

Nama: Muhammad Abdulloh Hamzan

EMAIL: abdullah.hamzan@gmail.com

Tugas: Final Project

Permasalahan:

HELP International telah berhasil mengumpulkan sekitar \$ 10 juta. Saat ini, CEO LSM perlu memutuskan bagaimana menggunakan uang ini secara strategis dan efektif. Jadi, CEO harus mengambil keputusan untuk memilih negara yang paling membutuhkan bantuan. Oleh karena itu, Tugas teman-teman adalah mengkategorikan negara menggunakan beberapa faktor sosial ekonomi dan kesehatan yang menentukan perkembangan negara secara keseluruhan. Kemudian kalian perlu menyarankan negara mana saja yang paling perlu menjadi fokus CEO.

Jawab:

1. Langkah pertama yang di lakukan yaitu menyiapkan sebuah data. Dimana pada final Project ini data yang digunakan yaitu Data_Negara_HELP.csv setelah data sudah tersedia langkah selanjutnya yaitu proses pembacaan dari data tersebut. Pada proses pembacaan data menggunakan library pandas. Setelah berhasil melakukan pembacaan data maka akan muncul seperti gambar di bawah ini

```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

df=pd.read_csv('Data_Negara_HELP.csv')
df
```

	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	50.9	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 x 10 columns

Gambar 1 hasil pembacaan data

Dari hasil pembacaan data ini dimana terdapat 167 baris dan terdapat 10 kolom dimana pada kolom tersebut terdapat Kolom Negara, Kematian anak,Ekspor,Kesehatan,Impor,Pendapatan,Inflasi,Harapan hidup,Jumah fertility dan GPD perkapita.

2. Langkah kedua yaitu memberikan informasi yang lebih spesifik tentang data dengan menggunakan method info dari data frame. Hasil yang di dapatakan dapat dilihat pada gambar dibawah ini:

```
[3] ▶ ▶ MI
df.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
```

Gambar 2 informasi data

Pada gambar 2 didapatkan informasi yaitu tentang type data yang digunakan. Pada kolom Negara menggunakan type data object, kematian anak menggunakan type data float dan seterusnya dapat dilihat pada gambar 2.

3. Langkah ketiga yaitu memberikan informasi yang lebih spesifik tentang data menggunakan method describe dari data frame. Hasil yang didapatkan dapat dilihat pada gambar dibawah ini:

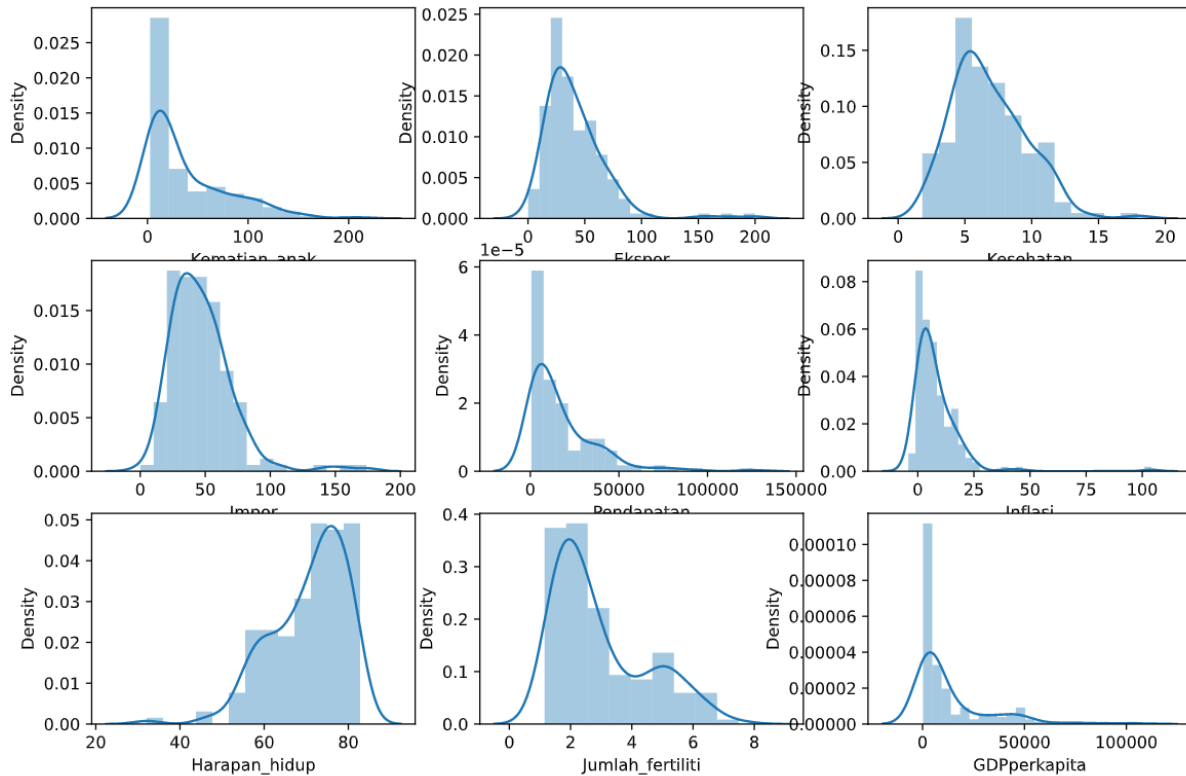
```
df.describe()
```

	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

Gambar 3 informasi describe data

Pada gambar 3 didapatkan informasi berupa jumlah, nilai rata-rata, standar deviasi, nilai minimum, nilai maksimum, kuartil 1 (25%), kuartil 2 (50%) atau median dan kuartil 3 (75%) pada setiap columnnya. Untuk nilai-nilainya dapat dilihat pada gambar 3.

4. Langkah keempat yaitu melakukan Univariate analisis. Dimana pada Analisa ini akan mengetahui hubungan setiap kolom dengan density atau kerapatannya menggunakan method distplot dari seaborn. Hasil yang didapatkan dapat dilihat pada gambar dibawah ini:



Gambar 4 hasil distplot

Pada gambar 4 didapatkan hasil hubungan Density(kerapatan) dengan setiap kolom pada data sebagai berikut:

1. Hubungan density dengan kematian anak didapatkan yang paling tinggi sekitar 30an tingkat kematian anak. Untuk lebih jelasnya dapat dilihat pada gambar 4 baris 1 kolom 1.
2. Hubungan density dengan Ekspor didapatkan yang paling tinggi sekitar 25 tingkat ekspor. Untuk lebih jelasnya dapat dilihat pada gambar 4 baris 1 kolom 2
3. Hubungan density dengan Kesehatan didapatkan yang paling tinggi sekitar 5 tingkat kesehatan. Untuk lebih jelasnya dapat dilihat pada gambar 4 baris 1 kolom 3
4. Hubungan density dengan Impor didapatkan yang paling tinggi sekitar 40an tingkat impor. Untuk lebih jelasnya dapat dilihat pada gambar 4 baris 2 kolom 1
5. Hubungan density dengan Pendapatan didapatkan yang paling tinggi sekitar 20.000an tingkat pendapatan. Untuk lebih jelasnya dapat dilihat pada gambar 4 baris 2 kolom 2

6. Hubungan density dengan Inflasi didapatkan yang paling tinggi sekitar 8an tingkat inflasi. Untuk lebih jelasnya dapat dilihat pada gambar 4 baris 2 kolom 3
 7. Hubungan density dengan Harapan Hidup didapatkan yang paling tinggi sekitar 70an sampai 80an tingkat Harapan hidup. Untuk lebih jelasnya dapat dilihat pada gambar 4 baris 3 kolom 1
 8. Hubungan density dengan Jumlah fertiliti didapatkan yang paling tinggi sekitar 2 tingkat Jumlah fertiliti. Untuk lebih jelasnya dapat dilihat pada gambar 4 baris 3 kolom 2
 9. Hubungan density dengan GDPperkapita didapatkan yang paling tinggi sekitar 10.000an tingkat GDP perkapita. Untuk lebih jelasnya dapat dilihat pada gambar 4 baris 3 kolom 3
5. Langkah kelima mengurutkan baris pada tabel berdasarkan GPD perkapita dengan mengurutkannya di mulai dari GPD terbesar sampai terendah. Dapat dilihat pada gambar dibawah ini :

```
#mengurutkan nilai berdasarkan tingkat GDP Perkapita
gdp_perkapita= df.sort_values('GDPperkapita', ascending=False)
gdp_perkapita
```

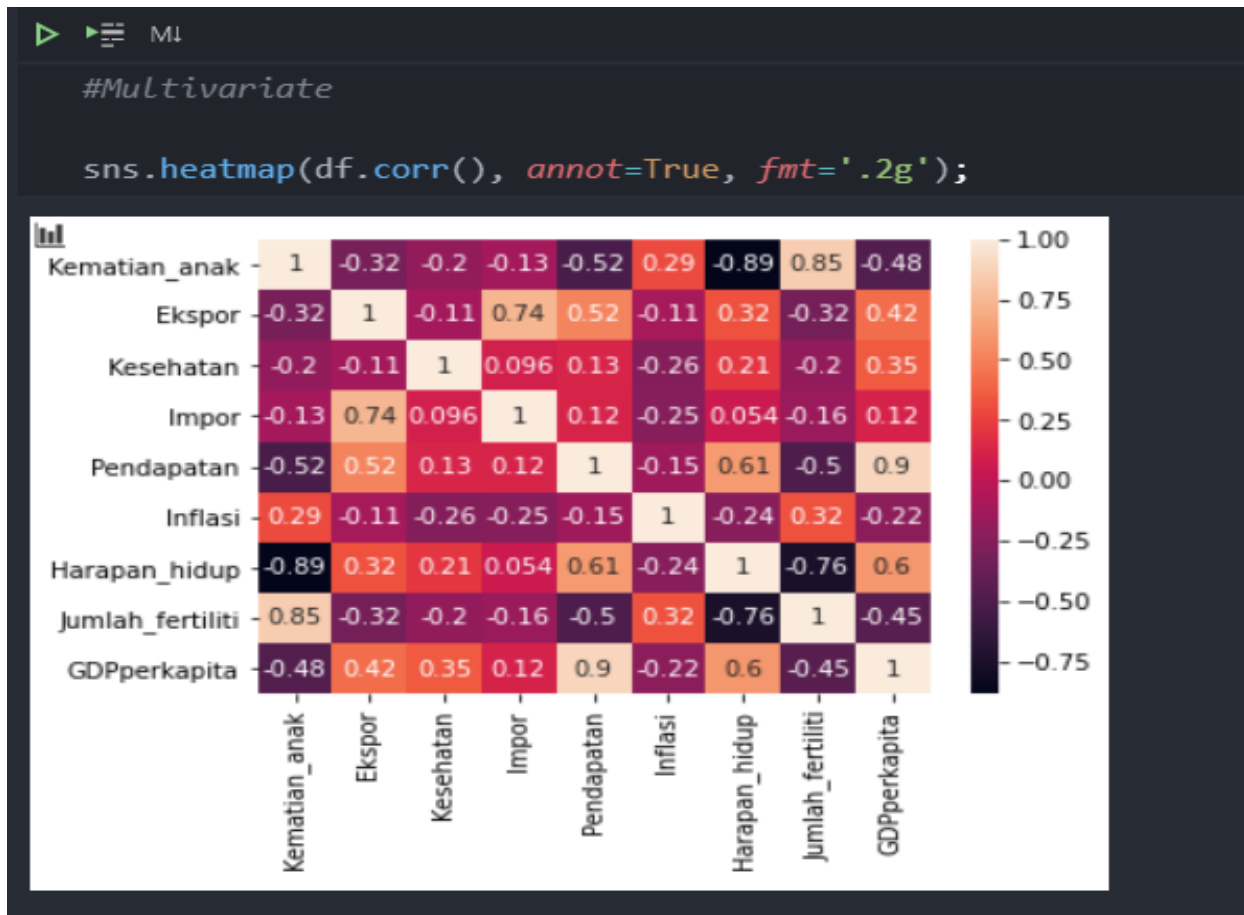
	Negara	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
91	Luxembourg	2.8	175.00	7.77	142.0	91700	3.620	81.3	1.63	105000
114	Norway	3.2	39.70	9.48	28.5	62300	5.950	81.0	1.95	87800
145	Switzerland	4.5	64.00	11.50	53.3	55500	0.317	82.2	1.52	74600
123	Qatar	9.0	62.30	1.81	23.8	125000	6.980	79.5	2.07	70300
44	Denmark	4.1	50.50	11.40	43.6	44000	3.220	79.5	1.87	58000
...
132	Sierra Leone	160.0	16.80	13.10	34.5	1220	17.200	55.0	5.20	399
112	Niger	123.0	22.20	5.16	49.1	814	2.550	58.8	7.49	348
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

167 rows x 10 columns

Gambar 5 Data setelah di urutkan berdasarkan perkapita

Berdasarkan gambar 5 didapatkan urutan Negara besarkan nilai dari GPD perkapitanya. Dari hasil pengurutan berdasarkan GPD perkapita didapatkan nilai GPD perkapita terendah diperoleh oleh Negara Burundi dengan nilai GPD perkapitanya 231.

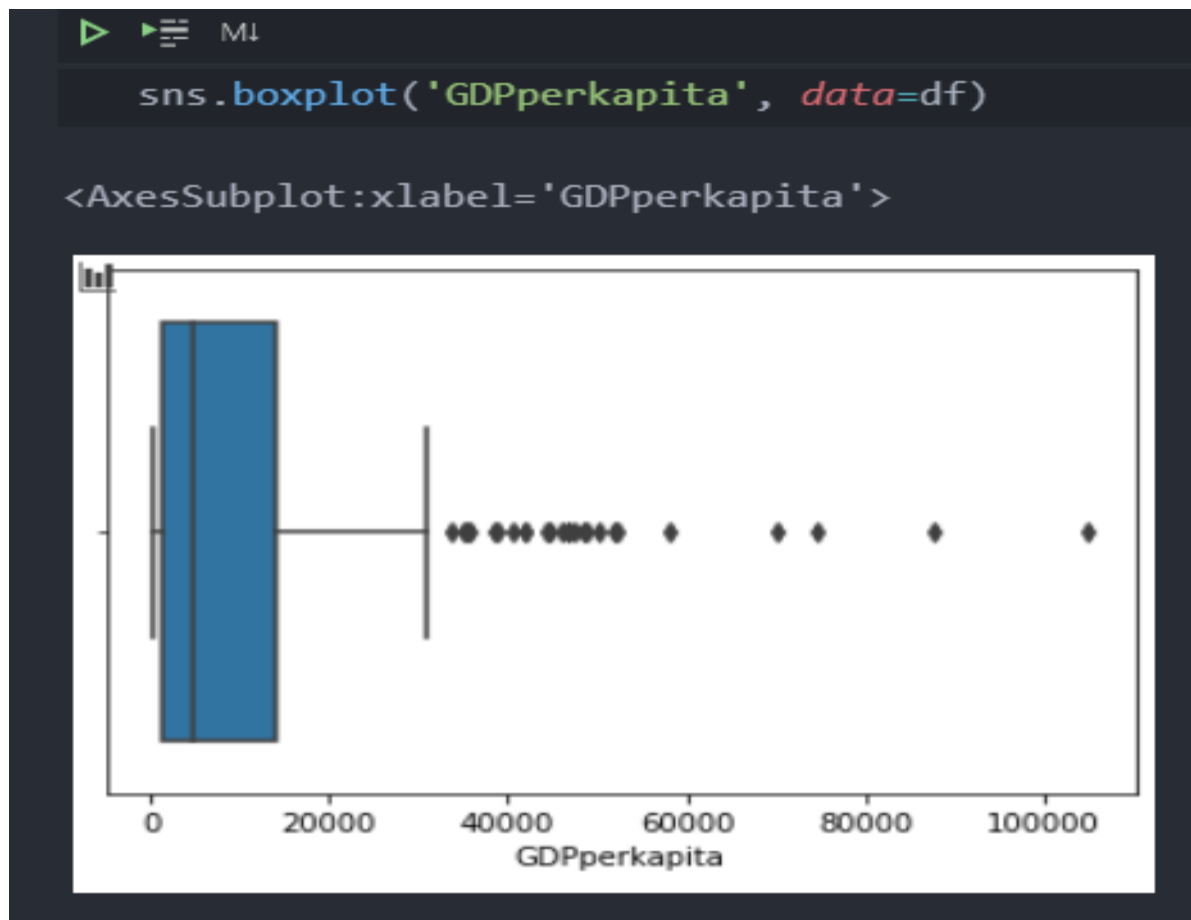
6. Analisa multivariate dengan melihat korelasi data menggunakan seaborn dengan method heatmap. Sehingga di dapatkan hasil dapat dilihat pada gambar dibawah ini:



Gambar 6 hasil plotting korelasi data

Pada gambar 6 menunjukkan hasil korelasi datanya dimana korelasi yang tinggi yaitu antran kematian anak dengan harapan hidupnya, Ekspor dengan Impor, Pendapatan dengan GDP perkapita, harapan hidup dengan kematian anak , harapan hidup dengan jumlah fertility dan Jumlah fertiliti dengan kematian anak. Dari data tersebut untuk korelasi tinggi di tandai dengan mendekati -1 dan 1 sedangkan ketika tidak memiliki korelasi data ditandai dengan 0.

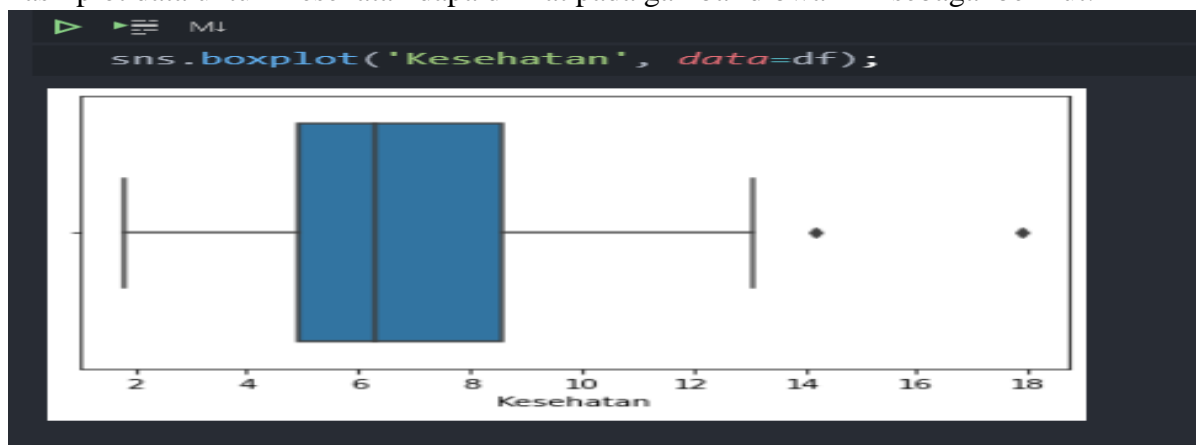
7. Melihat data outlier dengan menggunakan library seaborn dengan method boxplot untuk mendapatkan hasil plot data untuk GDP perkapita dapat di lihat pada gambar dibawah ini sebagai berikut:



Gambar 7 hasil boxplot data GDP perkapita

Berdasarkan gambar 7 terdapat nilai penciran atau outlier. Nilai outlier pada gambar 7 berpengaruh juga terhadap data sehingga perlu untuk di drop atau di hapus dari data supaya tidak mengganggu data lainnya pada GDP perkapita nya

8. Melihat data outlier dengan menggunakan library seaborn dengan method boxplot untuk mendapatkan hasil plot data untuk Kesehatan dapat dilihat pada gambar dibawah ini sebagai berikut:



Gambar 8 hasil boxplot data Kesehatan

Berdasarkan gambar 8 terdapat 2 data pencilan atau outlier. Nilai outlier pada gambar 8 berpengaruh juga terhadap data sehingga perlu di drop atau di hapus dari data supaya tidak mengganggu data lainnya pada data Kesehatan.

9. Langkah Sembilan yaitu membuat sebuah fungsi yang dapat memfiltering data outliernya. Untuk hasil filteringgnya dapat dilihat pada gambar di bawah ini:

```
[10] ▶ ML

def finding_outlier(df):
    q1=df.quantile(0.25)
    q3=df.quantile(0.75)
    iqr=q3-q1
    upper_bound=q3+(1.5*iqr)
    lower_bound=q1-(1.5*iqr)
    df_final=df[(df < upper_bound) & (df > lower_bound)]
    return df_final
finding_outlier(df[['Kesehatan','GDPperkapita']])
```

	Kesehatan	GDPperkapita
0	7.58	553.0
1	6.55	4090.0
2	4.17	4460.0
3	2.85	3530.0
4	6.03	12200.0
...
162	5.25	2970.0
163	4.91	13500.0
164	6.84	1310.0
165	5.18	1310.0
166	5.89	1460.0

167 rows × 2 columns

Gambar 9 hasil filtering data dengan tidak mengambil data outliernya atau pencilannya

Pada gambar diatas dimana sebuah fungsi dengan nama fungsi finding_outlier yang digunakan untuk filtering data dengan tidak mengambil nilai outliernya. Pada fungsi diatas data yang di filtering yaitu data untuk kolom Kesehatan dan kolom GDPperkapita hasil filteringgnya dapat dilihat pada gambar 9.

10. Langkah 10 yaitu membuat sebuah fungsi yang dapat menghapus atau membuang nilai outlier yang terdapat pada kolom Kesehatan dan GDP perkapita. Untuk hasil data yang sudah di drop dapat dilihat pada gambar dibawah ini:

```
[14] ▶ ▶≡ MI
def remove_outlier(df):
    q1=df.quantile(0.25)
    q3=df.quantile(0.75)
    iqr=q3-q1
    upper_bound=q3+(1.5*iqr)
    lower_bound=q1-(1.5*iqr)
    df_final=df[(df < upper_bound) & (df > lower_bound)]
    return df_final

df2=remove_outlier(df[['Kesehatan','GDPperkapita']])
df2.dropna(axis=0, inplace=True)
```

```
[15] ▶ ▶≡ MI
df2
```

	Kesehatan	GDPperkapita
0	7.58	553.0
1	6.55	4090.0
2	4.17	4460.0
3	2.85	3530.0
4	6.03	12200.0
...
162	5.25	2970.0
163	4.91	13500.0
164	6.84	1310.0
165	5.18	1310.0
166	5.89	1460.0

141 rows × 2 columns

Gambar 10 hasil drop data untuk kolom Kesehatan dan GdP perkapita

Pada gambar 10 dimana terlihat hasil perubahan datanya setelah dilakukan dropna datanya, yang semula jumlah barisnya ada 167 menjadi 141 setelah di lakukan drop data outlier nya.

11. Menampilkan isi data dengan kolom Kesehatan dan kolom GDP perkapita dapat dilihat pada gambar di bawah ini:

```
[19] ▶ ML
df2['GDPperkapita'].unique()

array([ 553., 4090., 4460., 3530., 12200., 10300., 3220., 5840.,
        28000., 20700., 758., 16000., 6030., 4340., 2180., 1980.,
        4610., 6350., 11200., 6840., 575., 231., 786., 1310.,
        3310., 446., 897., 12900., 4560., 6250., 769., 334.,
        2740., 8200., 1220., 13500., 30800., 19800., 5450., 4660.,
        2600., 2990., 17100., 482., 14600., 3650., 8750., 562.,
        2960., 26900., 7370., 2830., 648., 547., 3040., 662.,
        13100., 1350., 3110., 6530., 4500., 30600., 4680., 3680.,
        9070., 967., 1490., 880., 1140., 11300., 8860., 1170.,
        327., 12100., 12000., 4540., 413., 459., 7100., 708.,
        21100., 1200., 8000., 1630., 2650., 6680., 419., 988.,
        5190., 592., 348., 2330., 19300., 1040., 8080., 3230.,
        5020., 2130., 12600., 22500., 8230., 10700., 563., 3450.,
        1000., 5410., 10800., 399., 16600., 23400., 1290., 7280.,
        22100., 30700., 2810., 6230., 1480., 8300., 738., 702.,
        5080., 3600., 488., 3550., 4140., 4440., 595., 2970.,
        11900., 1380., 1460.])

[20] ▶ ML
df2['Kesehatan'].unique()

array([ 7.58, 6.55, 4.17, 2.85, 6.03, 8.1 , 4.4 , 5.88, 7.89,
        4.97, 3.52, 7.97, 5.61, 5.2 , 4.1 , 4.84, 11.1 , 8.3 ,
        9.01, 6.87, 6.74, 11.6 , 5.68, 5.13, 4.09, 3.98, 4.53,
        7.96, 5.07, 7.59, 4.51, 7.91, 2.46, 10.9 , 5.3 , 7.76,
        5.97, 7.88, 6.22, 8.06, 4.66, 6.91, 4.48, 2.66, 4.86,
        3.5 , 5.69, 10.1 , 5.22, 10.3 , 5.86, 6.85, 4.93, 8.5 ,
        5.38, 7.33, 4.05, 2.61, 5.6 , 8.41, 7.63, 4.81, 8.04,
        4.29, 4.75, 11.3 , 6.18, 4.47, 6.68, 7.03, 11.8 , 3.88,
        7.04, 7.09, 3.77, 6.59, 4.39, 6.33, 4.98, 8.65, 4.41,
        6. , 11.7 , 5.44, 9.11, 5.21, 1.97, 6.78, 5.25, 5.16,
        2.77, 2.2 , 5.87, 5.08, 3.61, 7.46, 11. , 5.58, 10.5 ,
        6.47, 5.66, 10.4 , 3.4 , 13.1 , 8.79, 9.41, 8.55, 8.94,
        6.93, 9.54, 2.94, 6.32, 7.01, 5.98, 6.01, 9.12, 7.65,
        6.21, 2.5 , 7.72, 8.35, 5.81, 4.91, 6.84, 5.18, 5.89])
```

Gambar 11 Menampilkan nilai data dengan Kolom GDPperkapita dan Kesehatan

Pada gambar 11 dapat dilihat isi data semua kolom untuk Kesehatan dan semua kolom untuk GDP per kapitanya pada gambar diatas.

12. Melakukan Scaling data menggunakan library sklearn.preprocessing dengan mengimport StandarScaler yang digunakan untuk menscaling data. Setelah berhasil menscaling data langkah selanjutnya yaitu mengclustering data dengan KMeans. Kmeans itu method yang diambil dari sklearn.cluster yang digunakan untuk mengclustering data. Dimana hasil clustering data dapat dilihat pada gambar dibawah ini:

```
[21] ▶ MI
#feature scaling
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
df_std =sc.fit_transform(df2.astype(float))

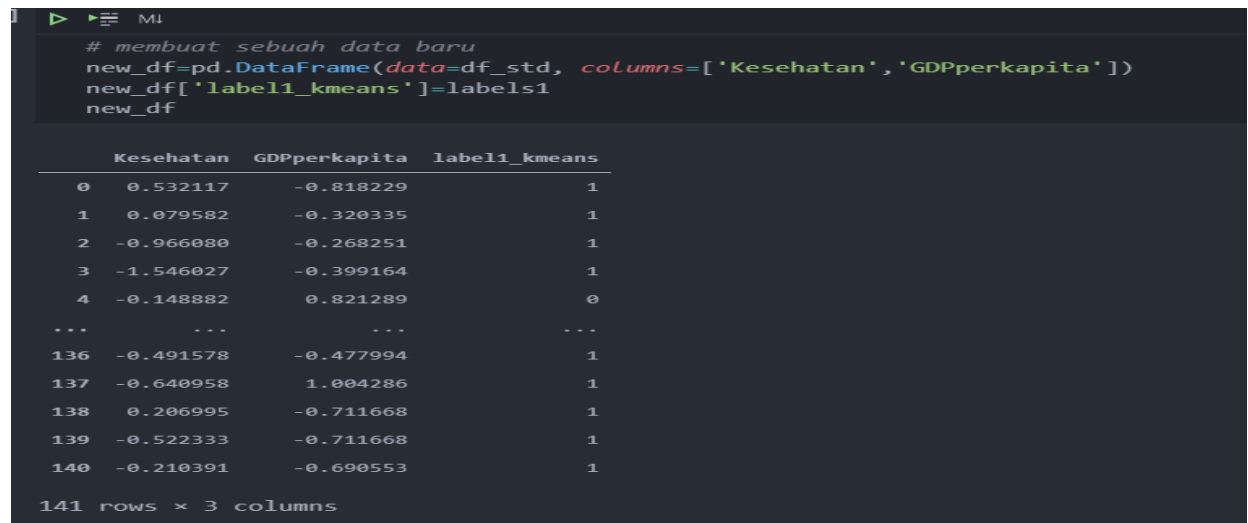
[22] ▶ MI
from sklearn.cluster import KMeans
kmeans1=KMeans(n_clusters=2,random_state=42).fit(df_std)
labels1=kmeans1.labels_
labels1

array([1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0,
       1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 1, 1])
```

Gambar 12 hasil clustering data dengan n_clusters=2

Pada gambar 12 dimana clustering datanya dibagi dengan 2 clusters dan untuk random_satenya adalah 42 sehingga menghasilkan data seperti pada gambar di atas yang outputnya 0 dan 1.

13. Membuat sebuah data frame yang baru dengan menampilkan kolom kesehatan, GDP perkapita dan juga menambahkan kolom label1_kmeans. Pada kolom label1_kmeans disini digunakan menampilkan cluster data yang memiliki nilai 0 dan 1. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini:



```
# membuat sebuah data baru
new_df=pd.DataFrame(data=df_std, columns=['Kesehatan', 'GDPperkapita'])
new_df['label1_kmeans']=labels1
new_df
```

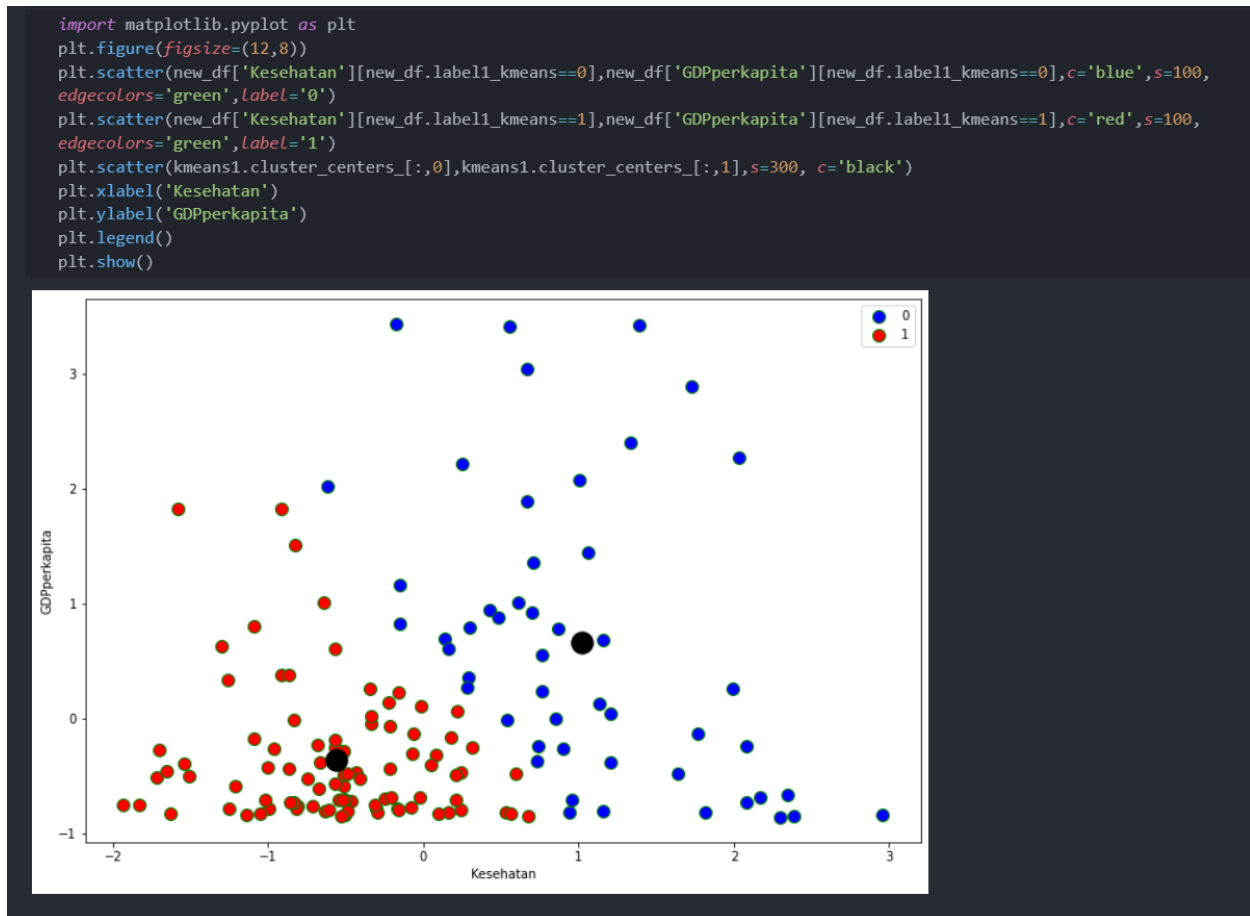
	Kesehatan	GDPperkapita	label1_kmeans
0	0.532117	-0.818229	1
1	0.079582	-0.320335	1
2	-0.966080	-0.268251	1
3	-1.546027	-0.399164	1
4	-0.148882	0.821289	0
...
136	-0.491578	-0.477994	1
137	-0.640958	1.004286	1
138	0.206995	-0.711668	1
139	-0.522333	-0.711668	1
140	-0.210391	-0.690553	1

141 rows x 3 columns

Gambar 13 data frame baru yang terdapat kolom Kesehatan dan GDP perkapita

Pada gambar 13 terlihat sebuah tabel yang memiliki 141 baris dan 3 kolom. Dimana pada kolom yang baru terdapat kolom label1_Kmeans digunakan untuk mempermudah pembaca mahami cluster datanya dimana pada kolom tersebut hanya memiliki 2 nilai yaitu 0 dan 1.

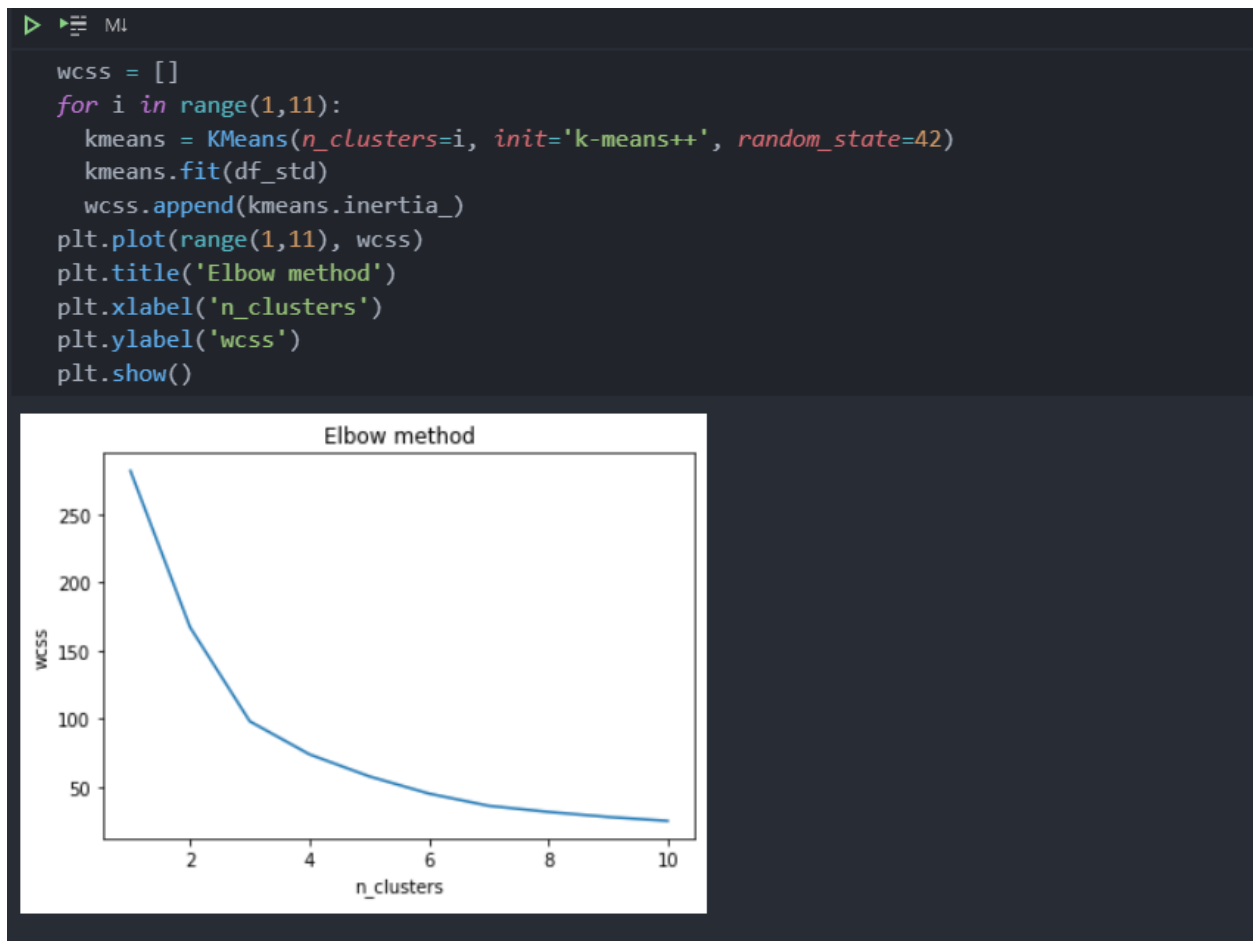
14. Langkah empat belas yaitu selanjutnya memploting data dengan method scatter untuk mengetahui sebaran data dari kesehatan dan GDP perkapita. Untuk lebih jelasnya dapat dilihat pada gambar dibawah hasil plotting data dengan method scatter sebagai berikut:



Gambar 14 hasil scatter new_df(data baru) Kesehatan dengan GDP perkapita

Berdasarkan hasil plotting data dengan method scatter didapatkan hasil dua cluster yaitu untuk cluster pertama ditandai dengan warna merah dan untuk cluster kedua di tandai dengan warna biru. Pada cluster pertama terlihat agak lebih meumpuk atau lebih rapat dari pada cluster kedua. untuk nilai kesahatan pada warna merah di cluster pertama nilainya dari -2 sampai dengan 0,8 an sedangkan nilai kesahatan pada warna biru di cluster kedua nilainya dari 0.8an sampai dengan 3.

15. Langkah 15 yaitu mencari nilai cluster yang tepat dengan menggunakan method Elbow. Hasilnya dapat dilihat pada gambar dibawah ini:



Gambar 15 hasil plotting data dari range 1 sampai 11 dengan wcss

Berdasarkan gambar 15 dapat dilihat grafik hubungan antran $n_clusters$ dengan wcss nya. Pada grafik diatas dapat kita pilih clusters yang tepat digunakan. Pada grafik diatas cluster yang sekiranya mendekati yaitu 4.

16. Langkah 16 yaitu membuat clustering data baru dengan clusters=4 s dan juga menambahkan kolom labels2_kmeans. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini:

```
[23] ▶ ▶≡ ML
      kmeans2 = KMeans(n_clusters = 4, init='k-means++', random_state=42)
      kmeans2.fit(df_std)
      labels2 = kmeans2.labels_

[24] ▶ ▶≡ ML
      new_df['label2_kmeans'] = labels2
      new_df
```

	Kesehatan	GDPperkapita	label1_kmeans	label2_kmeans
0	0.532117	-0.818229	1	2
1	0.079582	-0.320335	1	1
2	-0.966080	-0.268251	1	1
3	-1.546027	-0.399164	1	1
4	-0.148882	0.821289	0	0
...
136	-0.491578	-0.477994	1	1
137	-0.640958	1.004286	1	0
138	0.206995	-0.711668	1	1
139	-0.522333	-0.711668	1	1
140	-0.210391	-0.690553	1	1

141 rows × 4 columns

Gambar 16 hasil 4 cluster dan juga penambahan kolom untuk label2_kmeans

Pada gambar 16 sudah terlihat ada penambahan kolom baru yaitu label2_kmeans. Pada kolom ini yaitu nilai-nilai yang terdapat di dalam kolom tersebut adalah dari 0 sampai 3 karena penambahan cluster baru yang ditentukan berdasarkan hasil plot grafik menggunakan method elbow

17. Melakukan plotting data dengan method scatter pada kolom Kesehatan dan GDP perkapita dengan cluster yang baru. Untuk hasil plottingnya dapat dilihat pada gambar dibawah ini:



Gambar 17 hasil plotting data menggunakan method scatter dengan n_clusters=4

Pada gambar 17 dari hasil plotting data dapat dilihat terdapat 4 warna yang berbeda yang menandakan cluster datanya. Untuk warna biru menandakan cluster 1, warna merah menandakan cluster 2, warna hijau menandakan cluster 3 dan warna kuning menandakan cluster 4 sedangkan untuk warna hitam yang besar itu centroidnya. pada gambar diatas juga dapat dilihat sebaran data masing-masing cluster. Untuk sebaran data yang paling rapat yaitu warna merah pada cluster 2 sedangkan untuk sebaran data yang paling renggang atau menjauh yaitu warna kuning pada cluster 4.

18. Langkah ke 18 yaitu menentukan cluster yang baik digunakan. Berdasarkan hasil perhitungan dengan silhouette_score yang di import dari sklearn.metric untuk hasilnya dapat dilihat pada gambar dibawah ini:

```
[26] > ML

from sklearn.metrics import silhouette_score

print(silhouette_score(df_std, labels= labels1))
print(silhouette_score(df_std, labels= labels2))

0.4323768091297318
0.4416345225917429
```

Gambar 18 hasil perhitungan skor terbaik dari kedua cluster dengan silhouette_score

Pada gambar 18 didapatkan hasil untuk n_clusters=2 adalah 0.4324 sedangkan untuk n_clusters=4 didapatkan hasil adalah 0.4417. Dari hasilnya antara n_cluster=2 dan n_clusters=4 didapatkan hasil n_clusters=4 lebih besar dari n_clusters=2. Oleh karena itu untuk n_clusters yang baik digunakan berdasarkan nilainya yaitu n_clusters=4.

19. Langkah ke 19 yaitu menambahkan kolom baru dengan nama K_means_labels yang digunakan untuk menampung data cluster yang bernilai 0 sampai dengan 3. Sebelum melakukan menambahkan kolom terlebih dahulu di sesuaikan ukuran baris dari data yang di tambahkan dengan cara mengambil nilai yang paling belakang saja dengan method tail. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini:

```
[61] > ML

#summary
df_baru=df.tail(141)

df_baru['K_means_labels']= kmeans2.labels_

[62] > ML

df_baru
```

	Negara	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita	K_means_labels
26	Burundi	93.6	8.92	11.60	39.2	764	12.300	57.7	6.26	231	2
27	Cambodia	44.4	54.10	5.68	59.5	2520	3.120	66.1	2.88	786	1
28	Cameroon	100.0	22.20	5.13	27.0	2660	1.910	57.3	5.11	1310	1
29	Canada	5.6	29.10	11.30	31.0	40700	2.870	81.3	1.63	47400	1
30	Cape Verde	26.5	32.70	4.09	61.8	5830	0.505	72.5	2.67	3310	0
...
162	Vanuatu	29.2	46.60	5.25	52.7	2950	2.620	63.0	3.50	2970	1
163	Venezuela	17.1	28.50	4.91	17.6	16500	45.900	75.4	2.47	13500	0
164	Vietnam	23.3	72.00	6.84	80.2	4490	12.100	73.1	1.95	1310	1
165	Yemen	56.3	30.00	5.18	34.4	4480	23.600	67.5	4.67	1310	1
166	Zambia	83.1	37.00	5.89	30.9	3280	14.000	52.0	5.40	1460	1

141 rows x 11 columns

```
[~] > ML
```

Gambar 19 Table setelah melakukan penambahan kolom K-means-labels

Pada Gambar 19 terlihat bahwa ada penambahan kolom baru yaitu K_menas_labels yang berisi nilai 0 samapai dengan 3. Dimana nilai-nilai tersebut digunakan untuk cluster data.

20. Langkah 20 yaitu melakukan removing outlier dan melakukan clustering data dengan menjadi 4 bagian. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini:

```

> ▶ MI

df3 = remove_outlier(df[['Kesehatan', 'GDPperkapita', 'Pendapatan']])
df3.dropna(axis=0, inplace=True)

> ▶ MI

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
df_std = sc.fit_transform(df3.astype(float))

> ▶ MI

kmeans3 = KMeans(n_clusters = 4, init='k-means++', random_state=42).fit(df_std)
labels3 = kmeans3.labels_

> ▶ MI

new_df = pd.DataFrame(data=df_std, columns=['Kesehatan', 'GDPperkapita', 'Pendapatan'])
new_df['label3_kmeans'] = labels3
new_df

```

	Kesehatan	GDPperkapita	Pendapatan	label3_kmeans
0	0.532117	-0.818229	-0.949892	2
1	0.079582	-0.320335	-0.107301	1
2	-0.966080	-0.268251	0.193479	1
3	-1.546027	-0.399164	-0.515431	1
4	-0.148882	0.821289	0.821371	0
...
136	-0.491578	-0.477994	-0.814186	1
137	-0.640958	1.004286	0.558062	0
138	0.206995	-0.711668	-0.658226	1
139	-0.522333	-0.711668	-0.659239	1
140	-0.210391	-0.690553	-0.780766	1

141 rows × 4 columns

Gambar 20 hasil removing outlier dan clustering data dengan n_clusters=4

Pada gambar 20 terlihat jelas untuk data yang di remove outliernya yaitu pada kolom Kesehatan,GDPperkapita dan Pendapatan. Setelah berhasil melakukan clustering data langkah selanjutnya yaitu membuat data frame dengan nama kolomnya yaitu Kesehatan,GDPperkapita, dan Pendapatan. Untuk hasilnya dapat dilihat pada gambar diatas.

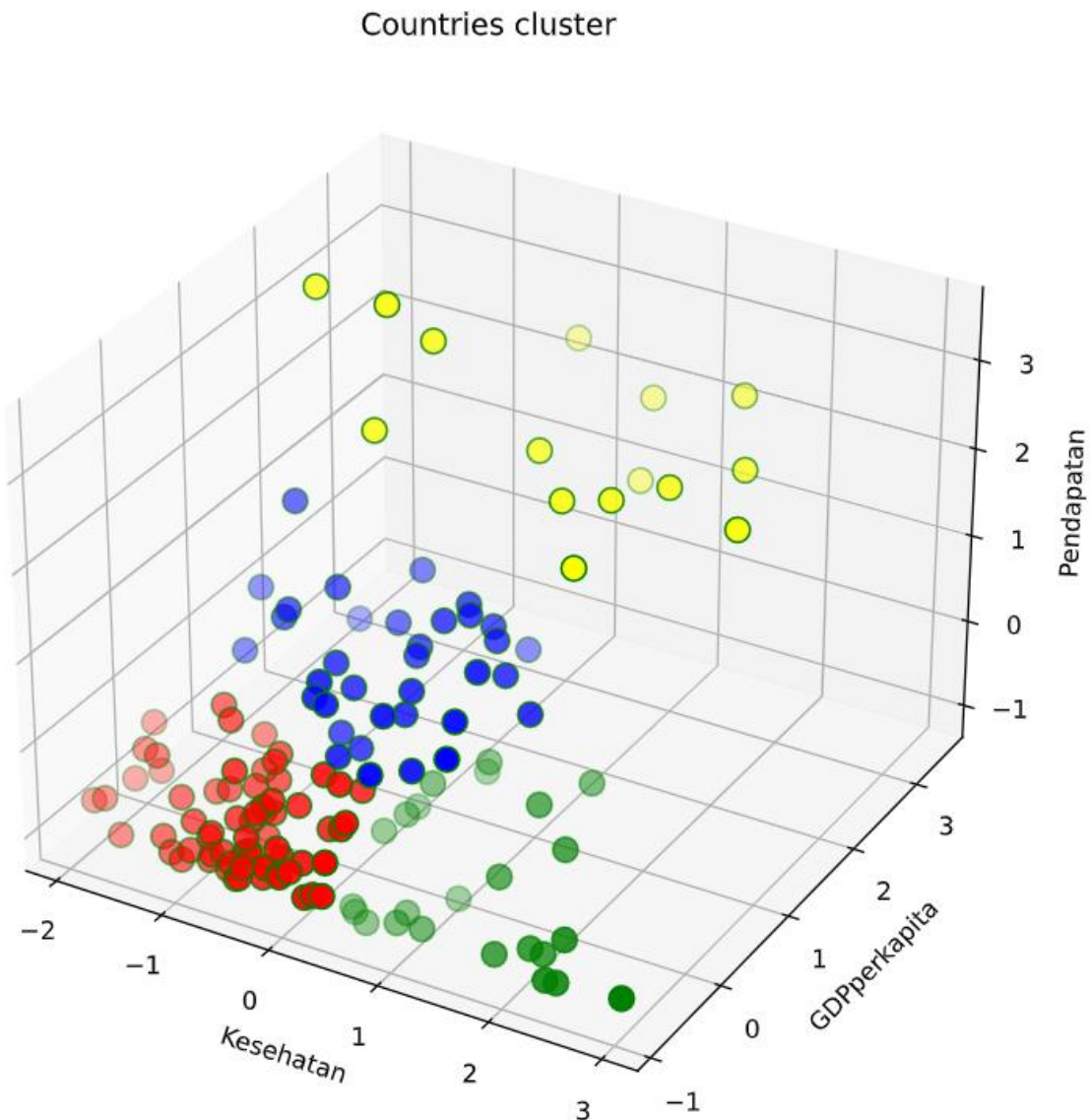
21. Selanjutnya yaitu melakukan plotting dengan method scatter untuk 3 dimensinya. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini

```

fig = plt.figure(figsize=(12,8))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(new_df['Kesehatan'][new_df.label3_kmeans==0], new_df['GDPperkapita'][new_df.label3_kmeans==0], new_df['Pendapatan']
[new_df.label3_kmeans==0], c='blue', s=100, edgecolors='green')
ax.scatter(new_df['Kesehatan'][new_df.label3_kmeans==1], new_df['GDPperkapita'][new_df.label3_kmeans==1], new_df['Pendapatan']
[new_df.label3_kmeans==1], c='red', s=100, edgecolors='green')
ax.scatter(new_df['Kesehatan'][new_df.label3_kmeans==2], new_df['GDPperkapita'][new_df.label3_kmeans==2], new_df['Pendapatan']
[new_df.label3_kmeans==2], c='green', s=100, edgecolors='green')
ax.scatter(new_df['Kesehatan'][new_df.label3_kmeans==3], new_df['GDPperkapita'][new_df.label3_kmeans==3], new_df['Pendapatan']
[new_df.label3_kmeans==3], c='yellow', s=100, edgecolors='green')

plt.title('Countries cluster')
plt.xlabel('Kesehatan')
plt.ylabel('GDPperkapita')
ax.set_zlabel('Pendapatan')
plt.show()

```



Gambar 21 hasil plotting data dengan method scatter untuk 3 dimensi

Pada gambar 21 terlihat bahwa untuk sumbu x mengacu ke nilai kesehatan, sumbu y mengacu ke nilai GDP perkapita sedangkan untuk sumbu z mengacu ke nilai pendapatan. Untuk warna biru menandakan cluster 1, warna merah menandakan cluster 2, warna hijau menandakan cluster 3 dan warna kuning menandakan cluster 4. Pada gambar diatas untuk cluster 2 memiliki kerapatan yang tinggi nilainya daripada cluster lainnya.