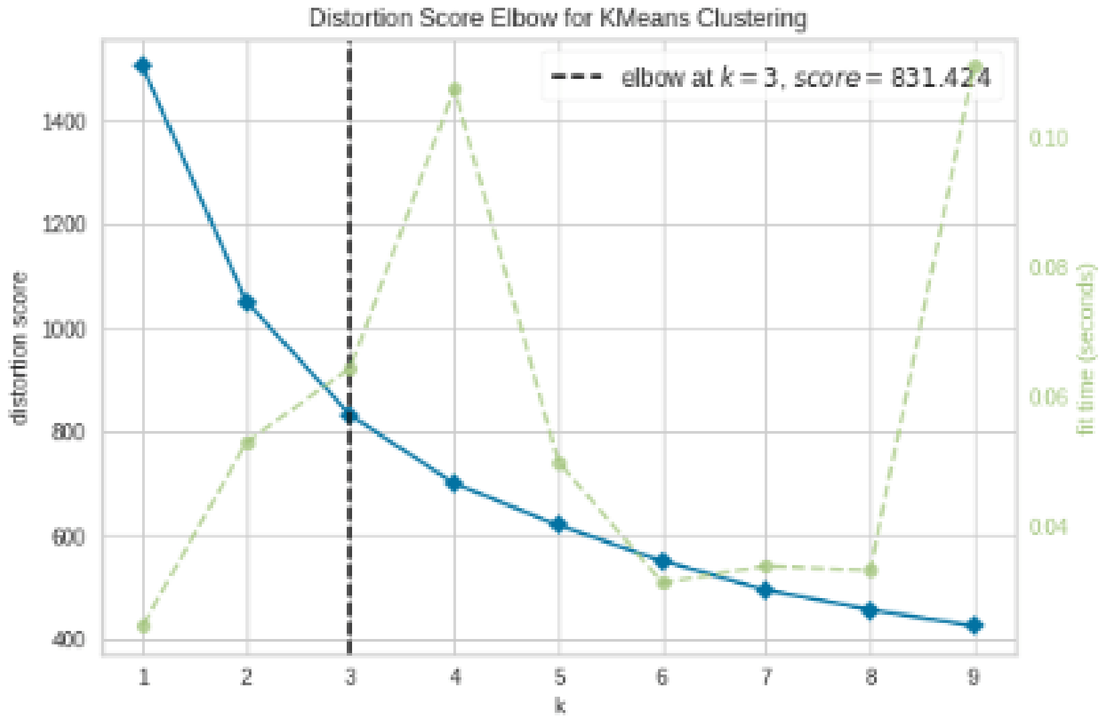




Pembuatan Model K-Means dan visualisasi Elbow

2

```
# Pembuatan Model K-Means dan visualisasi Elbow

model = KMeans()
visualize = KElbowVisualizer(model, k=(1,10))
visualize.fit(df_country_scaled)
visualize.poof()
```




```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe75e62b940>
```

Inisiasi model K-Means

3

```
# Model K-Means

model = KMeans(n_clusters=3, random_state=1)
model.fit(df_country_scaled)
```

```
KMeans(n_clusters=3, random_state=1)
```

Menampilkan Model Label

4

```
# Menampilkan Model Label

model.labels_
```

```
array([0, 1, 1, 0, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 1, 0, 1, 1, 1, 0,
       1, 2, 1, 0, 0, 1, 0, 2, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 2, 2,
       2, 1, 1, 1, 1, 0, 0, 1, 1, 2, 2, 0, 0, 1, 2, 0, 2, 1, 1, 0, 0, 1,
       0, 1, 2, 1, 1, 1, 0, 2, 2, 2, 1, 2, 1, 1, 0, 0, 2, 1, 0, 1, 1, 0,
       0, 1, 1, 2, 1, 0, 0, 1, 1, 0, 2, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1,
       2, 2, 0, 0, 2, 1, 0, 1, 1, 1, 1, 1, 2, 2, 1, 1, 0, 1, 1, 0, 1, 1,
       0, 2, 2, 2, 1, 0, 2, 2, 1, 1, 0, 1, 2, 2, 1, 0, 1, 0, 0, 1, 1, 1,
       1, 0, 1, 2, 2, 2, 1, 1, 1, 1, 1, 0, 0], dtype=int32)
```

Pengklasterisasi atau pembagian data ke dalam cluster

5

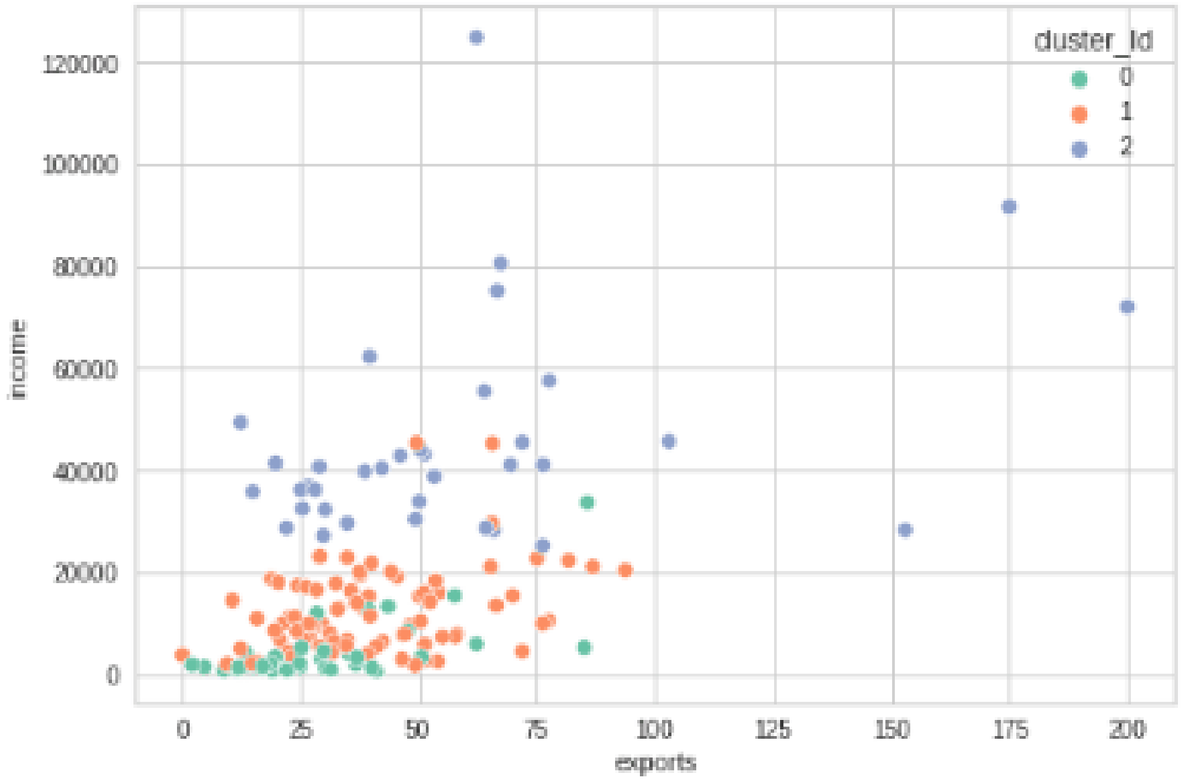
```
# Pengklasterisasi

df_country['cluster_Id'] = model.labels_
df_country.head()
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp	cluster_Id
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553	0
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090	1
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460	1
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530	0
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200	1

Scatterplot

```
# Scatterplot expor dan penghasilan

sns.scatterplot( x='exports', y='income', hue = 'cluster_Id' ,data = df_country, palette = 'Set2')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe75c3d8dc0>
```

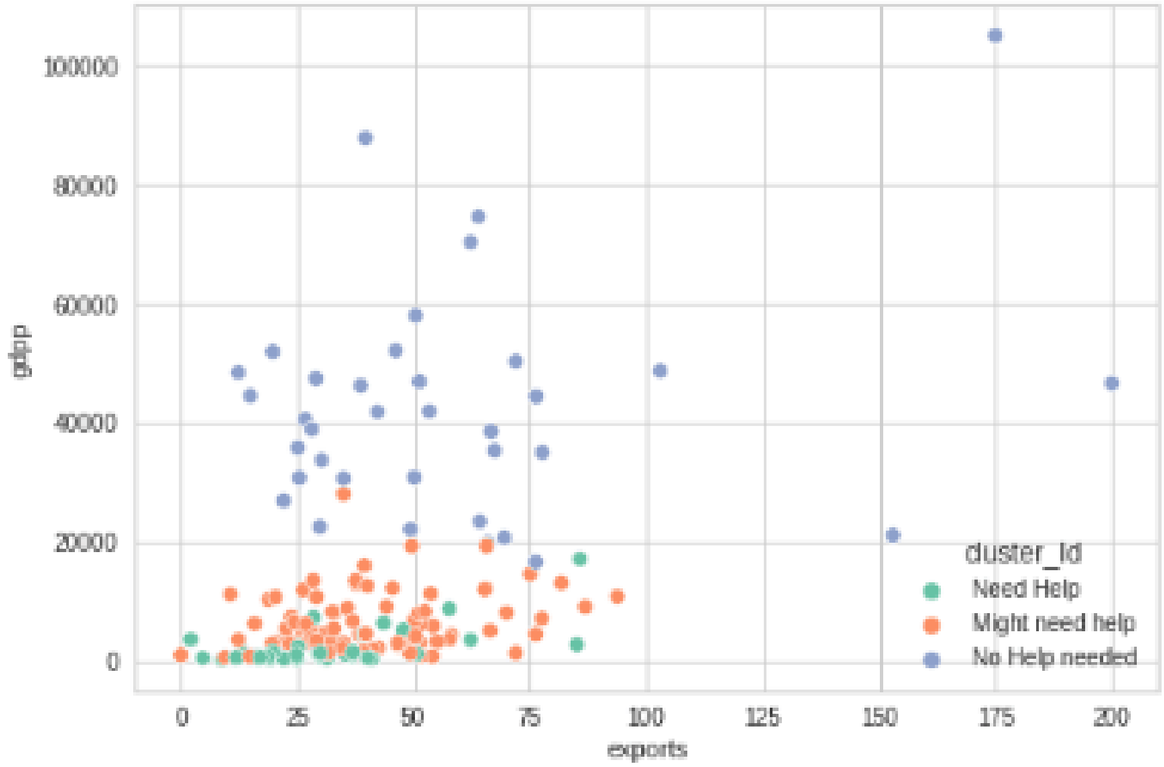


6

Scatterplot

```
# Scatterplot expor dan gdp

sns.scatterplot(x='exports', y='gdpp' , hue = 'cluster_Id' ,data = df_country, palette = 'Set2')
```

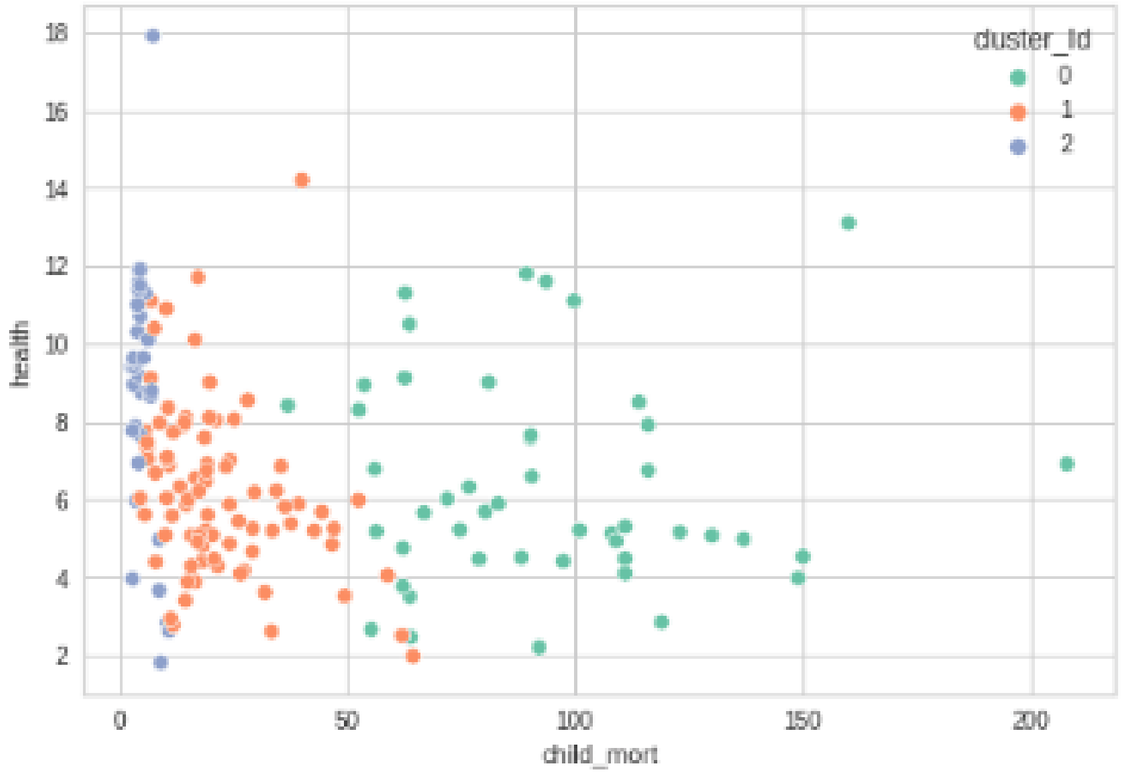
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe75c1a6580>
```



Scatterplot

```
# Scatterplot kelahiran dan kesehatan

sns.scatterplot(x='child_mort', y='health' , hue = 'cluster_Id' ,data = df_country, palette = 'Set2')
```

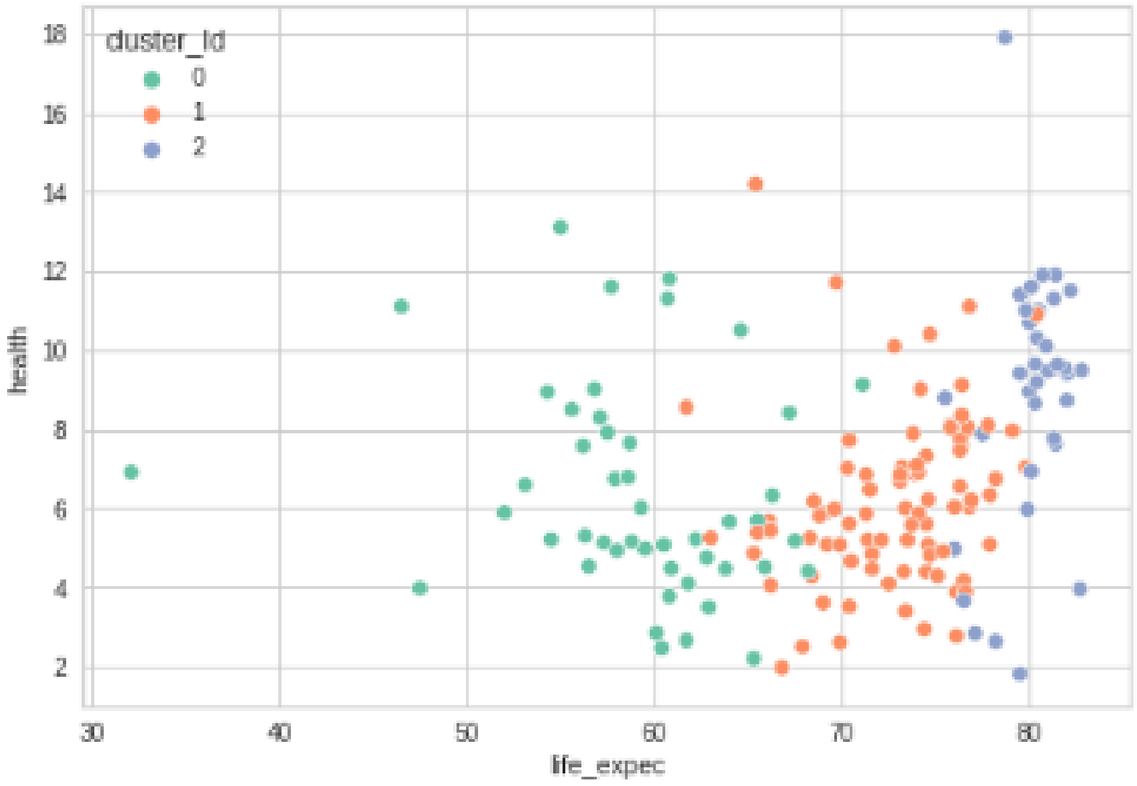
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe75c2c6580>
```



Scatterplot

```
# Scatterplot life expec dan kesehatan

sns.scatterplot(x='life_expec', y='health' , hue = 'cluster_Id' ,data = df_country, palette = 'Set2')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe75c28a6a0>
```



Menampilkan daftar negara per cluster

```
# Menampilkan daftar negara perkluster

df_country.groupby(['cluster_Id','country']).mean()
```

cluster_Id	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610.0	9.440	56.2	5.82	553.0
	Angola	119.0	62.3	2.85	42.9	5900.0	22.400	60.1	6.16	3530.0
	Benin	111.0	23.8	4.10	37.2	1820.0	0.885	61.8	5.36	758.0
	Botswana	52.5	43.6	8.30	51.3	13300.0	8.920	57.1	2.88	6350.0
	Burkina Faso	116.0	19.2	6.74	29.6	1430.0	6.810	57.9	5.87	575.0
...	...	...	...	...	...	...	...	...	...	...
2	Sweden	3.0	46.2	9.63	40.7	42900.0	0.991	81.5	1.98	52100.0
	Switzerland	4.5	64.0	11.50	53.3	55500.0	0.317	82.2	1.52	74600.0
	United Arab Emirates	8.6	77.7	3.66	63.6	57600.0	12.500	76.5	1.87	35000.0
	United Kingdom	5.2	28.2	9.64	30.8	36200.0	1.570	80.3	1.92	38900.0
	United States	7.3	12.4	17.90	15.8	49400.0	1.220	78.7	1.93	48400.0

167 rows × 9 columns

Pembagian kelompok cluster

```
# Penginisiasian Kluster berdasarkan negara yang membutuhkan, mungkin membutuhkan, dan tidak membutuhkan bantuan

cat = {0:'Need Help',1:'Might need help',2:'No Help needed'}
df_country['cluster_Id']=df_country['cluster_Id'].map(cat)
```

8

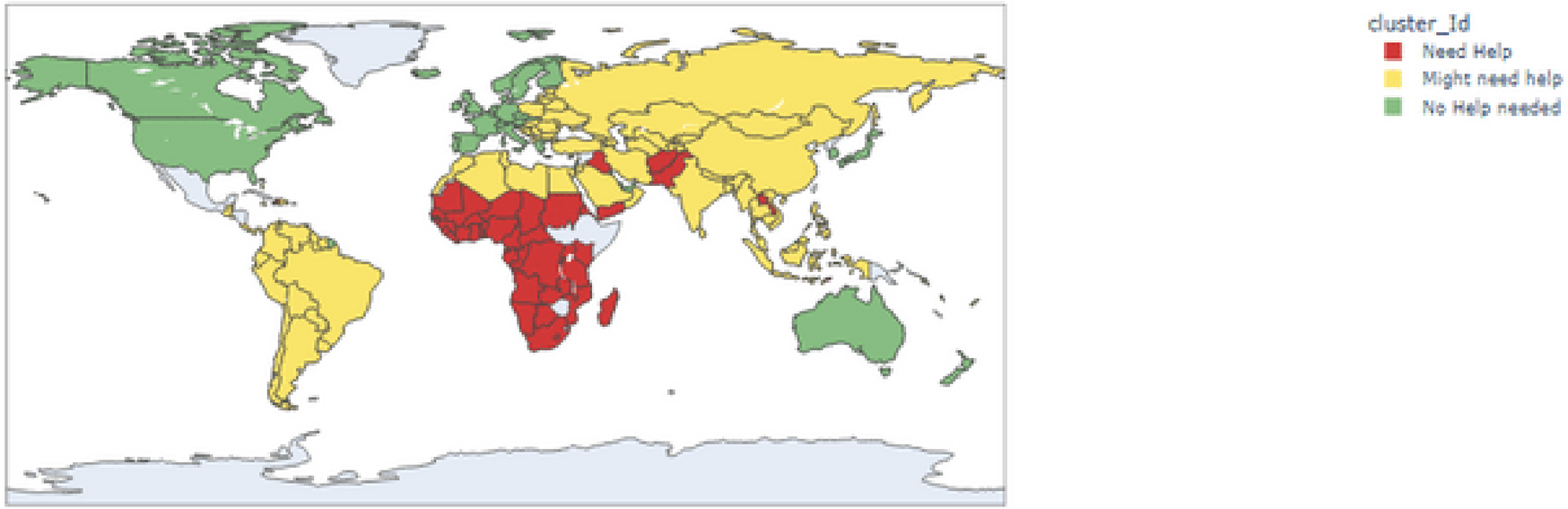
- Cluster 0 adalah kelompok negara yang membutuhkan bantuan
- Cluster 1 adalah kelompok negara yang mungkin membutuhkan bantuan
- Cluster 2 adalah kelompok negara yang tidak membutuhkan bantuan

Pembagian kelompok cluster

9

```
# Menampilkan peta dari negara yang membutuhkan, mungkin membutuhkan, dan tidak membutuhkan bantuan

px.choropleth(df_country, locationmode='country names', locations='country', color=df_country['cluster_Id'], title='Countries that need help',
              color_discrete_map={'Need Help':'#d13838','Might need help':'#fae56b','No Help needed':'#87bd82'} ,projection='equirectangular')
```

Countries that need help



List negara yang membutuhkan bantuan

```python
# List negara yang membutuhkan

need_help = df_country[df_country['cluster_Id']=='Need Help']['country']
need_help
```

```
0                  Afghanistan
3                       Angola
17                       Benin
21                    Botswana
25                Burkina Faso
26                     Burundi
28                    Cameroon
31     Central African Republic
32                        Chad
36                     Comoros
37             Congo, Dem. Rep.
38                  Congo, Rep.
40                Cote d'Ivoire
49            Equatorial Guinea
50                     Eritrea
55                       Gabon
56                      Gambia
59                       Ghana
63                      Guinea
64                Guinea-Bissau
66                       Haiti
72                        Iraq
80                       Kenya
81                    Kiribati
84                         Lao
87                     Lesotho
88                     Liberia
93                  Madagascar
94                      Malawi
97                        Mali
99                  Mauritania
106                 Mozambique
108                    Namibia
112                      Niger
113                    Nigeria
116                   Pakistan
126                     Rwanda
129                    Senegal
132               Sierra Leone
137               South Africa
142                      Sudan
147                   Tanzania
149                 Timor-Leste
150                       Togo
155                     Uganda
165                      Yemen
166                     Zambia
Name: country, dtype: object
```

List negara yang mungkin membutuhkan bantuan

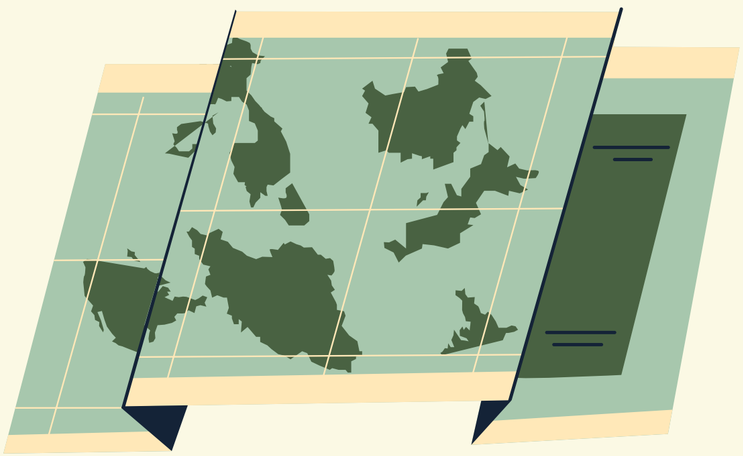
```python
# List negara yang mungkin membutuhkan

might_need_help = df_country[df_country['cluster_Id']=='Might need help']['country']
might_need_help
```

```
1                  Albania
2                  Algeria
4      Antigua and Barbuda
5                Argentina
6                  Armenia
               ...
160                Uruguay
161             Uzbekistan
162                Vanuatu
163              Venezuela
164                Vietnam
Name: country, Length: 84, dtype: object
```

List negara yang tidak membutuhkan bantuan

```python
# List negara yang tidak membutuhkan

no_need_help = df_country[df_country['cluster_Id']=='No Help needed']['country']
no_need_help
```

```
7                 Australia
8                   Austria
11                  Bahrain
15                  Belgium
23                   Brunei
29                   Canada
42                   Cyprus
43           Czech Republic
44                  Denmark
53                  Finland
54                   France
58                  Germany
60                   Greece
68                  Iceland
73                  Ireland
74                   Israel
75                    Italy
77                    Japan
82                   Kuwait
91               Luxembourg
98                    Malta
110            Netherlands
111            New Zealand
114                 Norway
122               Portugal
123                  Qatar
133              Singapore
134         Slovak Republic
135               Slovenia
138            South Korea
```



```
139                  Spain
144                 Sweden
145            Switzerland
157    United Arab Emirates
158         United Kingdom
159          United States
Name: country, dtype: object
```

Jumlah negara yang membutuhkan bantuan, mungkin membutuhkan bantuan, dan tidak membutuhkan bantuan

```
# Banyaknya negara yang tidak membutuhkan
no_need_help.count()
```

```
36
```

11

```
# Banyaknya negara yang mungkin membutuhkan
might_need_help.count()
```

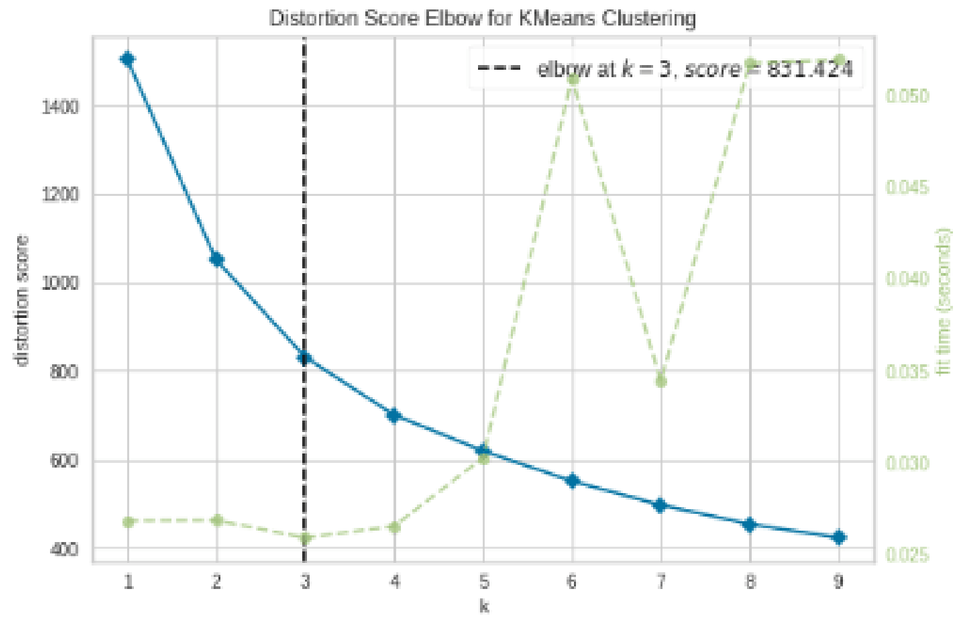
```
84
```

```
# Banyaknya negara yang membutuhkan
need_help.count()
```

```
47
```

EVALUASI

```python
# Evaluasi

model = KMeans()
visualize = KElbowVisualizer(model, k=(1,10))
visualize.fit(df_country_scaled)
visualize.poof()
```




```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe7620c4ca0>
```

Dari hasil pemrosesan data diatas dapat disimpulkan bahwa dengan menggunakan metode algoritma K-Means Clustering telah didapatkan :

- 47 Negara yang membutuhkan bantuan dari LSM
- 84 Negara yang mungkin membutuhkan bantuan dari LSM
- 36 Negara yang tidak membutuhkan bantuan dari LSM

TERIMAKASIH