CLUSTERING THE COUNTRIES BY USING K-MEANS FOR HELP INTERNATIONAL

Nur Rochman Darmawan

OUTLINE

1. Project Understanding
2. The Data
3. Clustering
4. Recommendation

PROJECT UNDERSTANDING

HELP International adalah LSM kemanusiaan internasional yang berkomitmen untuk memerangi kemiskinan dan menyediakan fasilitas dan bantuan dasar bagi masyarakat di negara-negara terbelakang saat terjadi bencana dan bencana alam.

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.

THE DATA

Data yang digunakan adalah data negara HELP. Data ini memiliki 10 kolom dan 167 baris.

Penjelasan kolom pada data adalah sebagai berikut:

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

EXPLORATORY DATA ANALYSIS (EDA) (1)

Seperti yang ditampilkan disamping, kolom **Negara** merupakan satu-satunya kolom yang bertipe data **object**. Sedangkan kolom lainnya bertipe data **float** atau **int**.

Selain itu, semua data pada kolom tidak mempunyai nilai **NULL** atau **NaN**, hal ini terlihat pada kolom **Non-Null Count** yang seluruhnya bernilai **167** (sesuai dengan **Rangeindex**)

```
1  # Melihat info dari dataset
2  dataset.info()
```

✓ 0.0s

```
<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
```

EXPLORATORY DATA ANALYSIS (EDA) (2)

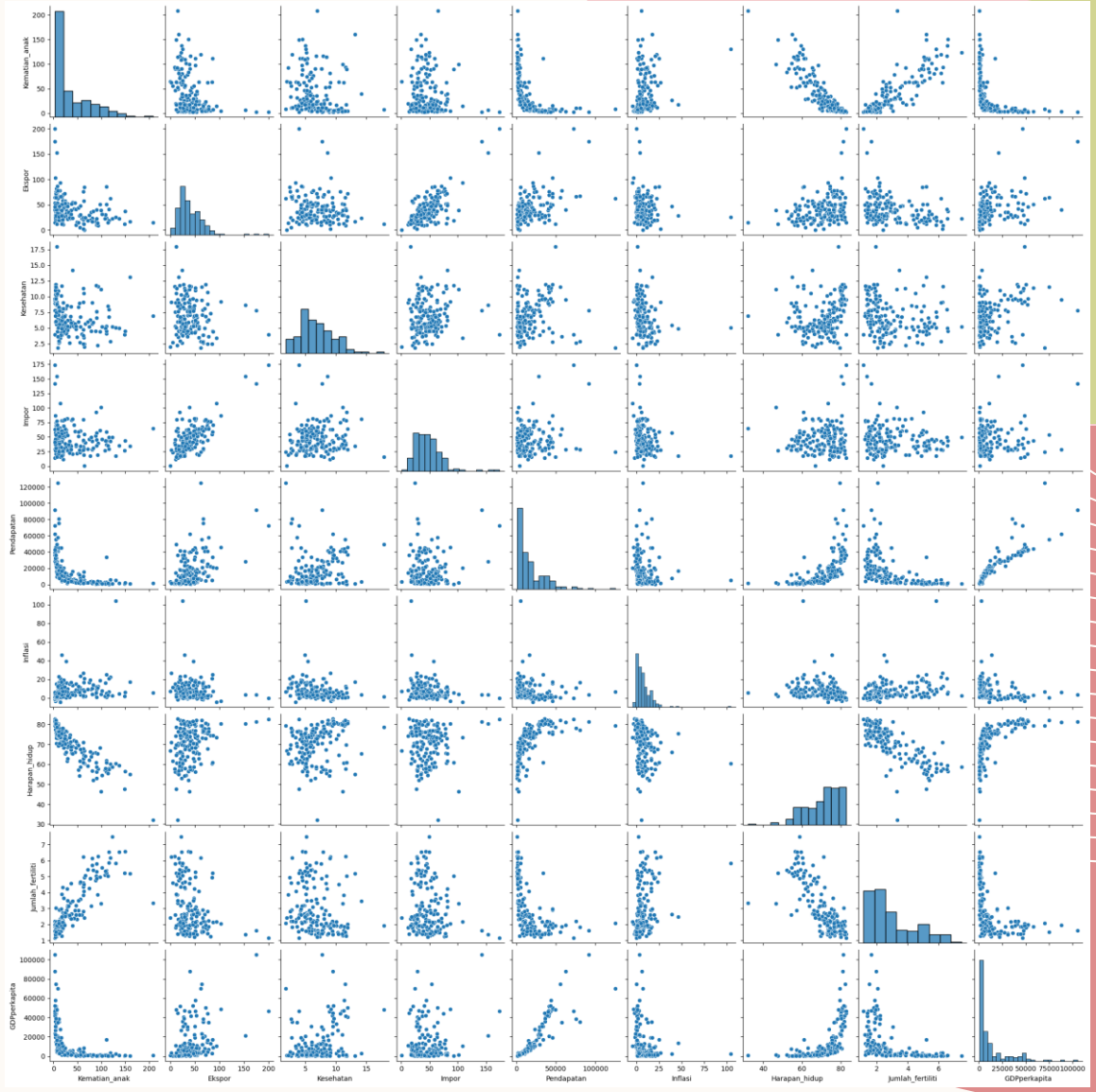
	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
count	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000
mean	38.270060	41.108976	6.815689	46.890215	17144.688623	7.781832	70.555689	2.947964	12964.155689
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 diatas menunjukkan statistic deskriptif dari data yang akan kita gunakan. Perhatikan kolom **Kematian_anak**, **Pendapatan**, **Inflasi**, dan **GDPperkapita**. Nilai **std** (standar deviasi) lebih besar dibandingkan dengan **mean** (rerata), hal ini menunjukkan bahwa sebaran data pada kolom-kolom tersebut sangat bervariasi. Sedangkan kolom-kolom lainnya memiliki sebaran data yang cenderung terpusat pada reratanya, sehingga karakteristik data pada kolom-kolom tersebut cenderung sama.

EXPLORATORY DATA ANALYSIS (EDA) (3)

Gambar disamping merupakan pair plot dari semua kolom bertipe numerik pada dataset yang kita gunakan.

- Kolom **GDPperkapita** dan **Pendapatan** memiliki korelasi yang linier (sebanding).
- Kolom **Kematian_anak** dan **Harapan_hidup** memiliki korelasi tidak linier (berbanding terbalik).
- Kolom **Kematian_anak** dan **Jumlah_fertility** memiliki korelasi yang relatif linier (sebanding).
- Kolom **Ekspor** dan **Impor** memiliki korelasi yang relatif linier (sebanding).
- Kolom **Jumlah_fertility** dan **Harapan_hidup** memiliki korelasi yang relatif tidak linier (berbanding terbalik).
- Kolom **Pendapatan** dan **Harapan_hidup** memiliki korelasi yang relatif linier (sebanding).



FEATURE SELECTION

	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
Kematian_anak	1.000000	-0.318093	-0.200402	-0.127211	-0.524315	0.288276	-0.886676	0.848478	-0.483032
Ekspor	-0.318093	1.000000	-0.114408	0.737381	0.516784	-0.107294	0.316313	-0.320011	0.418725
Kesehatan	-0.200402	-0.114408	1.000000	0.095717	0.129579	-0.255376	0.210692	-0.196674	0.345966
Impor	-0.127211	0.737381	0.095717	1.000000	0.122406	-0.246994	0.054391	-0.159048	0.115498
Pendapatan	-0.524315	0.516784	0.129579	0.122406	1.000000	-0.147756	0.611962	-0.501840	0.895571
Inflasi	0.288276	-0.107294	-0.255376	-0.246994	-0.147756	1.000000	-0.239705	0.316921	-0.221631
Harapan_hidup	-0.886676	0.316313	0.210692	0.054391	0.611962	-0.239705	1.000000	-0.760875	0.600089
Jumlah_fertiliti	0.848478	-0.320011	-0.196674	-0.159048	-0.501840	0.316921	-0.760875	1.000000	-0.454910
GDPperkapita	-0.483032	0.418725	0.345966	0.115498	0.895571	-0.221631	0.600089	-0.454910	1.000000

Pada pembuatan *machine learning* ini, saya akan menggunakan kolom **Pendapatan** sebagai representasi dari segi sosial-ekonomi dan kolom **Harapan_hidup** sebagai representasi dari sisi kesehatan.

Hal ini dikarenakan koefisien korelasi antar kolom yang cukup besar sehingga 2 variabel saling berhubungan positif (linier).

MISSING VALUE

Selanjutnya kita akan mengecek apakah data kita mengandung missing value atau tidak dengan method *.isnull()* dan *.sum()*.

Selanjutnya kita akan mengecek apakah data kita mengandung missing value atau tidak menggunakan method *.isnull()* dan *.sum()*.

Gambar disamping merupakan hasil pengecekan missing value. Missing value dari masing-masing kolom bernilai 0, yang berarti bahwa kolom-kolom tersebut tidak mempunyai missing value.

Setelah itu, kita mengecek tipe data dari masing-masing kolom apakah terdapat tipe data **object** atau tidak. Karena jika ada tipe data **object**, maka terdapat value yang rusak.

```
1  # Kita mengecek apakah ada missing value pada dataset kita
2  print(df_selected.isnull().sum())
3
4  # Kita mengecek apakah tipe data sudah sesuai (bukan berbentuk object)
5  print("\n")
6  print(df_selected.info())
```

✓ 0.0s

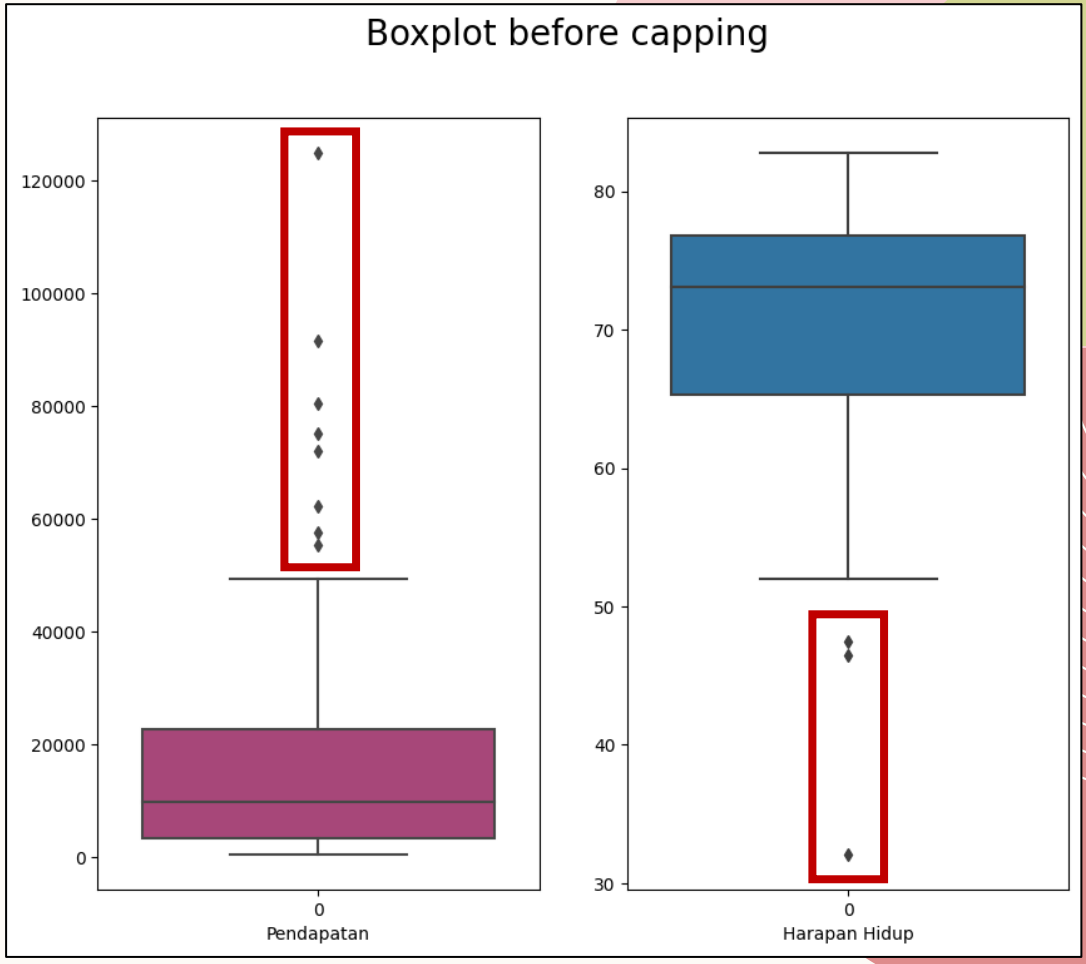
```
Pendapatan       0
Harapan_hidup    0
dtype: int64


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Pendapatan     167 non-null    int64
 1   Harapan_hidup  167 non-null    float64
dtypes: float64(1), int64(1)
memory usage: 2.7 KB
None
```

OUTLIER

Selanjutnya kita akan mengecek apakah data kita mengandung outlier menggunakan visualisassi boxplot dari Seaborn

Pada gambar boxplot disamping, masih terdapat outlier pada kolom-kolom yang akan kita gunakan.

Oleh karena itu, kita harus menangani data outlier agar tidak merusak hasil model cluster kita.



MENANGANI OUTLIER (1)

Ada beberapa cara untuk menangani outlier yang kita temukan, bisa dengan menghapus data outlier yang akan kita gunakan dan bisa dengan mengganti data outlier tersebut dengan batas extreme atas-bawah dari masing-masing kolom. Kita akan menggunakan metode yang kedua, yaitu dengan mengganti data outlier tersebut.

Pertama kita tentukan terlebih dahulu batas extreme bawah dan batas extreme atas dari masing-masing kolom.

```
4  # Kita mendefinisikan Q1 dan Q3 dari masing-masing kolom
5  quantile = df_selected.quantile([.25, .75])
6
7  Q1_pendapatan, Q3_pendapatan = [x[0] for x in quantile.values]
8  Q1_harapan, Q3_harapan = [x[1] for x in quantile.values]
9
10 # Kita mendefinisikan IQR dari masing-masing kolom
11 IQR_pendapatan = Q3_pendapatan - Q1_pendapatan
12 IQR_harapan = Q3_harapan - Q1_harapan
13
14 # Kita mendefinisikan min_limit dan max_limit dari masing-masing kolom
15 min_limit_pendapatan = Q1_pendapatan - (1.5 * IQR_pendapatan)
16 max_limit_pendapatan = Q3_pendapatan + (1.5 * IQR_pendapatan)
17 min_limit_harapan = Q1_harapan - (1.5 * IQR_harapan)
18 max_limit_harapan = Q3_harapan + (1.5 * IQR_harapan)
```

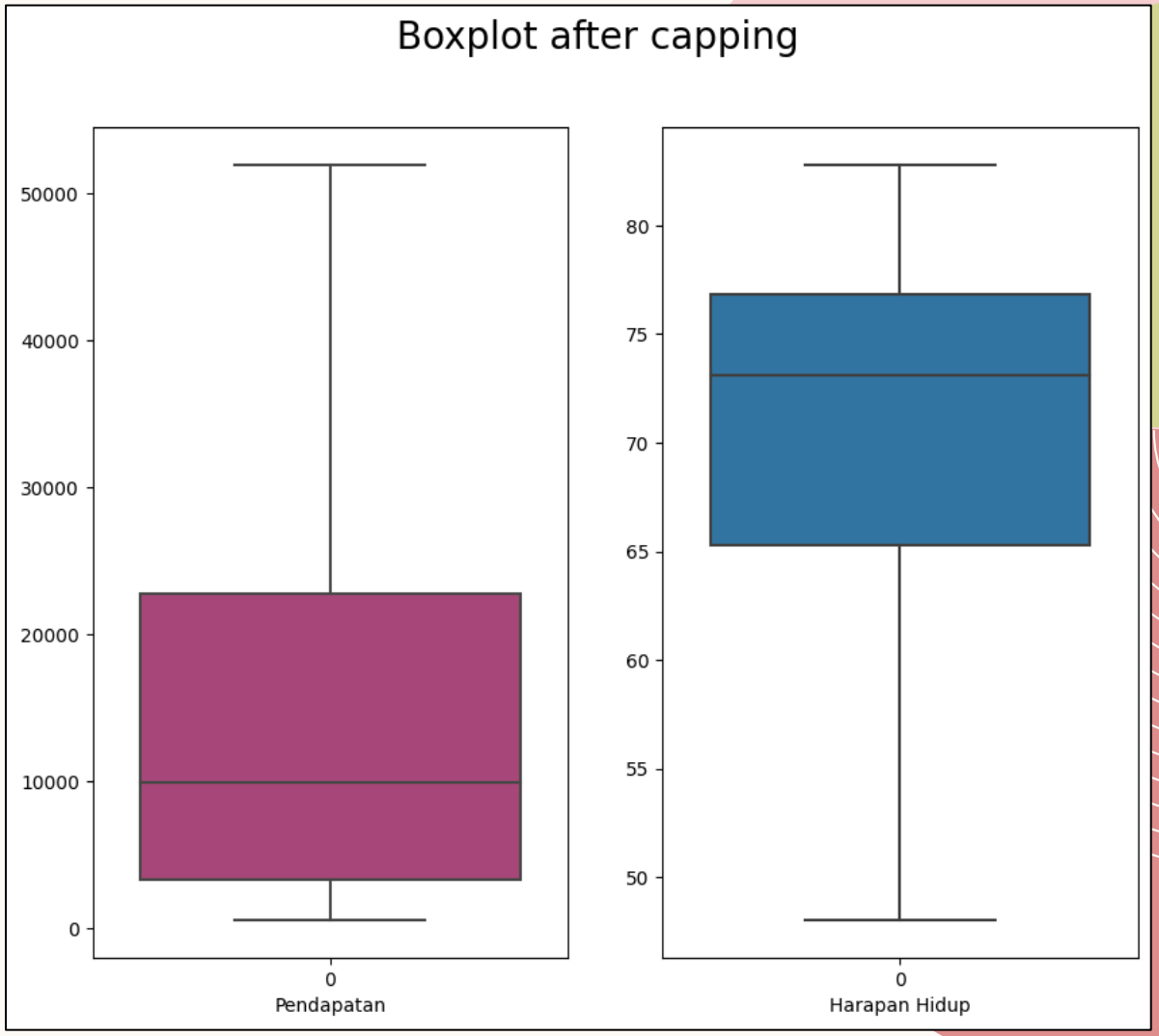
MENANGANI OUTLIER (2)

Setelah kita mendapatkan batas extreme bawah dan batas extreme atas pada masing-masing kolom, kita dapat mengganti data outlier tersebut dengan batas extreme.

```
# Kita mengubah data yang lebih kecil dari min_limit menjadi min_limit
# Dan mengubah data yang lebih besar dari max_limit menjadi max_limit

df_selected_new = pd.DataFrame({
    "Pendapatan": np.where(
        df_selected['Pendapatan'] < min_limit_pendapatan, min_limit_pendapatan,
        (np.where(
            df_selected['Pendapatan'] > max_limit_pendapatan, max_limit_pendapatan, df_selected['Pendapatan']
        ))
    ),
    "Harapan_hidup": np.where(
        df_selected['Harapan_hidup'] < min_limit_harapan, min_limit_harapan,
        (np.where(
            df_selected['Harapan_hidup'] > max_limit_harapan, max_limit_harapan, df_selected['Harapan_hidup']
        ))
    )
})
```

MENANGANI OUTLIER (3)

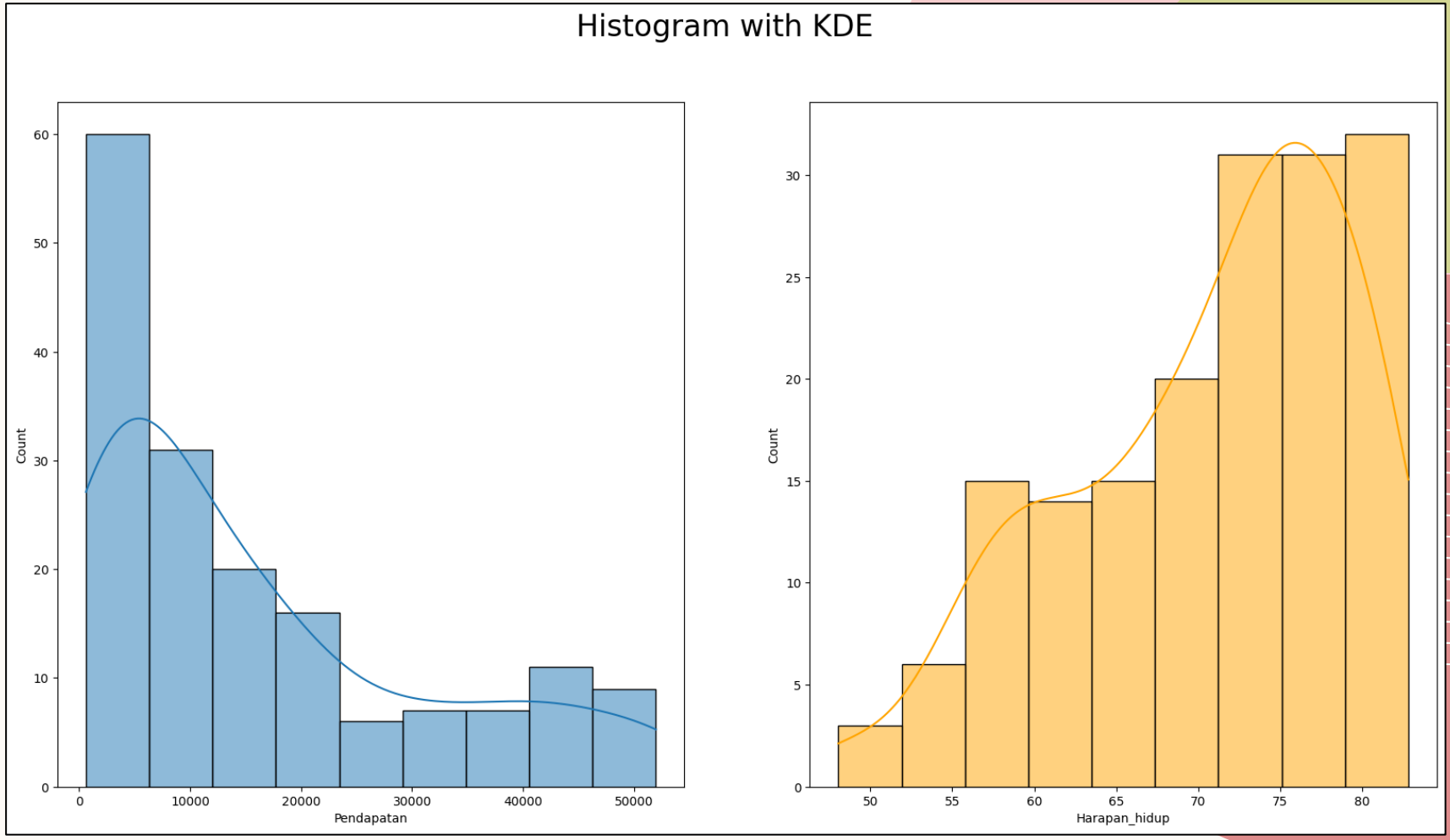
Pada gambar boxplot disamping (menggunakan Seaborn Boxplot), sudah tidak ada lagi outlier pada data di kolom-kolom yang akan kita gunakan.



UNIVARIATE ANALYSIS (1)

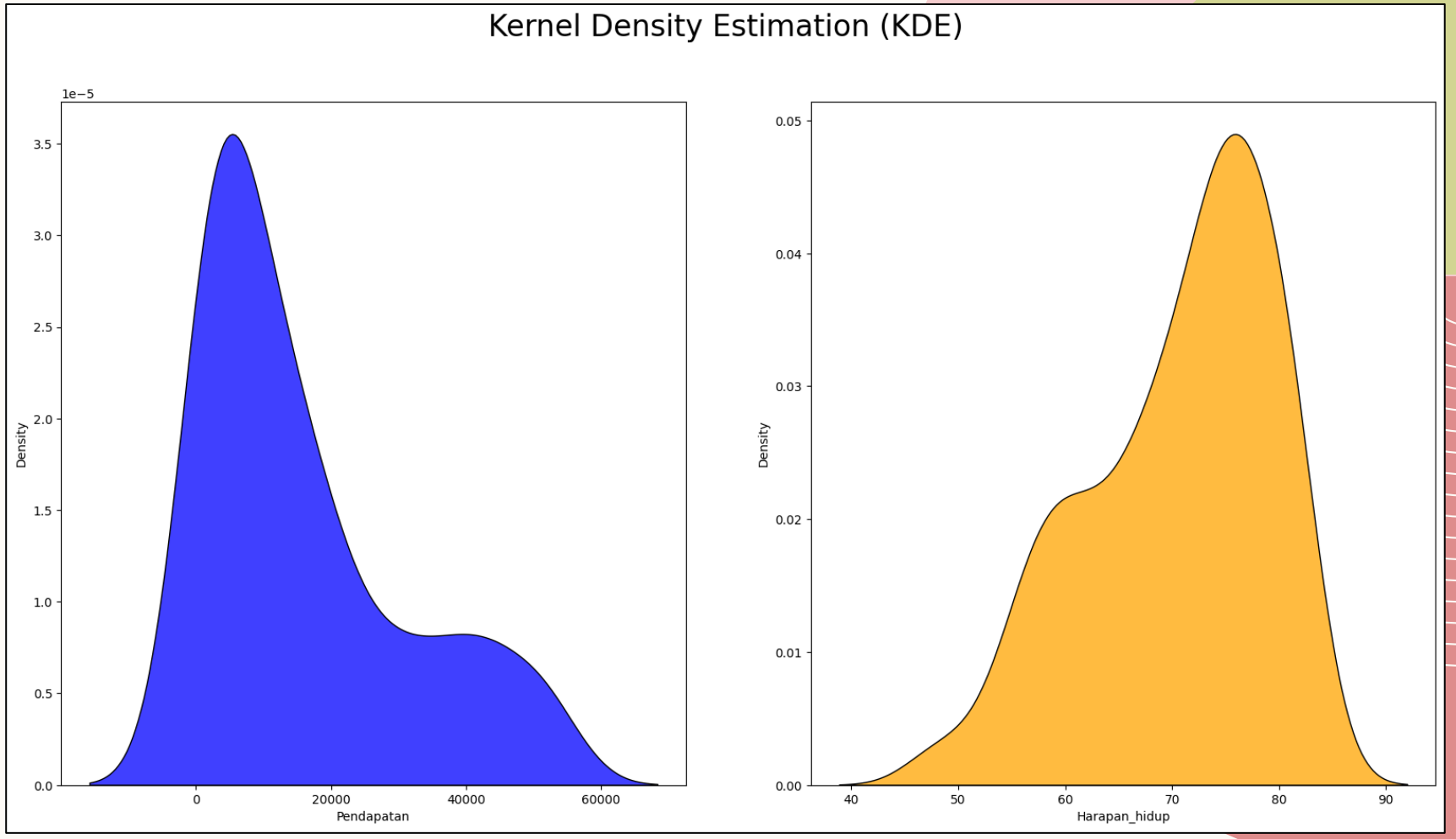
Kita menggunakan diagram Histogram untuk mengetahui persebaran data pada masing-masing kolom.

Kolom **Pendapatan** cenderung bersifat *positive skew*, sedangkan kolom **Harapan_hidup** cenderung bersifat *negative skew*.

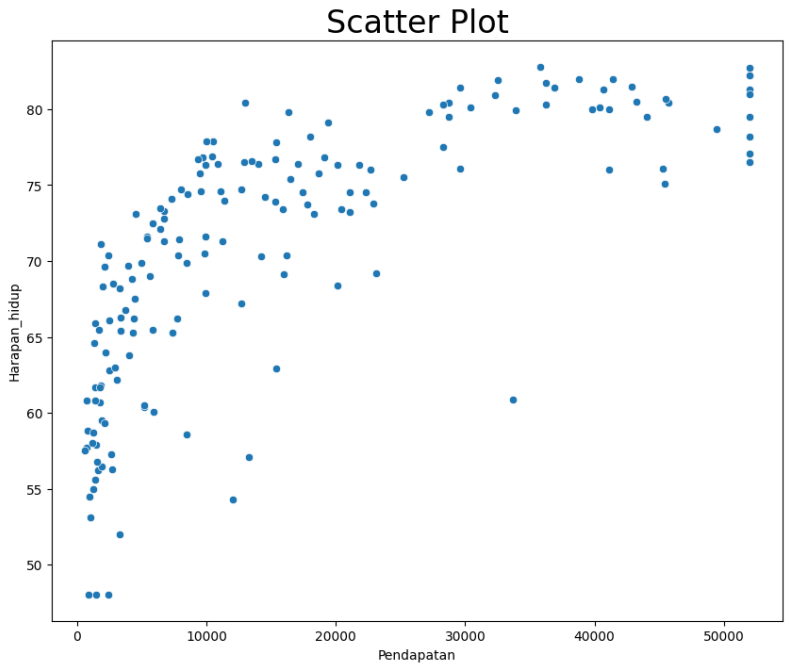


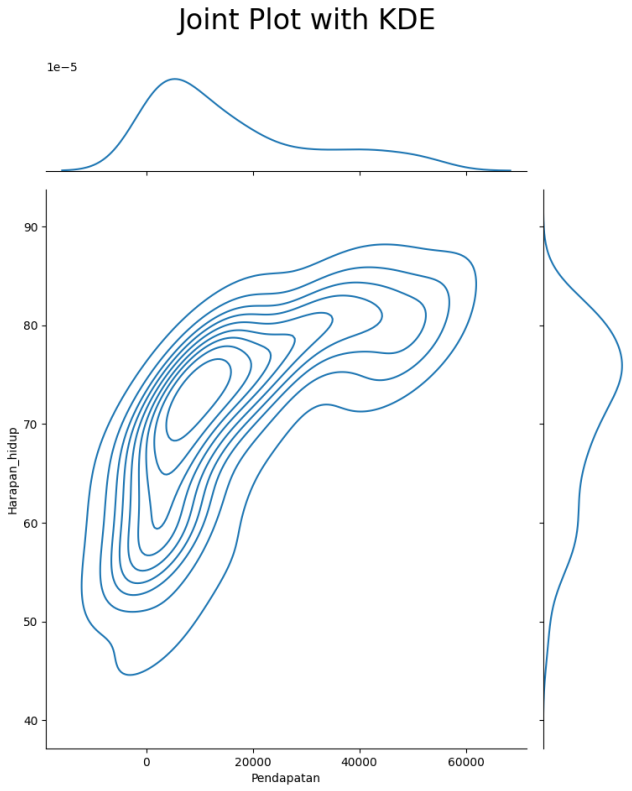
UNIVARIATE ANALYSIS (2)

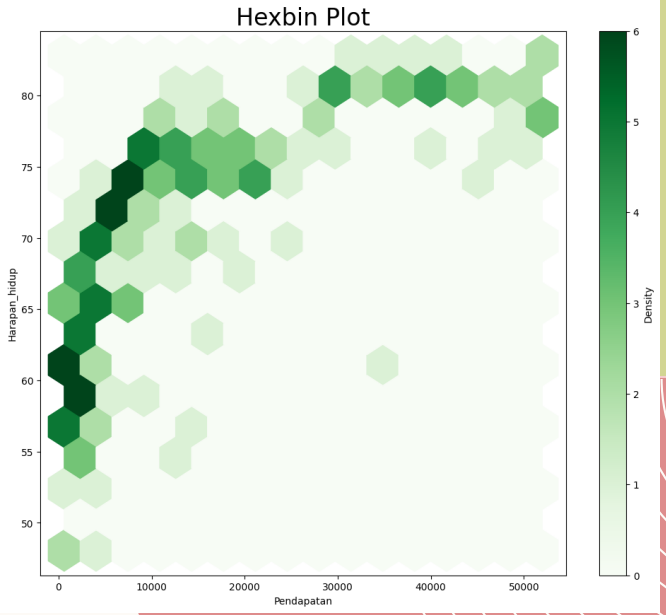
Kecenderungan *skewness* juga dapat dilihat dari diagram Kernel Density Estimation (KDE).



BIVARIATE ANALYSIS







Pada diagram *scatter plot*, *joint plot*, dan *hexbin plot* diatas, trend titik cenderung meningkat seiring bertambahnya nilai pada kolom-kolom dataset.

CLUSTER

Sebelum kita membuat model cluster, kita harus menormalisasi dataset kita agar mudah dipelajari oleh computer. Ada beberapa metode dalam menormalisasi data, namun kita akan menormalisasi data agar nilainya berada di antara 0 dan 1.

```python
# Menormalisasi dataset
def min_max_scaler(data):
    min_data = min(data)
    max_data = max(data)
    scaled_data = (data - min_data) / (max_data - min_data)
    return scaled_data

normalize_data = df_selected_new.apply(min_max_scaler)
```

	Pendapatan	Harapan_hidup
0	0.019490	0.234532
1	0.181489	0.812950
2	0.239318	0.818705
3	0.103021	0.346763
4	0.360038	0.827338
...	...	...
162	0.045582	0.430216
163	0.309413	0.787050
164	0.075567	0.720863
165	0.075372	0.559712
166	0.052007	0.113669

167 rows × 2 columns

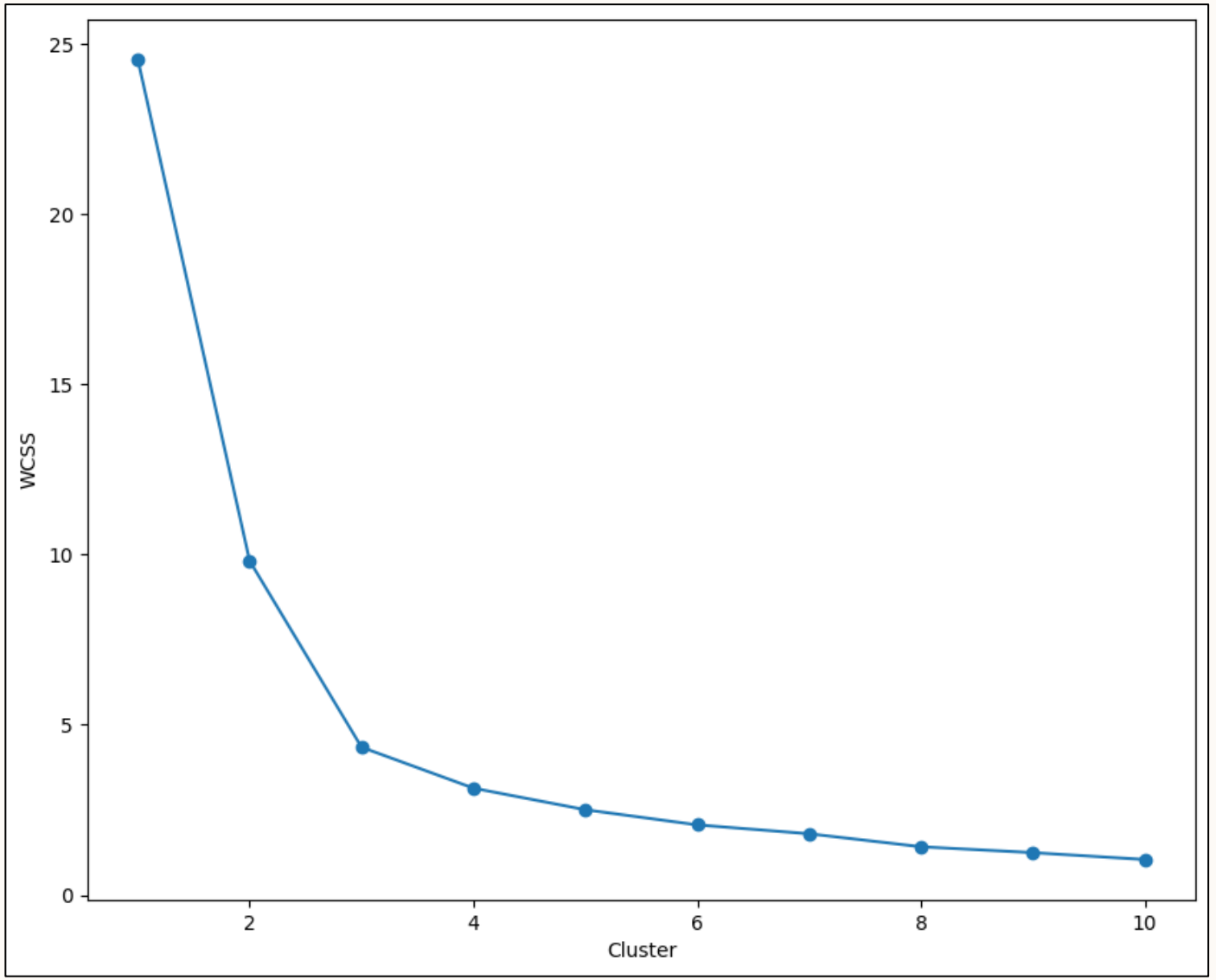
ELBOW METHOD (1)

Lalu kita menggunakan Elbow Method untuk menentukan jumlah cluster yang optimal sebagai acuan model kita dalam membuat clusternya.

```
# Mencari jumlah cluster menggunakan Elbow Method
wcss = []
k_list = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init="k-means++", random_state=1234, n_init="auto").fit(normalize_data)
    wcss.append(kmeans.inertia_)
    k_list.append(k)

plt.figure(figsize=(10, 8))
plt.plot(k_list, wcss, marker="o", label="Elbow Method")
plt.xlabel("Cluster")
plt.ylabel("WCSS")
plt.show()
```

ELBOW METHOD (2)

Pada diagram Elbow Method disamping, dapat dilihat bahwa cluster optimal berada pada angka 8.
Oleh karena itu, kita akan menggunakan 8 cluster dalam membuat model kita



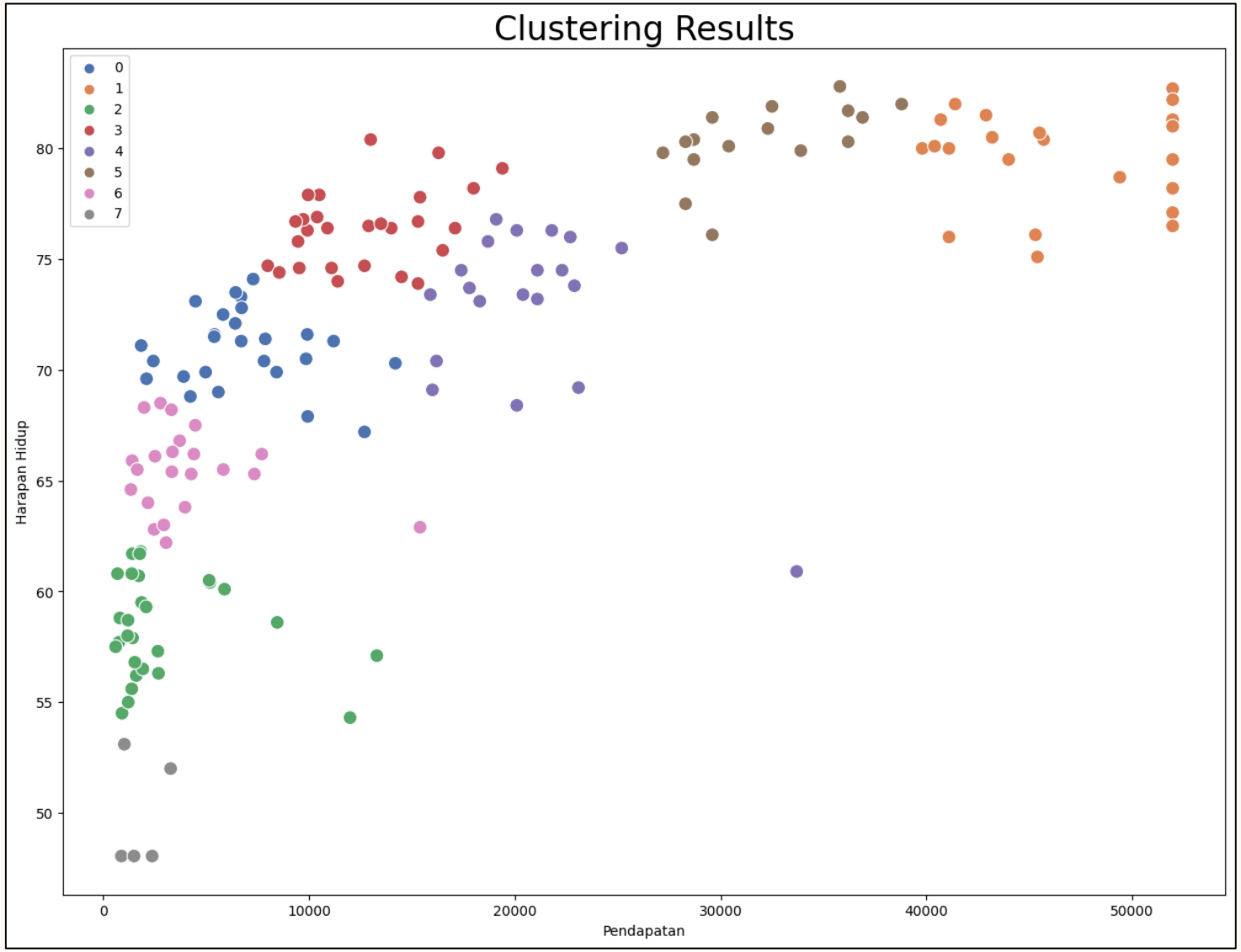
MODEL CLUSTER

```
# Kita akan melakukan Clustering menggunakan metode K-Means
# Dengan jumlah cluster yang dibuat adalah 8

kmeans_model = KMeans(n_clusters=8, random_state=1234, n_init="auto")
kmeans_model.fit(normalize_data)

kmeans_labels = kmeans_model.labels_
final_dataset = pd.DataFrame(df_selected_new, columns=["Pendapatan", "Harapan_hidup"])
final_dataset["Cluster"] = kmeans_labels
final_dataset["Negara"] = dataset["Negara"]

# Mengurutkan kolom-kolomnya
final_dataset = final_dataset[['Negara', 'Pendapatan', 'Harapan_hidup', 'Cluster']]
```

Langkah selanjutnya adalah membuat model cluster menggunakan *class* KMeans. Setelah membuat model, lalu kita mengurutkan kembali kolom-kolomnya agar bisa dimanfaatkan.

GRAFIK MODEL CLUSTER

Pada hasil Cluster disamping, dapat dilihat bahwa dataset yang kita gunakan sudah dikelompokkan menjadi 8 cluster.
Dapat dilihat dari gambar disamping, cluster nomor 7 merupakan kelompok negara yang memiliki **Pendapatan** dan **Harapan Hidup** yang relatif kecil. Sedangkan, cluster nomor 7 merupakan kelompok negara yang memiliki **Pendapatan** dan **Harapan Hidup** yang tinggi.



RECOMMENDATION (1)

Berdasarkan hasil cluster dari model yang dibuat, dapat disimpulkan bahwa kita seharusnya berfokus untuk membantu negara pada kelompok cluster nomor 7.

Hal ini didasari oleh kecilnya **Pendapatan** dan angka **Harapan Hidup** pada negara-negara yang termasuk ke dalam cluster 7.

```
1 # Negara dengan cluster nomor 7
2 final_dataset[final_dataset['Cluster'] == 7]
```

✓ 0.0s

	Negara	Pendapatan	Harapan_hidup	Cluster
31	Central African Republic	888.0	48.05	7
66	Haiti	1500.0	48.05	7
87	Lesotho	2380.0	48.05	7
94	Malawi	1030.0	53.10	7
166	Zambia	3280.0	52.00	7

RECOMMENDATION (2)

Diantara 5 negara yang termasuk cluster 7 tersebut, kami merekomendasikan negara **Central African Republic, Haiti,** dan **Malawi**.

Hal ini didasari oleh **Pendapatan** dan **Harapan Hidup** yang sangat kecil di negara-negara tersebut.

THANK YOU

Nur Rochman Darmawan
darmawan12.work@gmail.com