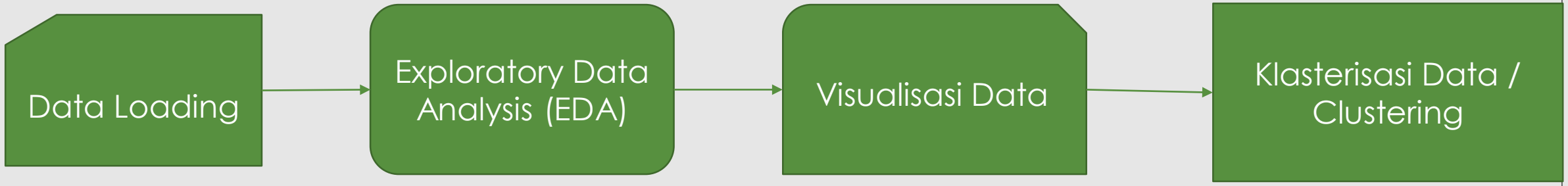
PROYEK AKHIR SANBERCODE

Aditya Aprianto

Alur project



Data Loading

- Import Library Pandas

  Import library pandas sangat penting untuk membaca data ke dalam dataframe.

- Load dataset

  Dataset diload menggunakan syntax pandas.read_csv(). Dataset yang digunakan adalah Data_Negara_HELP.csv.

Import Dataset

Hal pertama yang dilakukan adalah import dataset menggunakan library pandas. Lalu kita akan melihat datasetnya dengan memanggil variable data dan melihat 5 data pertama dengan memanggil head()

```
[1] import pandas as pd

    data = pd.read_csv("Data_Negara_HELP.csv")
    data
```

	Negara	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
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

Exploratory Data Analysis

- Deskripsi Variabel : Info Variabel dan Deskripsi Statistik
- Handling missing value
- Handling outlier
- Multivariate Analysis

Deskripsi Variabel

- **Negara** : Nama negara
- **Kematian_anak:** Kematian anak di bawah usia 5 tahun per 1000 kelahiran
- **Ekspor** : Ekspor barang dan jasa perkapita
- **Kesehatan** : Total pengeluaran kesehatan perkapita
- **Impor** : Impor barang dan jasa perkapita
- **Pendapatan** : Penghasilan bersih perorang
- **Inflasi** : Pengukuran tingkat pertumbuhan tahunan dari Total GDP
- **Harapan_hidup** : Jumlah tahun rata-rata seorang anak yang baru lahir akan hidup jika pola kematian saat ini tetap sama
- **Jumlah_fertiliti** : Jumlah anak yang akan lahir dari setiap wanita jika tingkat kesuburan usia saat ini tetap sama
- **GDPperkapita** : GDP per kapita. Dihitung sebagai Total GDP dibagi dengan total populasi.

Info Variabel

Mengecek info pada data dan melihat tipe data

Hal selanjutnya kita akan melihat info pada dataset untuk mengetahui tipe data. Terdapat satu tipe object yaitu kolom Negara, 7 tipe data float, dan 2 tipe data int yaitu Pendapatan dan GDPperkapita.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Negara            167 non-null    object 
 1   Kematian_anak     167 non-null    float64
 2   Ekspor            167 non-null    float64
 3   Kesehatan         167 non-null    float64
 4   Impor             167 non-null    float64
 5   Pendapatan        167 non-null    int64  
 6   Inflasi           167 non-null    float64
 7   Harapan_hidup     167 non-null    float64
 8   Jumlah_fertiliti  167 non-null    float64
 9   GDPperkapita      167 non-null    int64  
dtypes: float64(7), int64(2), object(1)
memory usage: 13.2+ KB
```

Deskripsi Statistik

```
data.describe()
```

	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
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

Handling Missing Value

- Menggunakan DataFrame.isnull().sum()
- Tidak ada nilai missing value

Mengecek nilai kosong atau missing value

Missing value adalah hilangnya beberapa data yang telah diperoleh. Dalam dunia data science, missing value erat kaitannya dalam proses perselisihan data (data wrangling) sebelum nantinya akan dilakukan analisis dan prediksi data.

Untuk mengecek nilai mising value kita dapat menggunakan syntax DataFrame.isnull(). Jika kita ingin mengetahui berapa total keseluruhan data yang hilang dapat ditambahkan .sum(). Karena tidak ada nilai missing value maka kita tidak perlu melakukan handling missing value.

```
data.isnull().sum()
```

```
Negara             0
Kematian_anak      0
Ekspor             0
Kesehatan          0
Impor              0
Pendapatan         0
Inflasi            0
Harapan_hidup      0
Jumlah_fertiliti   0
GDPperkapita       0
dtype: int64
```

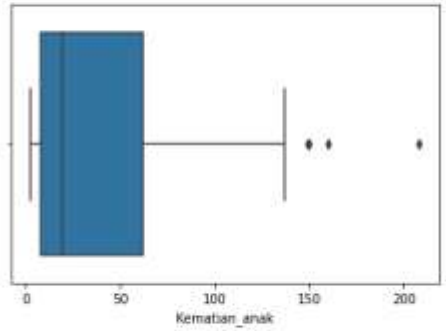
Handling Outlier

- Melihat outlier menggunakan seaborn boxplot
- Terdapat oulier pada masing-masing kolom
- Menghilangkan outlier menggunakan nilai interquartile (Batas kuartil atas dan kuartil bawah)
- Membuat data baru yang sudah dihilangkan outliernya

Melihat Outlier Menggunakan Seaborn Boxplot

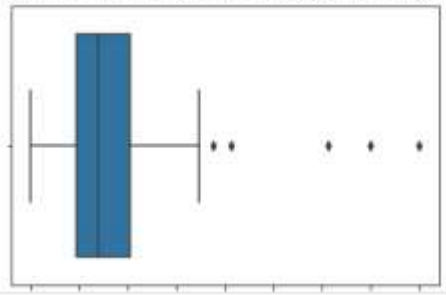
```
[5] import seaborn as sns
    import matplotlib.pyplot as plt

    sns.boxplot(x=data['Kematian_anak'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe83750d650>
```



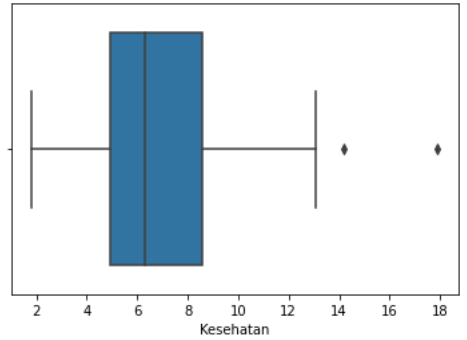

```
sns.boxplot(x=data['Ekspor'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe8292e6f90>
```



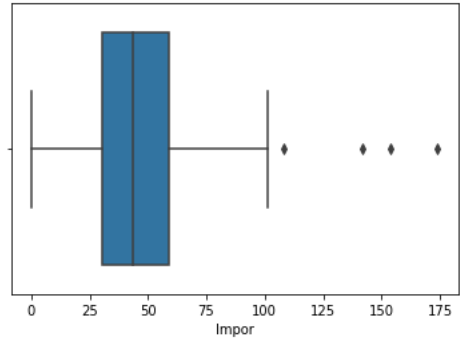
```
sns.boxplot(x=data['Kesehatan'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe828e1a610>
```




```
[8] sns.boxplot(x=data['Impor'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe828d95e50>
```



Menghilangkan Outlier

Dapat dilihat banyak kolom yang terdapat outliernya. Untuk menghilangkan outlier dapat menggantinya dengan nilai IQR, mean, median, dan modus. Disini kita akan mengganti nilai outlier dengan nilai IQR

```
[10] Q1 = data.quantile(0.25)
     Q3 = data.quantile(0.75)
     IQR=Q3-Q1
     data=data[~((data<(Q1-1.5*IQR))|(data>(Q3+1.5*IQR))).any(axis=1)]

     # Cek ukuran dataset setelah kita drop outliers
     data.shape
```

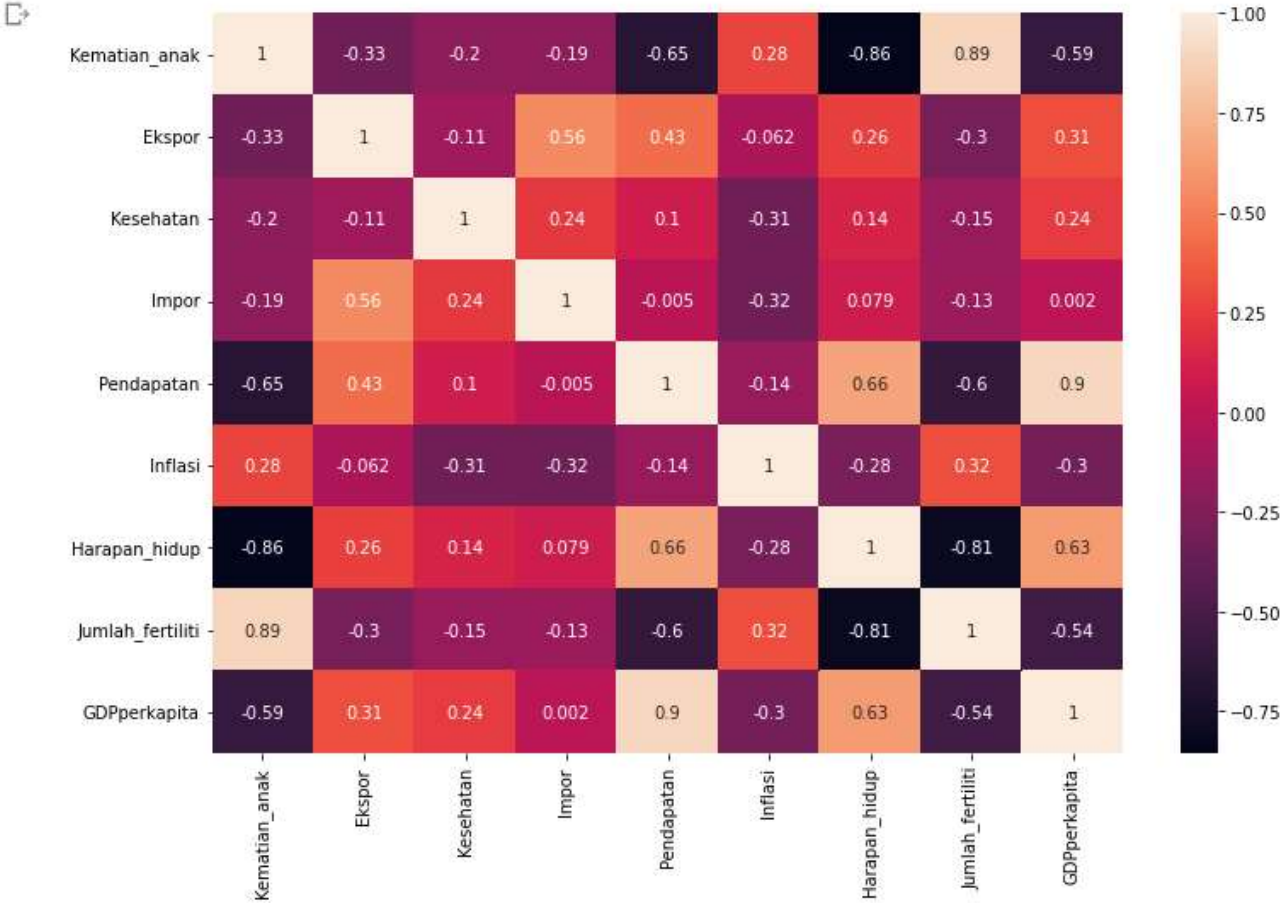
```
(128, 10)
```

Melakukan Multivariate Analysis

- Melihat adanya korelasi antar kolom data menggunakan heatmap (semakin mendekati +1 atau -1 berarti memiliki korelasi yang kuat)
- GDPperkapita dan Pendapatan memiliki korelasi yang paling besar dengan nilai positif
- Menaambil data GDPperkapita dan Pendapatan

Heatmap

```
plt.figure(figsize=(12,8))

sns.heatmap(data.corr(), annot=True, fmt='.2g');
```



Visualisasi Data

- Penggunaan Scatterplot dan regplot untuk mengetahui arah hubungan antar variabel
- Menggunakan distribution plot untuk mengetahui jenis distribusi suatu variabel
- Penggunaan barplot untuk melihat data negara yang memiliki GDP dan pendapatan tertinggi dan terendah

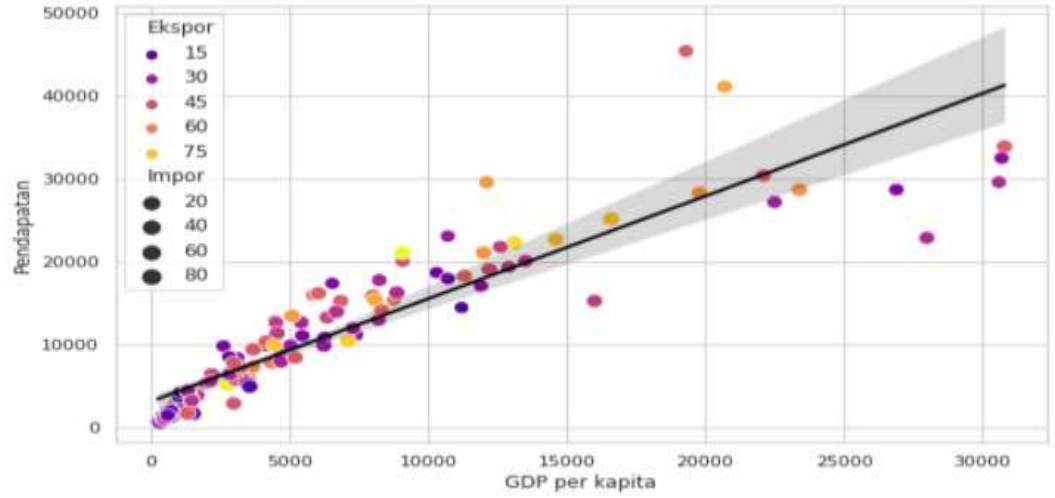
Scatterplot Dan Regplot

Korelasi antara GDPperkapita dan Pendapatan

Dengan menggunakan scatterplot dan regplot kita dapat melihat hubungan kedua kolom tersebut. Dari hasil gambar dapat disimpulkan bahwa GDP perkapita dan pendapatan memiliki hubungan yang positif jika dilihat dari arah regresinya. Artinya peningkatan yang terjadi pada variabel GDPperkapita juga diikuti peningkatan pada variabel Pendapatan. Dan sebaliknya, jika variabel GDP perkapita mengalami penurunan maka variabel Pendapatan juga mengalami penurunan

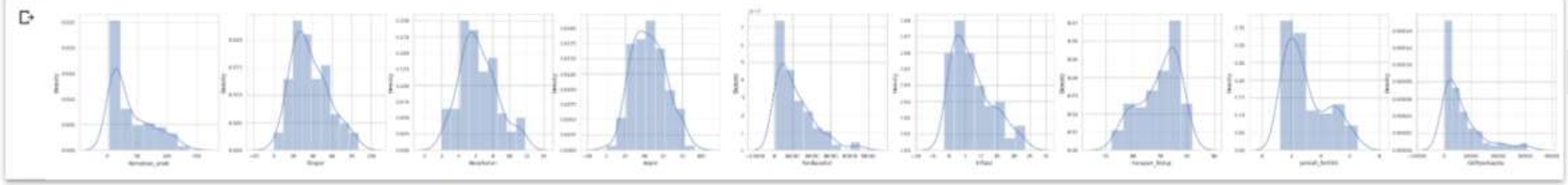
```
sns.set_theme(style='whitegrid')
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='GDPperkapita', y='Pendapatan', hue='Ekspor', size='Impor', sizes=(80, 120), palette='plasma')
sns.regplot(x="GDPperkapita", y="Pendapatan", data=data, scatter=False, color='black')
plt.xlabel('GDP per kapita')
plt.ylabel('Pendapatan')
```

```
Text(0, 0.5, 'Pendapatan')
```



Melihat distribusi data menggunakan distplot seaborn

```
import warnings
warnings.filterwarnings('ignore')

plt.figure(figsize=(60, 60))

for i in enumerate(data.describe().columns):
  plt.subplot(9,9, i[0]+1)
  sns.distplot(data[i[1]])
plt.show()
```



```
GDP = data.sort_values('GDPperkapita', ascending=False)
```

- Kolom Ekspor, Kesehatan, Impor memiliki Distribusi normal
- Kolom Kematian_anak, pendapatan, Inflasi, Jumlah_fertili, dan GDPPerkapita memiliki kemiringan positif
- Kolom Harapan_hidup memiliki kemiringan negatif

Sortir Data By GDP

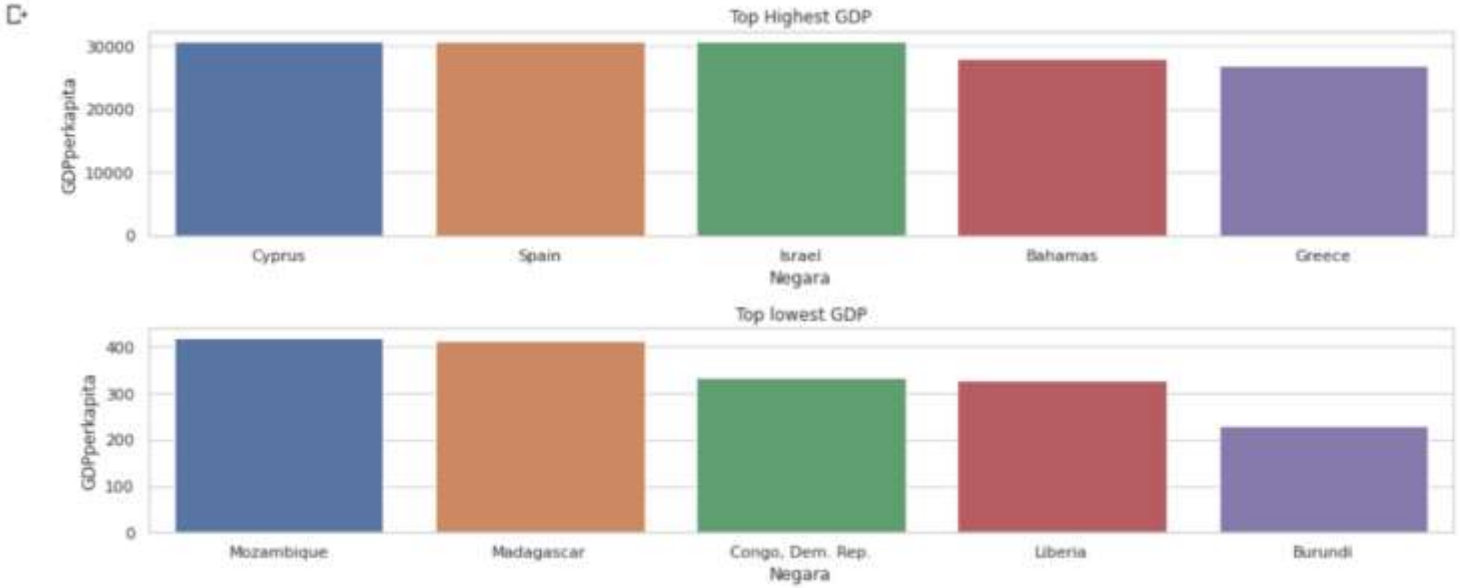
```
[14] GDP = data.sort_values('GDPperkapita', ascending=False)
     GDP
```

	Negara	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
42	Cyprus	3.6	50.20	5.97	57.5	33900	2.010	79.9	1.42	30800
139	Spain	3.8	25.50	9.54	26.8	32500	0.160	81.9	1.37	30700
74	Israel	4.6	35.00	7.63	32.9	29600	1.770	81.4	3.03	30600
10	Bahamas	13.8	35.00	7.89	43.7	22900	-0.393	73.8	1.86	28000
60	Greece	3.9	22.10	10.30	30.7	28700	0.673	80.4	1.48	26900
...	...	...	...	...	...	...	...	...	...	...
106	Mozambique	101.0	31.50	5.21	46.2	918	7.640	54.5	5.56	419
93	Madagascar	62.2	25.00	3.77	43.0	1390	8.790	60.8	4.60	413
37	Congo, Dem. Rep.	116.0	41.10	7.91	49.6	609	20.800	57.5	6.54	334
88	Liberia	89.3	19.10	11.80	92.6	700	5.470	60.8	5.02	327
26	Burundi	93.6	8.92	11.60	39.2	764	12.300	57.7	6.26	231

128 rows × 10 columns

Barplot berdasarkan ‘Negara’ dan ‘GDPperkapita’

```
plt.figure(figsize=(14,6))
plt.subplot(2,1,1)
sns.barplot(GDP.Negara.head(), GDP.GDPperkapita.head())
plt.title('Top Highest GDP')
plt.subplot(2,1,2)
sns.barplot(GDP.Negara.tail(), GDP.GDPperkapita.tail())
plt.title('Top lowest GDP')
plt.tight_layout()
plt.show()
```



Sorti Data By Income

Mozambique Madagascar Congo, Dem. Rep. Liberia Burundi

Negara

```
Income = data.sort_values('Pendapatan', ascending=False)
Income
```

	Negara	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
128	Saudi Arabia	15.7	49.60	4.29	33.0	45400	17.20	75.1	2.96	19300
115	Oman	11.7	65.70	2.77	41.2	45300	15.60	76.1	2.90	19300
11	Bahrain	8.6	69.50	4.97	50.9	41100	7.44	76.0	2.16	20700
42	Cyprus	3.6	50.20	5.97	57.5	33900	2.01	79.9	1.42	30800
139	Spain	3.8	25.50	9.54	26.8	32500	0.16	81.9	1.37	30700
...	...	...	...	...	...	...	...	...	...	...
94	Malawi	90.5	22.80	6.59	34.9	1030	12.10	53.1	5.31	459
106	Mozambique	101.0	31.50	5.21	46.2	918	7.64	54.5	5.56	419
26	Burundi	93.6	8.92	11.60	39.2	764	12.30	57.7	6.26	231
88	Liberia	89.3	19.10	11.80	92.6	700	5.47	60.8	5.02	327
37	Congo, Dem. Rep.	116.0	41.10	7.91	49.6	609	20.80	57.5	6.54	334

128 rows × 10 columns

Barplot Berdasarkan ‘Negara’ dan Pendapatan’

```
plt.figure(figsize=(14,6))
plt.subplot(2,1,1)
sns.barplot(Income.Negara.head(), Income.Pendapatan.head())
plt.title('Top Highest Income')
plt.subplot(2,1,2)
sns.barplot(Income.Negara.tail(), Income.Pendapatan.tail())
plt.title('Top lowest Income')
plt.tight_layout()
plt.show()
```

Kesimpulan Barplot

- Top lowest GDPperkapita yaitu negara Mozambique, Madagascar, Congo, Liberia, dan Burundi
- Top lowest Pendapatan yaitu negara Malawi, Mozambique, Burundi, Liberia, dan Congo
- CEO LSM dapat memilih negara Mozambique, Burundi, atau Liberia sebagai pilihan untuk menyumbangkan dana bantuan berdasarkan GDP dan pendapatan dari negara tersebut.

Standarisasi Data

```
[19] from sklearn import preprocessing
     from sklearn.preprocessing import LabelEncoder

     # Standarisasi menggunakan StandardScaler
     scaler = preprocessing.StandardScaler().fit(new_data)
     new_data = scaler.transform(new_data)
     new_data
```

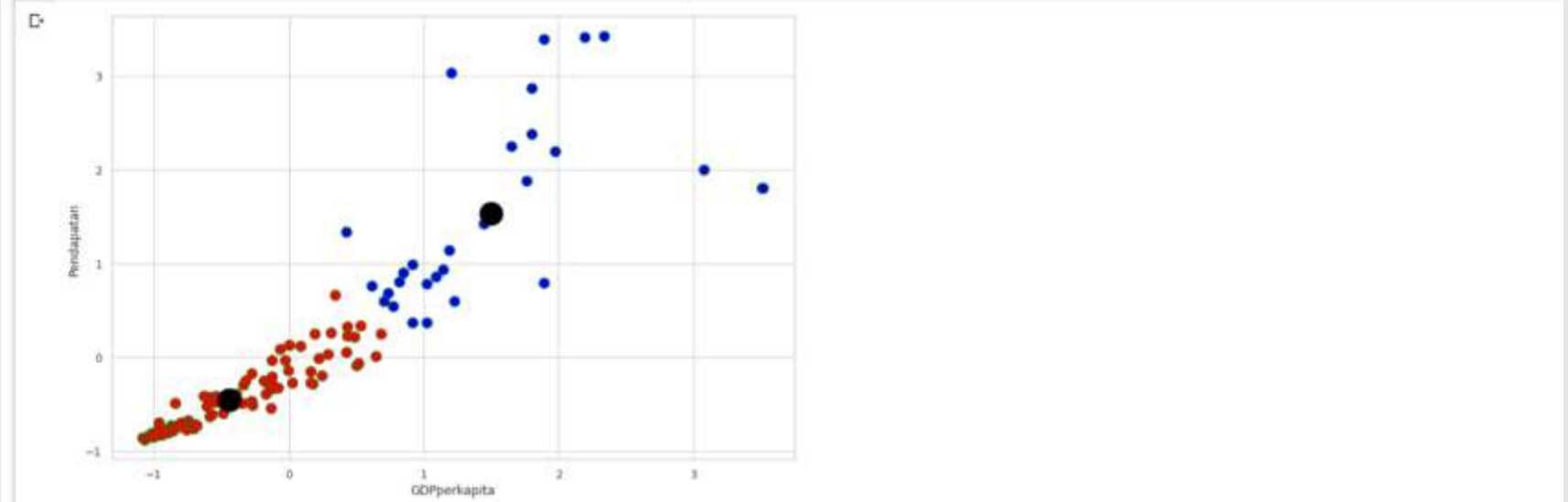
```
array([[-0.97979661, -0.82556137],
       [-0.12528564, -0.32838491],
       [ 0.17975012, -0.27637607],
       [-0.53918939, -0.407101  ],
       [ 0.81652513,  0.81159278],
       [ 0.77544287,  0.54452032],
       [-0.45702488, -0.45067598],
       [ 0.49813763, -0.08239712],
       [ 1.20680658,  3.0325111 ],
       [ 3.07604932,  2.00639061],
       [-0.89455093, -0.79674566],
       [ 0.42624367,  1.34573769],
       [ 0.51867875, -0.05568988],
       [-0.33583221, -0.2932438 ],
```

Clustering

- Menggunakan klasterisasi menggunakan K-Means
- Menggunakan Elbow Methods untuk mencari nilai cluster yang optimal.

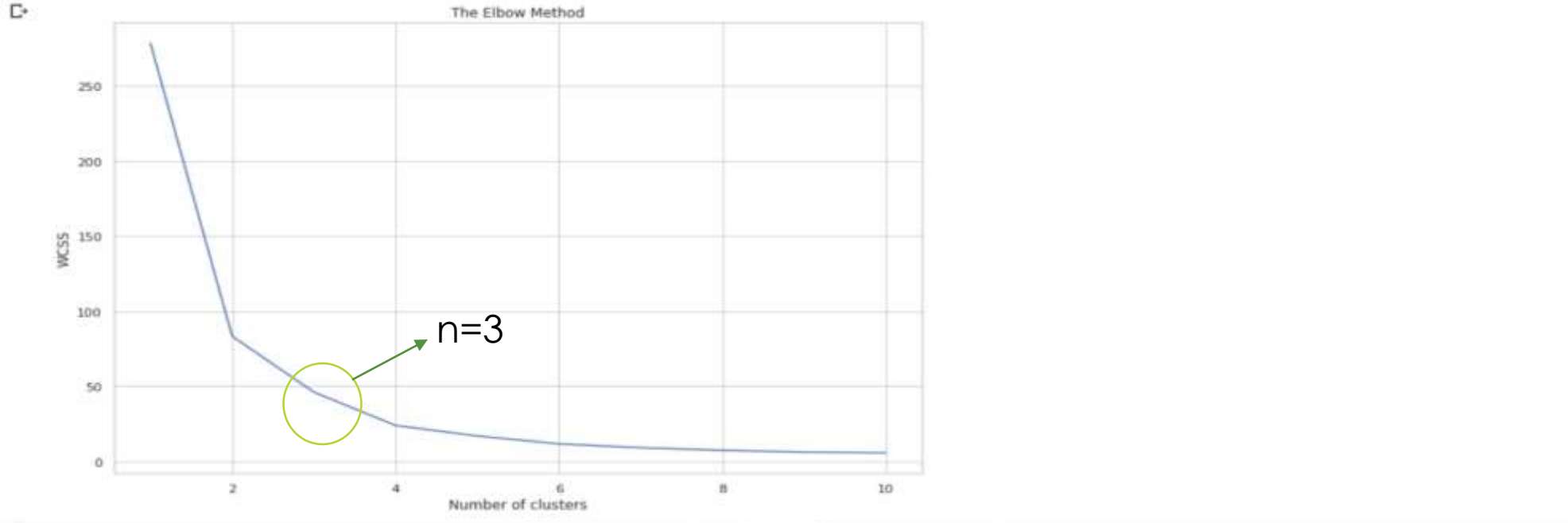
Clustering menggunakan 2 cluster

```
#Memvisualisasikan Hasil Clustering menggunakan scatterplot
plt.figure(figsize=(12,8))
plt.scatter(data_std.GDPperkapita[data_std.label_kmeans1 == 0], data_std['Pendapatan'][data_std.label_kmeans1 == 0], c='blue', s=100, edgecolor='green',linestyle='-')
plt.scatter(data_std.GDPperkapita[data_std.label_kmeans1 == 1], data_std['Pendapatan'][data_std.label_kmeans1 == 1], c='red', s=100, edgecolor='green',linestyle='-')

centers = kmeans1.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=500);
plt.xlabel('GDPperkapita')
plt.ylabel('Pendapatan')
plt.show()
```



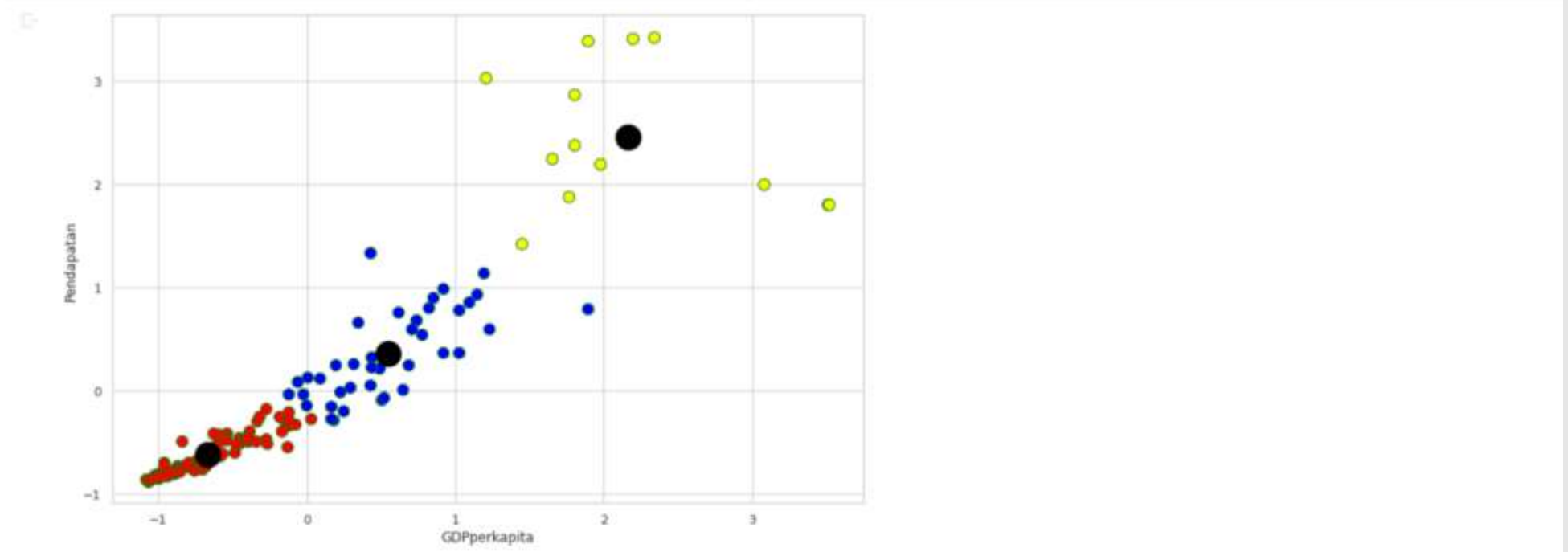
Mencari Nilai Optimum Menggunakan Elbow Methods

```
wcss = []
for i in range(1, 11):
  kmeans = KMeans(n_clusters= i, init='k-means++', random_state = 42 )
  kmeans.fit(data_std)
  wcss.append(kmeans.inertia_)

plt.figure(figsize= (12, 8))
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



Clustering menggunakan 3 cluster

```
[24] #Memvisualisasikan hasil clustering
     plt.figure(figsize=(12,8))
     plt.scatter(data_std.GDPperkapita[data_std.label_kmeans2 == 0], data_std['Pendapatan'][data_std.label_kmeans2 == 0], c='yellow', s=100, edgecolor='green',linestyle='-')
     plt.scatter(data_std.GDPperkapita[data_std.label_kmeans2 == 1], data_std['Pendapatan'][data_std.label_kmeans2 == 1], c='red', s=100, edgecolor='green',linestyle='-')
     plt.scatter(data_std.GDPperkapita[data_std.label_kmeans2 == 2], data_std['Pendapatan'][data_std.label_kmeans2 == 2], c='blue', s=100, edgecolor='green',linestyle='-')

     centers = kmeans2.cluster_centers_
     plt.scatter(centers[:, 0], centers[:, 1], c='black', s=500);
     plt.xlabel('GDPperkapita')
     plt.ylabel('Pendapatan')
     plt.show()
```



Title Lorem Ipsum

LOREM IPSUM DOLOR SIT AMET, CONSECTETUER ADIPISCING ELIT.

NUNC VIVERRA IMPERDIET ENIM. FUSCE EST. VIVAMUS A TELLUS.

PELLENTESQUE HABITANT MORBI TRISTIQUE SENECTUS ET NETUS.