



PRIORITASI BANTUAN PEMBANGUNAN GLOBAL BERBASIS DATA UNTUK LSM HELP INTERNATIONAL

By: Vivian Olivia Frederica Simanjuntak

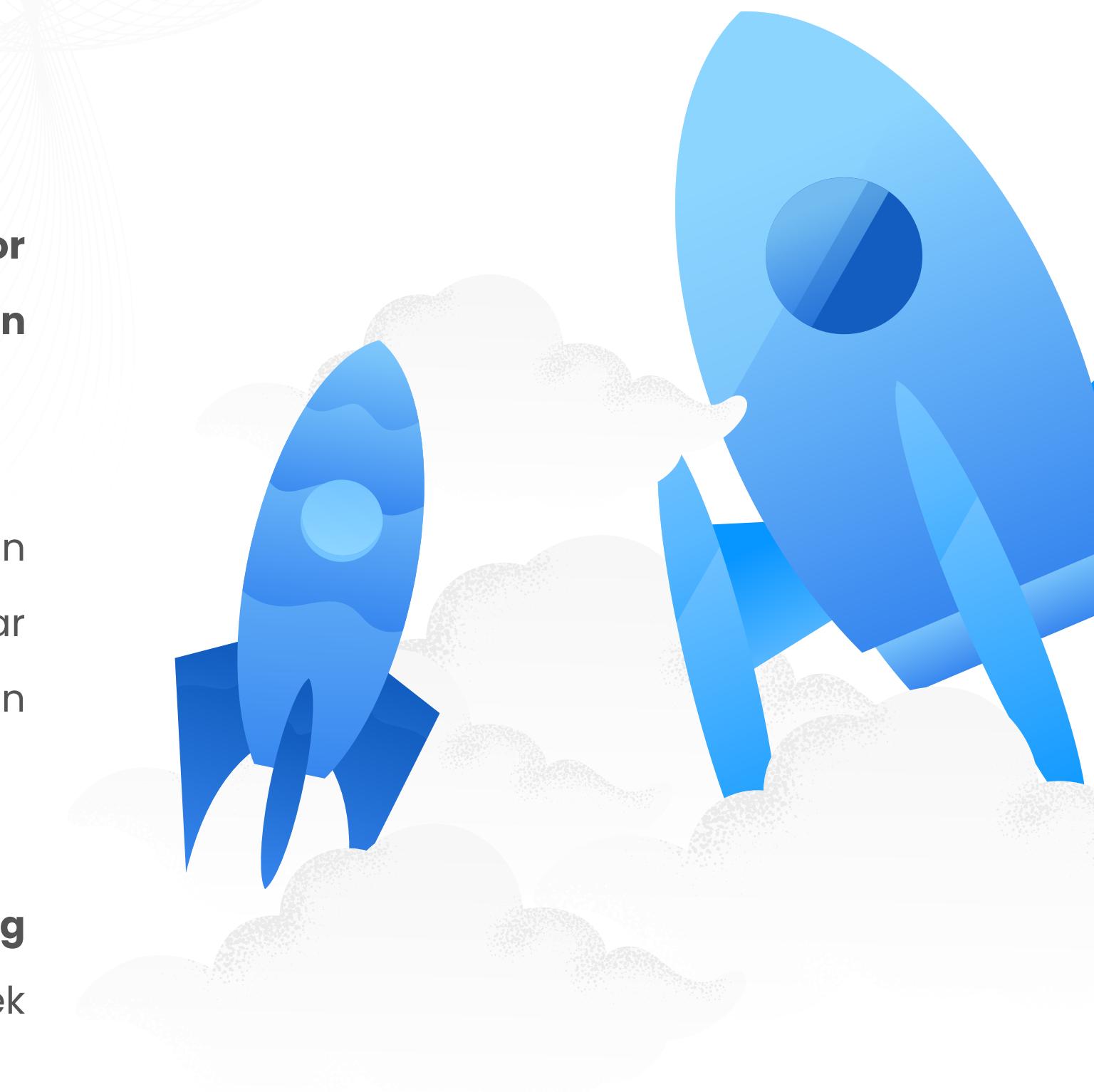
PROJECT UNDERSTANDING

Background

Proyek ini bertujuan **mengkategorikan negara-negara berdasarkan faktor sosial ekonomi dan kesehatan** untuk **menentukan prioritas bantuan pembangunan** bagi LSM HELP International.

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.

Dengan alokasi dana \$10 juta, CEO perlu **menentukan negara yang membutuhkan bantuan** terutama dalam kondisi krisis, dan tim proyek ditugaskan untuk memberikan analisis rekomendasi yang sesuai.



PROJECT UNDERSTANDING

Requirement

Presentasi proyek ini memerlukan:

- 1. Akses terhadap dataset** yang mencakup informasi sosial ekonomi dan kesehatan negara-negara. Pada proyek ini saya menggunakan data dari file 'Data_Negara_HELP.csv' yang telah disediakan.
- 2. Python dan algoritma analisis data.** Visualisasi yang efektif akan mendukung penyampaian temuan kepada pemangku kepentingan. Maka dari itu saya menggunakan bahasa pemrograman Python untuk analisis data dengan bantuan Jupyter Notebook sebagai platform untuk mengeksekusi kodingan.



DATESET UNDERSTANDING

Data Source

Data_Negara_HELP.csv file

Dataset Features

- 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.

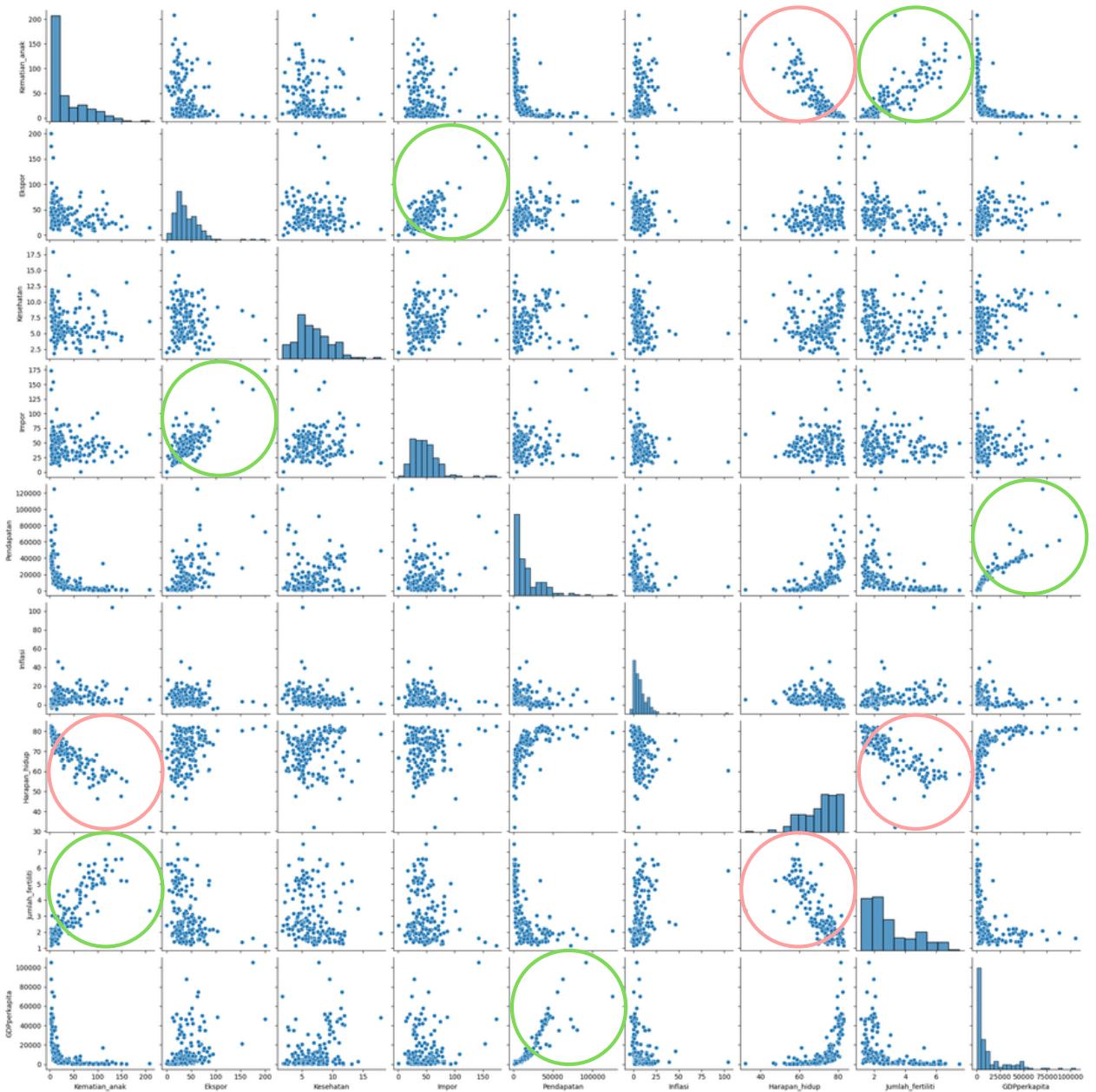
Number of Rows

167 rows



EDA 1: MULTIVARIATE

Pairplot



Explanation

Kode:

```
sns.pairplot(df, diag_kind='hist');  
plt.show()
```

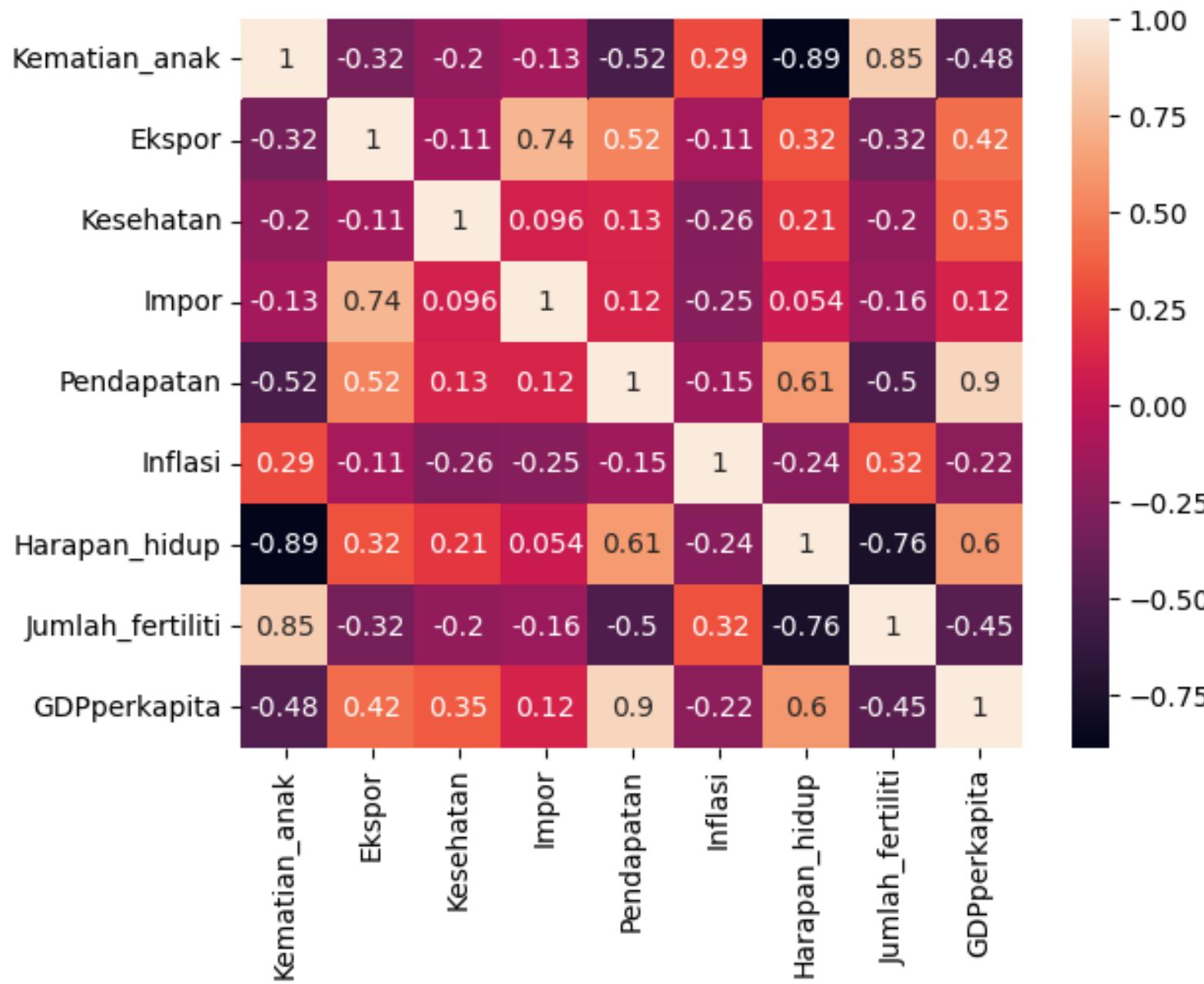
Penjelasan:

Pairplot merupakan jenis visualisasi yang digunakan untuk melihat korelasi antar data yang ada dalam sebuah tabel. Untuk menggunakan pairplot perlu melakukan import seaborn terlebih dahulu. `df` berarti dataframe yang saya gunakan dari file csv. `diag_kind = 'hist'` digunakan untuk menunjukkan histogram pada posisi diagonal

Dari hasil visualisasi, dapat dilihat bahwa beberapa *features* yang memiliki **korelasi positif** adalah 'Kematian_anak' dengan 'Jumlah_fertiliti', 'ekspor' dengan 'impor', dan 'pendapatan' dengan 'GDPperkapita'. Dimana korelasi antar *features* berbanding lurus. Sedangkan yang memiliki **korelasi negatif** adalah 'Kematian_anak' dengan 'Harapan_hidup' dan 'Jumlah_fertiliti' dan 'Harapan_hidup'. Yang berarti mereka bersifat berbanding terbalik.

EDA 1: MULTIVARIATE

Heatmap



Explanation

Kode:

```
sns.heatmap(df.corr(), annot=True, fmt='%.2g')
```

Penjelasan:

Heatmap merupakan jenis visualisasi lain yang digunakan untuk melihat korelasi antar *features*. Untuk menggunakan heatmap juga perlu *import* seaborn terlebih dahulu. `df` berarti dataframe yang saya gunakan dari file csv. `.corr()` untuk menghitung matriks korelasi antara kolom-kolom numerik dalam DataFrame. `annot=True` untuk menampilkan angka di dalam sel heatmap. `fmt='%.2g'` untuk menampilkan angka dengan dua angka desimal.

Cara membaca heatmap dan nilai korelasi yaitu jika **nilai korelasi > 0.7** berarti memiliki keterhubungan yang **kuat** (korelasi positif). Jika **nilai korelasi < -0.7** berarti memiliki keterhubungan yang **kuat** juga (korelasi negatif). Range dari nilai korelasi adalah 1 hingga -1. Semakin mendekati 1 atau -1 keterhubungan semakin kuat, jika 0 berarti tidak ada keterkaitan. Jika dilihat dari visualisasi, yang korelasinya paling kuat adalah 'Pendapatan' dengan 'GDPperkapita'.

FEATURE SELECTION

2 Features to be used as the base of analysis and clustering

1. Pendapatan

2. Harapan_hidup

Reason

Alasan saya memilih fitur **Pendapatan** karena menurut saya fitur ini bisa menjadi fitur yang tepat untuk mendeskripsikan keadaan **sosial ekonomi** suatu negara. Pendapatan merupakan fitur yang berisi data tentang penghasilan bersih perorangan di suatu negara, jadi bisa mengetahui keadaan ekonomi.

Alasan saya memilih fitur **Harapan_hidup** karena menurut saya fitur ini bisa menjadi fitur yang tepat untuk mendeskripsikan keadaan **kesehatan** suatu negara. Harapan_hidup merupakan fitur yang berisi data tentang jumlah tahun rata-rata seorang anak yang baru lahir akan hidup, jadi ini juga menyangkut tentang kesehatan seorang anak.

DATA CLEANING (MISSING VALUE)

Menggunakan .info()

```
# cara 1
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 11 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   
 10  Cluster       167 non-null   int32  
dtypes: float64(7), int32(1), int64(2), object(1)
memory usage: 13.8+ KB
```

Penjelasan:

Dari hasil koding tersebut dapat dilihat bahwa total jumlah baris pada Dataframe adalah 167. Kemudian jika dilihat di setiap kolomnya, jumlah non-null adalah 167, yang berarti semua datanya terisi dan **tidak ada kolom yang memiliki nilai null atau nilainya hilang.**

Menggunakan .describe()

```
# cara 2
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

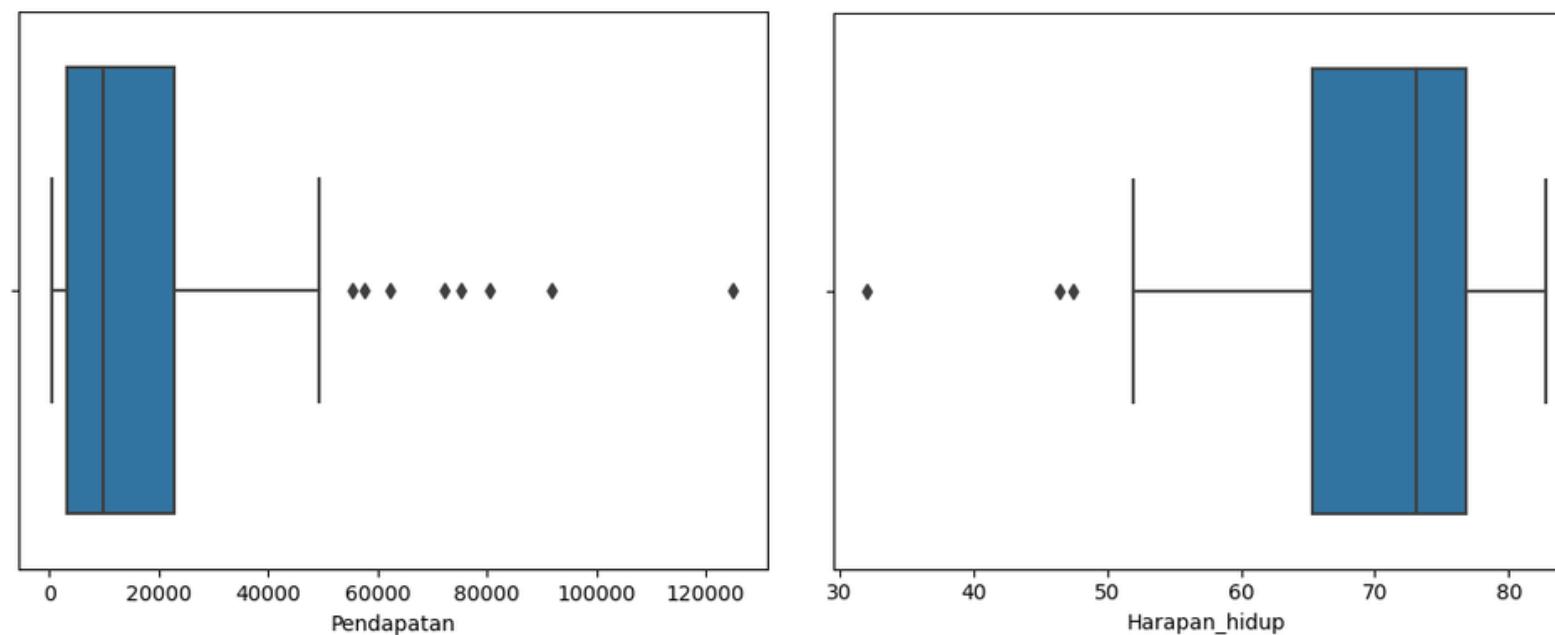
Penjelasan:

Ini merupakan cara kedua untuk mengecek apakah ada nilai null/NaN atau tidak. Kita telah mengetahui bahwa jumlah baris pada Dataframe adalah 167. Kemudian jika dilihat dari hasil count setiap kolom pada Dataframe, jumlah countnya semua mencapai 167, yang berarti data yang ada sudah lengkap dan **tidak ada missing value.**

DATA CLEANING (OUTLIERS)

1. Check Existence

```
sns.boxplot(x='Pendapatan', data=df)
plt.show()
sns.boxplot(x='Harapan_hidup', data=df)
plt.show()
```



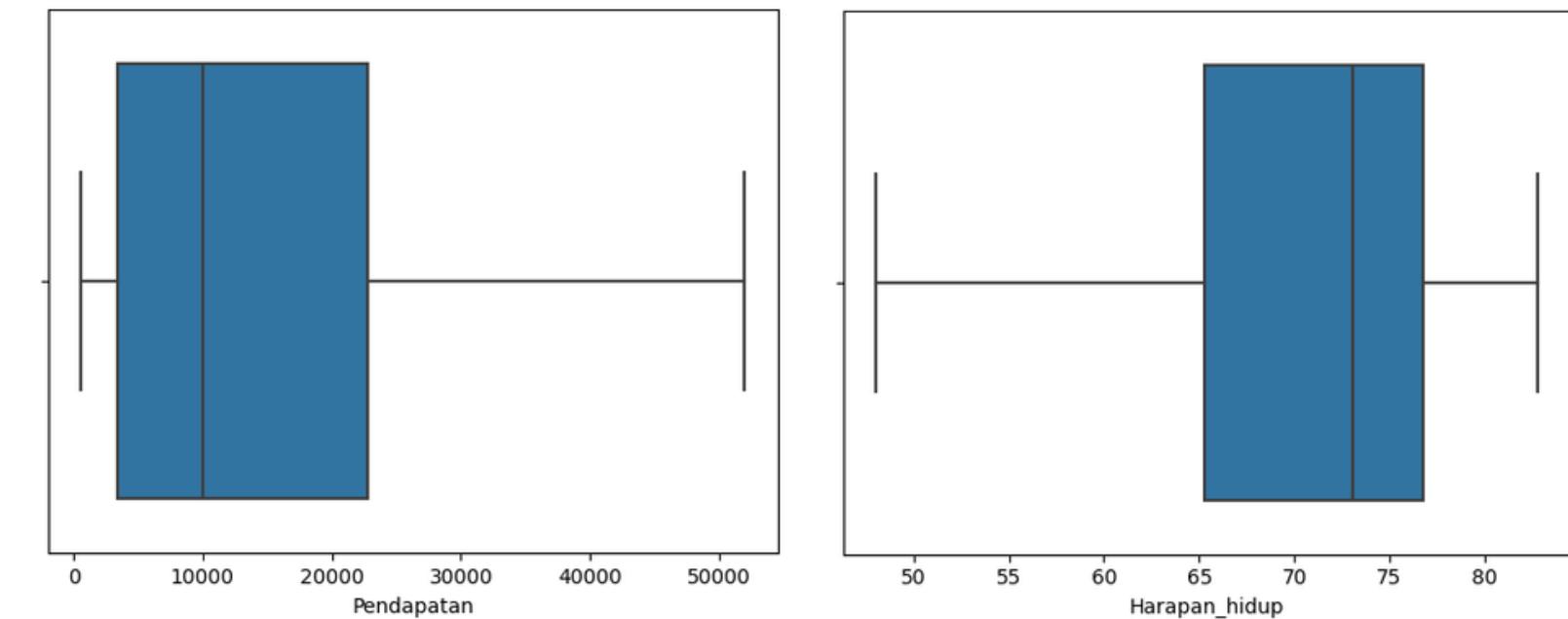
2. Handling

```
def handle_outlier_IQR(df):
    Q1 = df.quantile(0.25)
    Q3 = df.quantile(0.75)
    IQR = Q3 - Q1
    lb = Q1 - 1.5 * IQR
    ub = Q3 + 1.5 * IQR
    df_handled = df.copy()
    df_handled = df_handled.clip(lower=lb, upper=ub, axis=1)
    return df_handled

df_outlier_handled = handle_outlier_IQR(df[['Pendapatan', 'Harapan_hidup']])
```

3. Recheck

```
sns.boxplot(x='Pendapatan', data=df_outlier_handled)
plt.show()
sns.boxplot(x='Harapan_hidup', data=df_outlier_handled)
plt.show()
```



DATA CLEANING (OUTLIERS)

4. How I Handle Them

Cara saya mengatasi outliers, dimana outlier merupakan nilai yang jauh berbeda atau tidak biasa dari nilai-nilai lain dalam suatu dataset, pertama yaitu dengan mengeceknya menggunakan boxplot. Dapat dilihat ada beberapa titik/diamond di luar garis atau box. Untuk mengatasinya, saya menggunakan sebuah fungsi untuk **mengubah nilai outliers menjadi batas atas jika nilainya lebih dari Q3, dan menjadi batas bawah jika nilainya kurang dari Q1**. Setelah saya mengubah nilai outliers, maka saya mengecek kembali dengan boxplot, memastikan sudah tidak ada lagi titik yang berada di luar box.

5. Why Do I Handle Them in Such Way

Alasan saya mengatasi outliers dengan mengganti nilai outliers dan tidak menghapusnya adalah **karena asumsi data yang saya gunakan adalah real**, jadi saya tidak dapat secara langsung melakukan penghapusan. Maka dari itu saya tidak menghapusnya, melainkan menggantinya dengan nilai batas atas dan bawah.

EDA 2: UNIVARIATE

Code

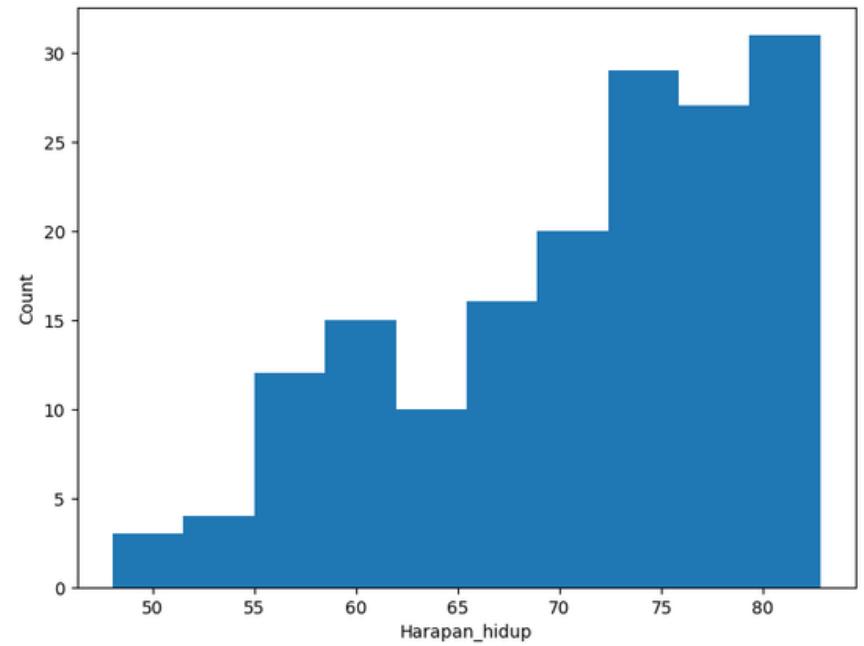
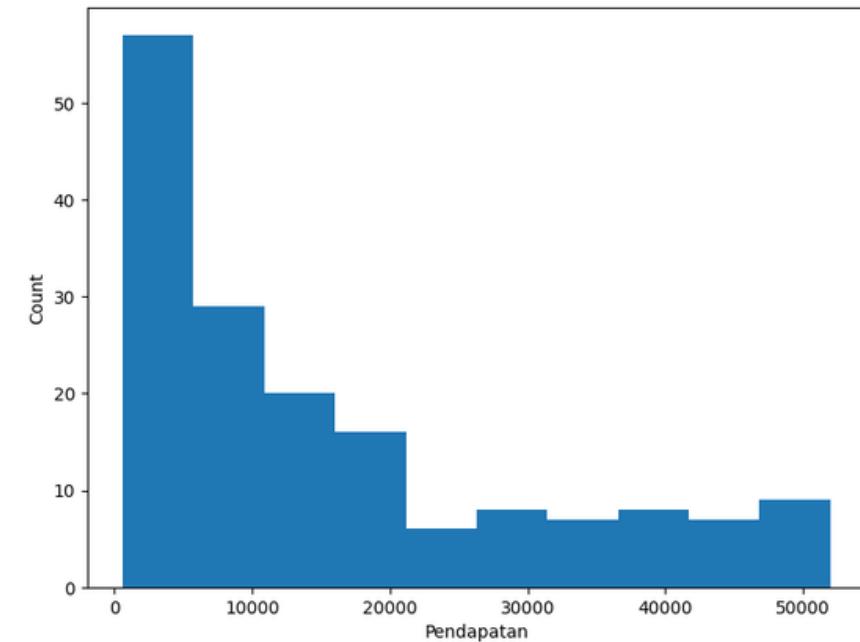
```
plt.figure(figsize=(8, 6))
plt.hist(df_outlier_handled['Pendapatan'], bins=10)
plt.xlabel('Pendapatan')
plt.ylabel('Count')
plt.show()

plt.figure(figsize=(8, 6))
plt.hist(df_outlier_handled['Harapan_hidup'], bins=10)
plt.xlabel('Harapan_hidup')
plt.ylabel('Count')
plt.show()
```

Explanation

Univariate adalah analisis statistik yang melibatkan hanya satu variabel dalam satu waktu. Features yang saya gunakan adalah 'Pendapatan' dan 'Harapan_hidup'. Jenis visualisasi yang saya gunakan adalah histogram, dimana saya membagi histogram ke 10 bagian. Saya menggunakan histogram karena datanya bersifat angka semua. Hasil yang bisa dilihat adalah untuk 'Pendapatan' paling banyak ada di range sekitar 0-5000. Sedangkan 'Harapan_hidup' paling banyak di ranges sekitar 80-83.

Result



EDA 2: BIVARIATE

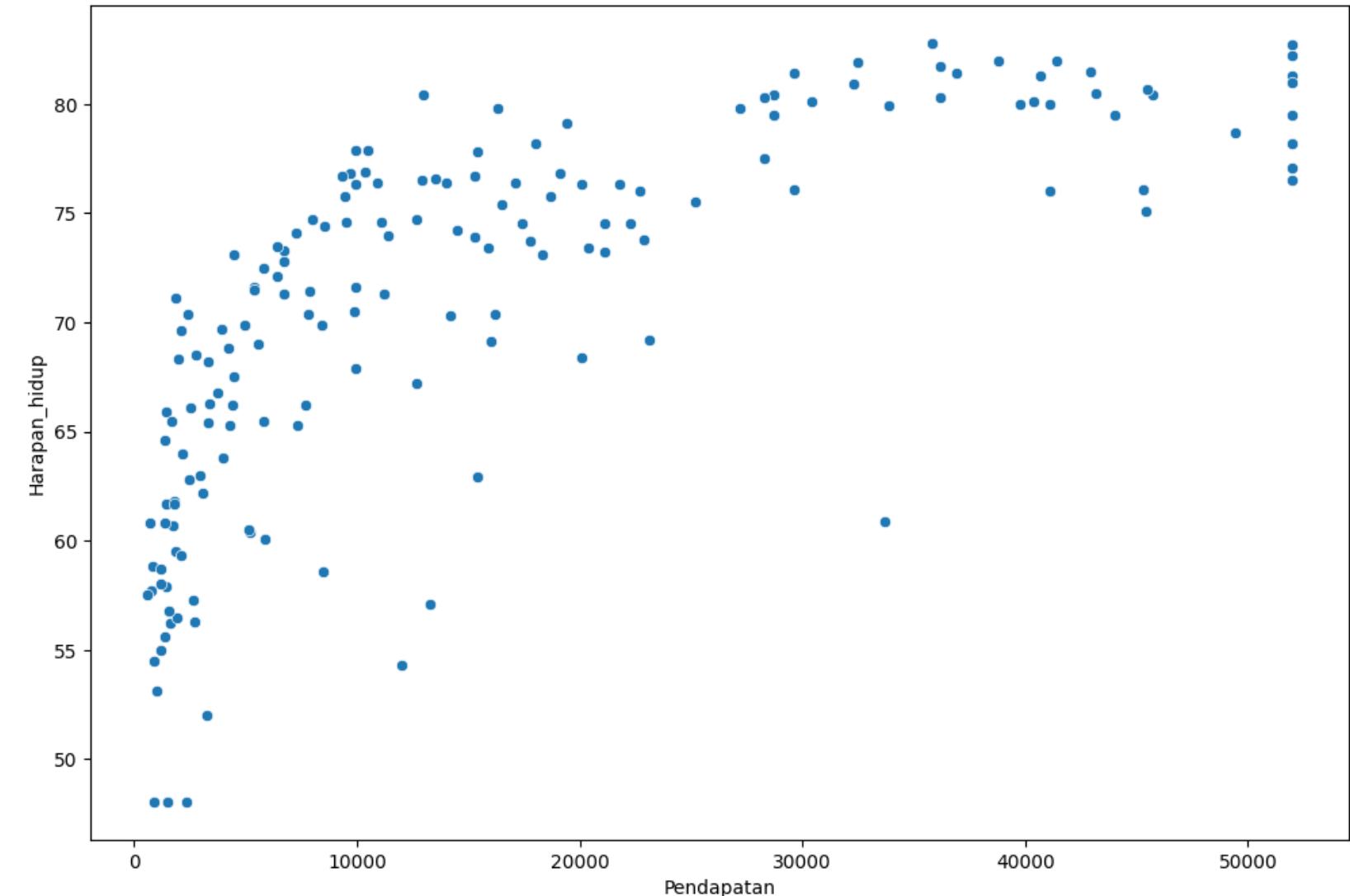
Code

```
plt.figure(figsize=(12,8))
sns.scatterplot(data=df_outlier_handled, x='Pendapatan', y='Harapan_hidup')
plt.xlabel('Pendapatan')
plt.ylabel('Harapan_hidup')
plt.show()
```

Explanation

Bivariate adalah analisis statistik yang melibatkan dua variabel dalam satu waktu. Features yang saya gunakan adalah 'Pendapatan' dan 'Harapan_hidup'. Jenis visualisasi yang saya gunakan adalah *scatterplot*, karena saya ingin melihat korelasi antara kedua *features* tersebut. Hasil dari *scatterplot* titiknya tersebar dari arah kiri bawah ke kanan atas, yang berarti kedua *features* ini memiliki korelasi yang positif, dimana semakin besar pendapatan, semakin besar pula harapan hidupnya.

Result



CLUSTERING: PART 1

1. Scale the Data

```
from sklearn.preprocessing import StandardScaler  
sc = StandardScaler()  
dfoutlier_std = sc.fit_transform(df_outlier_handled.astype(float))
```

Scaling data dilakukan untuk mengubah rentang nilai *features* dalam dataset sehingga memiliki skala yang serupa atau mendekati skala yang sama. Disini saya menggunakan **Standard Scaling (Z-Score Normalization)**, hal ini dilakukan dengan menggeser data sehingga memiliki **mean 0 dan deviasi standar 1**.

Explanation

Untuk menentukan jumlah cluster yang akan dibuat saya menggunakan metode:

- a. **Elbow Method** > menunjukkan belokan siku yang cukup tajam pada angka 3
- b. **Silhouette Method** > menunjukkan titik tertinggi di angka 3

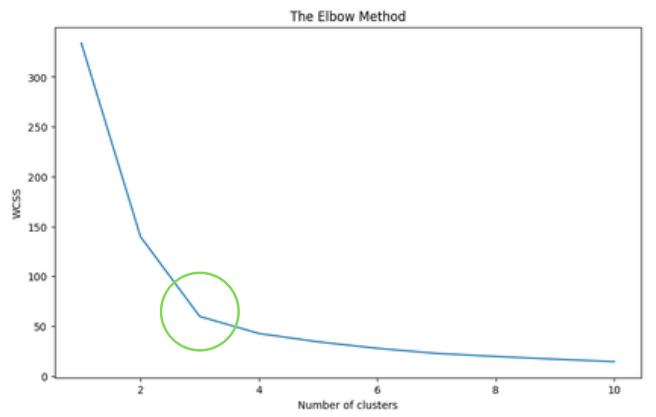
Oleh karena itu **Jumlah cluster** yang akan saya gunakan berdasarkan kedua metode ini yaitu **3**. Sehingga ketika divisualisasi lebih mudah dimengerti.

Kemudian data yang saya gunakan adalah yang sudah *discaling*.

2. Choose Number of Cluster

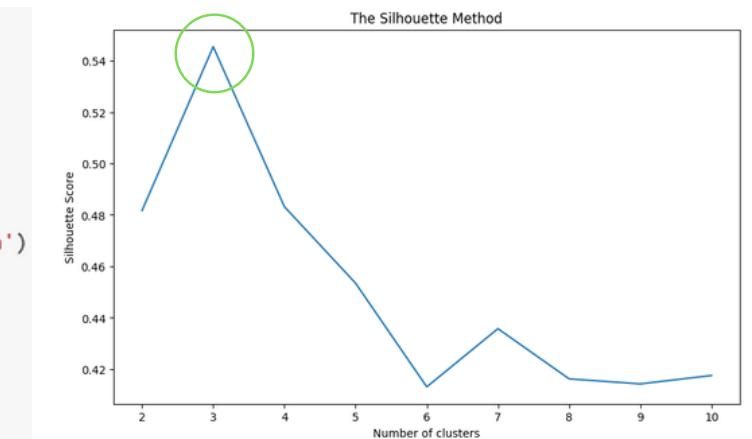
Elbow Method

```
from sklearn.cluster import KMeans  
  
plt.figure(figsize=(10,6))  
wcss = []  
for i in range(1, 11):  
    kmeans = KMeans(n_clusters = i, init='k-means++', random_state = 42)  
    kmeans.fit(dfoutlier_std)  
    wcss.append(kmeans.inertia_)  
  
plt.plot(range(1, 11), wcss)  
plt.title('The Elbow Method')  
plt.xlabel('Number of clusters')  
plt.ylabel('WCSS')  
plt.show()
```



Silhouette Method

```
from sklearn.metrics import silhouette_score  
  
plt.figure(figsize=(10, 6))  
silhouette_scores = []  
for k in range(2, 11):  
    kmeans = KMeans(n_clusters = k, init='k-means++', random_state=42)  
    kmeans.fit(dfoutlier_std)  
    labels = kmeans.labels_  
    silhouette_avg = silhouette_score(dfoutlier_std, labels, metric = 'euclidean')  
    silhouette_scores.append(silhouette_avg)  
  
plt.plot(range(2, 11), silhouette_scores)  
plt.title('The Silhouette Method')  
plt.xlabel('Number of clusters')  
plt.ylabel('Silhouette Score')  
plt.show()
```



CLUSTERING: PART 2

3. Do Clustering

```
kmeans2 = KMeans(n_clusters=3, random_state=42).fit(df_outlier_handled)
labels2 = kmeans2.labels_

new_dfoutlier = pd.DataFrame(data=df_outlier_handled, columns=['Pendapatan', 'Harapan_hidup'])
new_dfoutlier['label_kmeans2'] = labels2

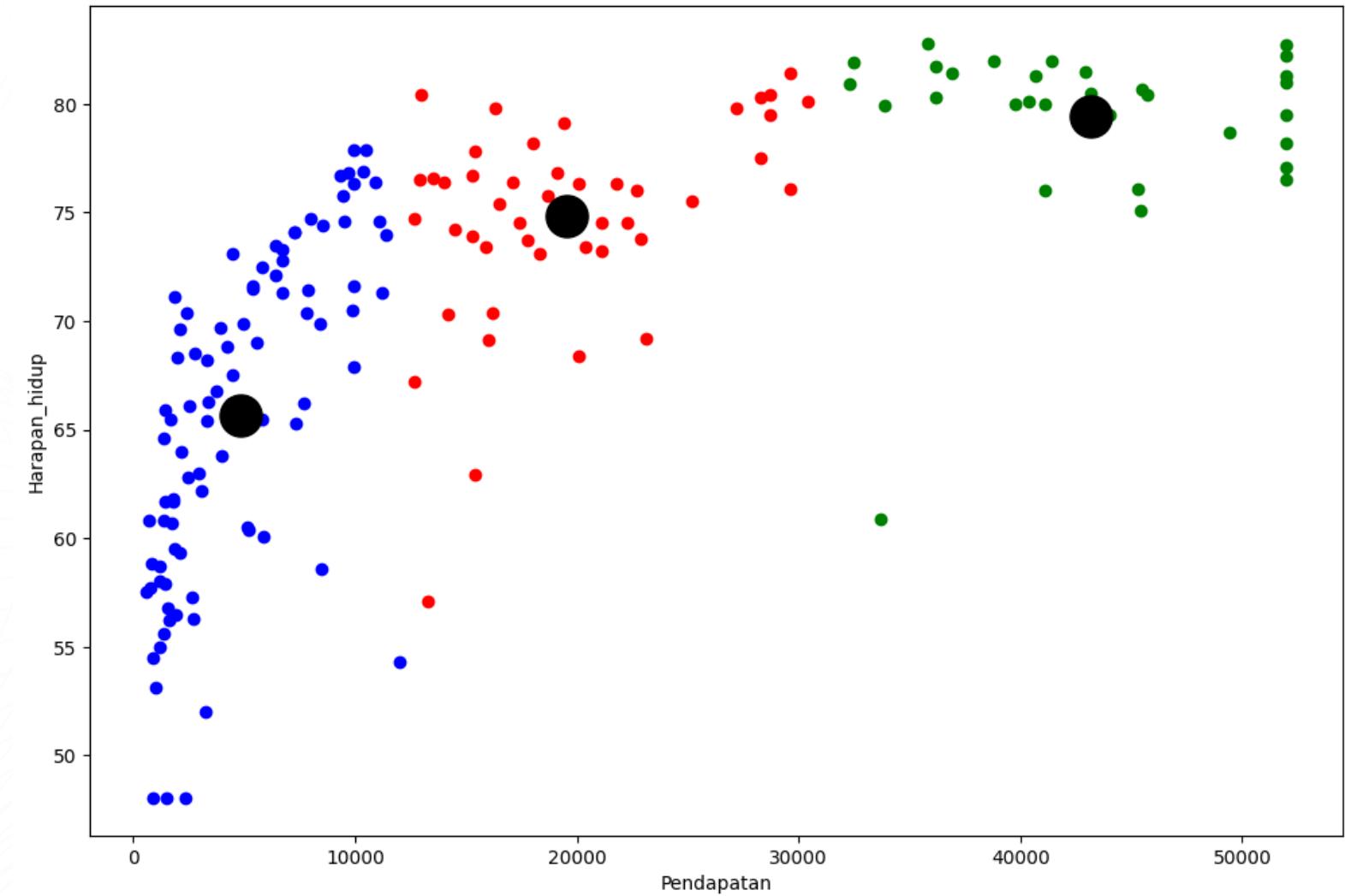
plt.figure(figsize=(12,8))
plt.scatter(new_dfoutlier.Pendapatan[new_dfoutlier.label_kmeans2 == 0],
new_dfoutlier.Harapan_hidup[new_dfoutlier.label_kmeans2==0], c='blue')
plt.scatter(new_dfoutlier.Pendapatan[new_dfoutlier.label_kmeans2 == 1],
new_dfoutlier.Harapan_hidup[new_dfoutlier.label_kmeans2==1], c='red')
plt.scatter(new_dfoutlier.Pendapatan[new_dfoutlier.label_kmeans2 == 2],
new_dfoutlier.Harapan_hidup[new_dfoutlier.label_kmeans2==2], c='green')

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

Explanation

Setelah menemukan jumlah cluster, maka saya melakukan clustering dengan scatter menggunakan matplotlib. Saya menggunakan Dataframe *df_outlier_handled*, agar data yang ditampilkan adalah data yang asli, bukan hasil scaling (yang ada angka minusnya). Jadi data yang ditampilkan bisa sesuai dengan data yang diberikan.

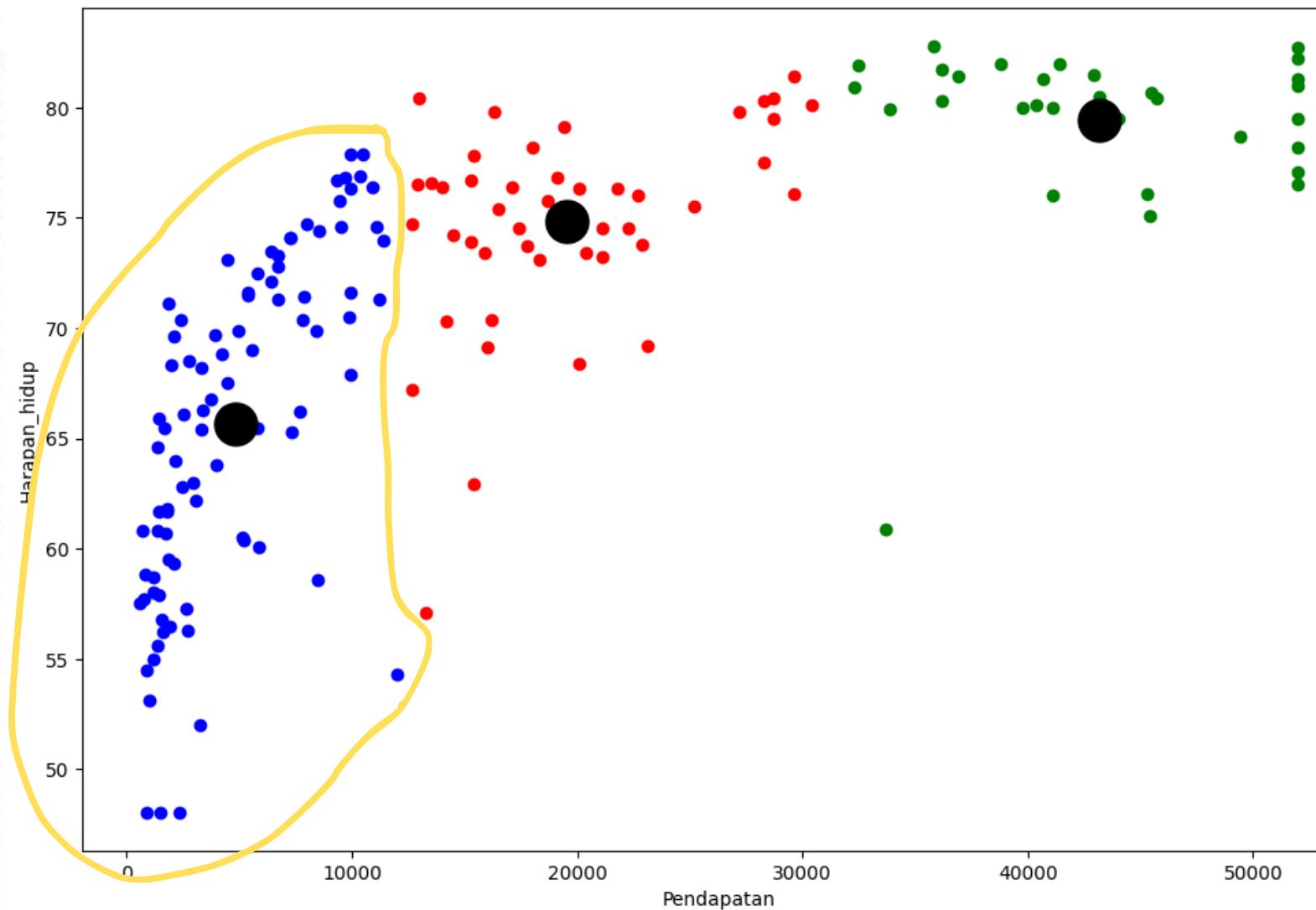
4. Result



Dapat dilihat terbentuk 3 kelompok yang memiliki perbedaan warna (biru, merah, hijau). Kemudian ada titik hitam juga yang memperjelas masing-masing cluster. Jadi sekarang dapat dilihat dengan lebih jelas perbedaan setiap cluster berdasarkan Pendapatan dan Harapan_hidupnya.

RECOMMENDATION

Country Cluster to Focus



Reason

Cluster yang ingin saya fokuskan yaitu cluster yang **berwarna biru**, atau cluster dengan **nilai 0**. Alasannya yaitu karena cluster yang berwarna biru berada di **posisi yang Harapan_hidupnya paling rendah dan Pendapatan paling rendah**. Jadi dapat dilihat negara yang berada di cluster biru adalah negara yang memiliki pendapatan dan harapan hidup sama-sama rendah. Jadi dari segi sosial ekonomi dan kesehatan sama-sama kurang.

RECOMMENDATION

Countries Included in the Cluster

```
new_dfoutlier['Negara']=df['Negara']
new_dfoutlier['Cluster'] = labels2

cluster_0_indices = new_dfoutlier.index[new_dfoutlier.label_kmeans2 == 0]

print("Countries in Cluster 0 (Blue):")
for i, index in enumerate(cluster_0_indices):
    if i == len(cluster_0_indices) - 1:
        print(new_dfoutlier['Negara'][index])
    else:
        print(new_dfoutlier['Negara'][index], end=", ")

count_cluster_0 = len(new_dfoutlier[new_dfoutlier.label_kmeans2 == 0])
print('Total ada', count_cluster_0, 'negara di cluster 0')
```

Result

Countries in Cluster 0 (Blue):

Afghanistan, Albania, Angola, Armenia, Bangladesh, Belize, Benin, Bhutan, Bolivia, Bosnia and Herzegovina, Burkina Faso, Burundi, Cambodia, Cameroon, Cape Verde, Central African Republic, Chad, China, Colombia, Comoros, Congo, Dem. Rep., Congo, Rep., Cote d'Ivoire, Dominican Republic, Ecuador, Egypt, El Salvador, Eritrea, Fiji, Gambia, Georgia, Ghana, Grenada, Guatemala, Guinea, Guinea-Bissau, Guyana, Haiti, India, Indonesia, Jamaica, Jordan, Kenya, Kiribati, Kyrgyz Republic, Lao, Lesotho, Liberia, Macedonia, FYR, Madagascar, Malawi, Maldives, Mali, Mauritania, Micronesia, Fed. Sts., Moldova, Mongolia, Morocco, Mozambique, Myanmar, Namibia, Nepal, Niger, Nigeria, Pakistan, Paraguay, Peru, Philippines, Rwanda, Samoa, Senegal, Sierra Leone, Solomon Islands, South Africa, Sri Lanka, St. Vincent and the Grenadines, Sudan, Tajikistan, Tanzania, Timor-Leste, Togo, Tonga, Tunisia, Turkmenistan, Uganda, Ukraine, Uzbekistan, Vanuatu, Vietnam, Yemen, Zambia.

Explanation

Untuk mengetahui negara apa saja yang ada pada cluster tersebut, pertama saya menambahkan 'Negara' ke df yang baru (new_dfoutlier). Kemudian saya mencari yang clusternya tergolong 0 (warna biru). Setelah itu saya melakukan looping untuk menampilkan seluruh negara yang ada dalam cluster 0. Saya juga menambahkan kode tambahan agar setelah setiap negara (kecuali negara di index terakhir) tidak enter melainkan dipisah dengan tanda koma (,). Kemudian saya juga mencari jumlah negara pada cluster 0, yaitu total ada **91 negara**.

RECOMMENDATION

Chosen Country to Receive Help

```
sorted_indices = cluster_0_indices[new_dfoutlier.loc[cluster_0_indices, 'Pendapatan'].argsort()]
sorted_indices = sorted_indices[new_dfoutlier.loc[sorted_indices, 'Harapan_hidup'].argsort(kind='mergesort')]

top_3_countries = new_dfoutlier['Negara'][sorted_indices[:3]].tolist()
top_3_pendapatan = new_dfoutlier.loc[sorted_indices[:3], 'Pendapatan'].tolist()
top_3_harapan_hidup = new_dfoutlier.loc[sorted_indices[:3], 'Harapan_hidup'].tolist()

print("Top 3 Countries in Cluster 2 dengan Pendapatan dan Harapan_hidup terendah")
print(top_3_countries)
```

Explanation

Setelah mengetahui seluruh list negara yang ada di cluster 0, saya melakukan sorting untuk melihat negara mana dari cluster 0 yang memiliki nilai Pendapatan dan Harapan_hidup yang rendah. Dimana hasilnya adalah:

1. Central African Republic



2. Haiti



3. Lesotho



RECOMMENDATION

Reason I Chose Them

```
print("Top 3 Countries in Cluster 2 dengan Pendapatan dan Harapan_hidup terendah")
for i in range(3):
    print(f"Country: {top_3_countries[i]}")
    print(f"Pendapatan: {top_3_pendapatan[i]}")
    print(f"Harapan_hidup: {top_3_harapan_hidup[i]}")
    print()
```

Disini saya juga mencoba menampilkan nilai dari Pendapatan dan Harapan_hidup top 3 countries yang telah saya pilih untuk dibantu. Perlu dicatat DataFrame yang saya gunakan adalah yang outliernya sudah ditangani. Maka dari itu nilai dari harapan_hidup adalah nilai yang sudah diubah ke batas bawah dan bukan nilai asli.

1. Central African Republic



Pendapatan: 888.0

Harapan_hidup: 48.05

2. Haiti



Pendapatan: 1500.0

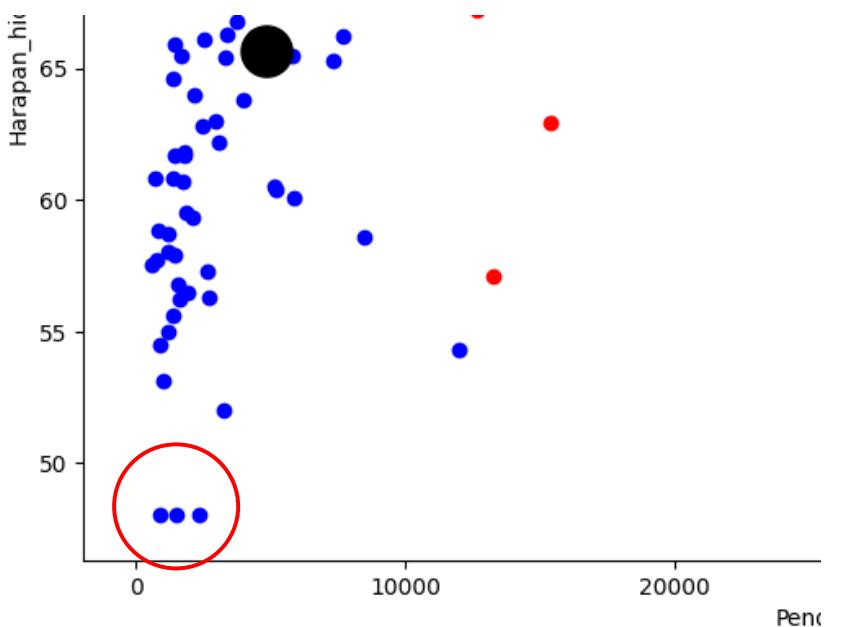
Harapan_hidup: 48.05

3. Lesotho



Pendapatan: 2380.0

Harapan_hidup: 48.05



RECOMMENDATION

Conclusion

Terdapat **3 negara** yang terpilih untuk menjadi prioritas untuk dibantu oleh HELP International, yaitu:

1. Central African Republic



Pendapatan: 888.0

Harapan_hidup: 48.05

2. Haiti



Pendapatan: 1500.0

Harapan_hidup: 48.05

3. Lesotho



Pendapatan: 2380.0

Harapan_hidup: 48.05

Hal ini dinilai dari kondisi krisis yang dialami oleh negara, terutama di bidang sosial ekonomi dan kesehatan. Dengan dana sejumlah \$10 juta, negara-negara ini akan diprioritaskan untuk menerima bantuan. Pengambilan keputusan ini tentu tidak secara asal atau acak, melainkan melewati beberapa proses pengolahan data menggunakan bantuan Python agar keputusan yang diambil lebih akurat dan relevan.



**THANKYOU FOR
READING**