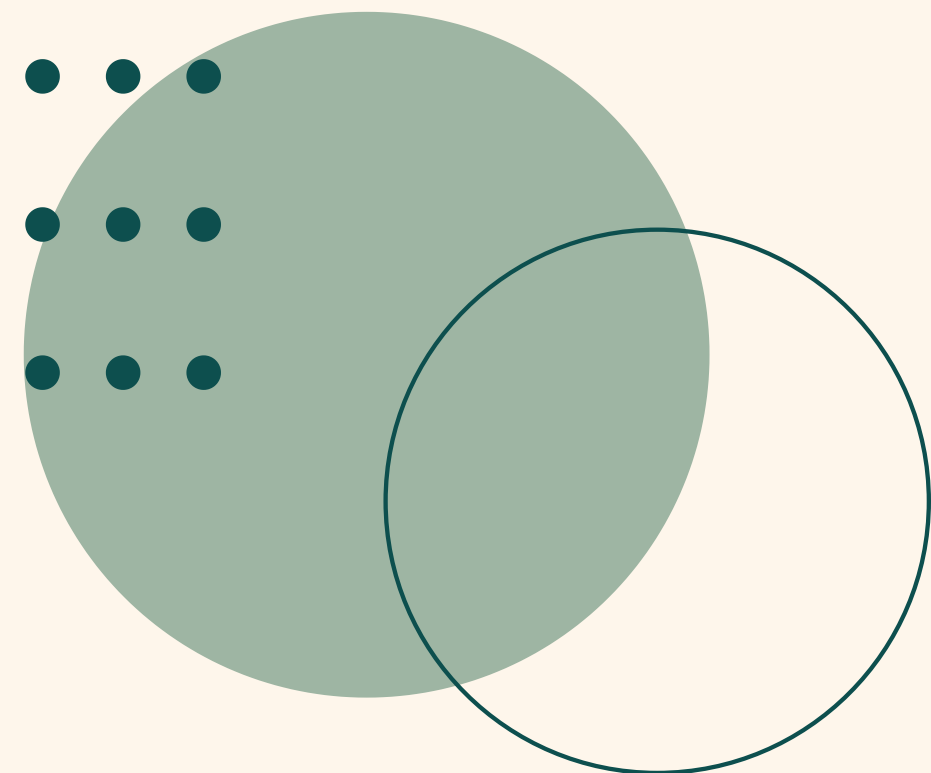
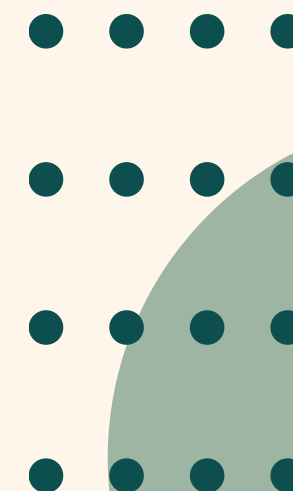


# FINAL PROJECT SANBERCODE BOOTCAMP

MIKHAEL KIRENIUS RANATA



# DAFTAR ISI

Background

Dataset

EDA

Data Cleaning

Clustering

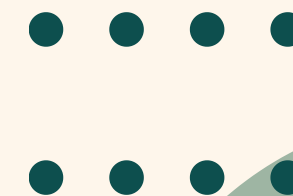
Hasil

Kesimpulan

# BACKGROUND

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.



# DATASET

## Dataset Info

Memiliki total row 167 dan tidak memiliki nilai null atau missing value didalamnya

## Dataset Features

Memiliki Column seperti Negara, Kematian anak, ekspor, kesehatan, impor, pendapatan, inflansi, harapan hidup, jumlah fertiliti, GDP perkapita.

01

02

03

## Dataset Data Type

Dataset ini memiliki data type object, float dan juga int.

# FEATURES SELECTION

01

Mengambil Column  
Pendapatan yang dimana  
pendapatan merupakan  
pendapatan setiap orang  
didalam negara.

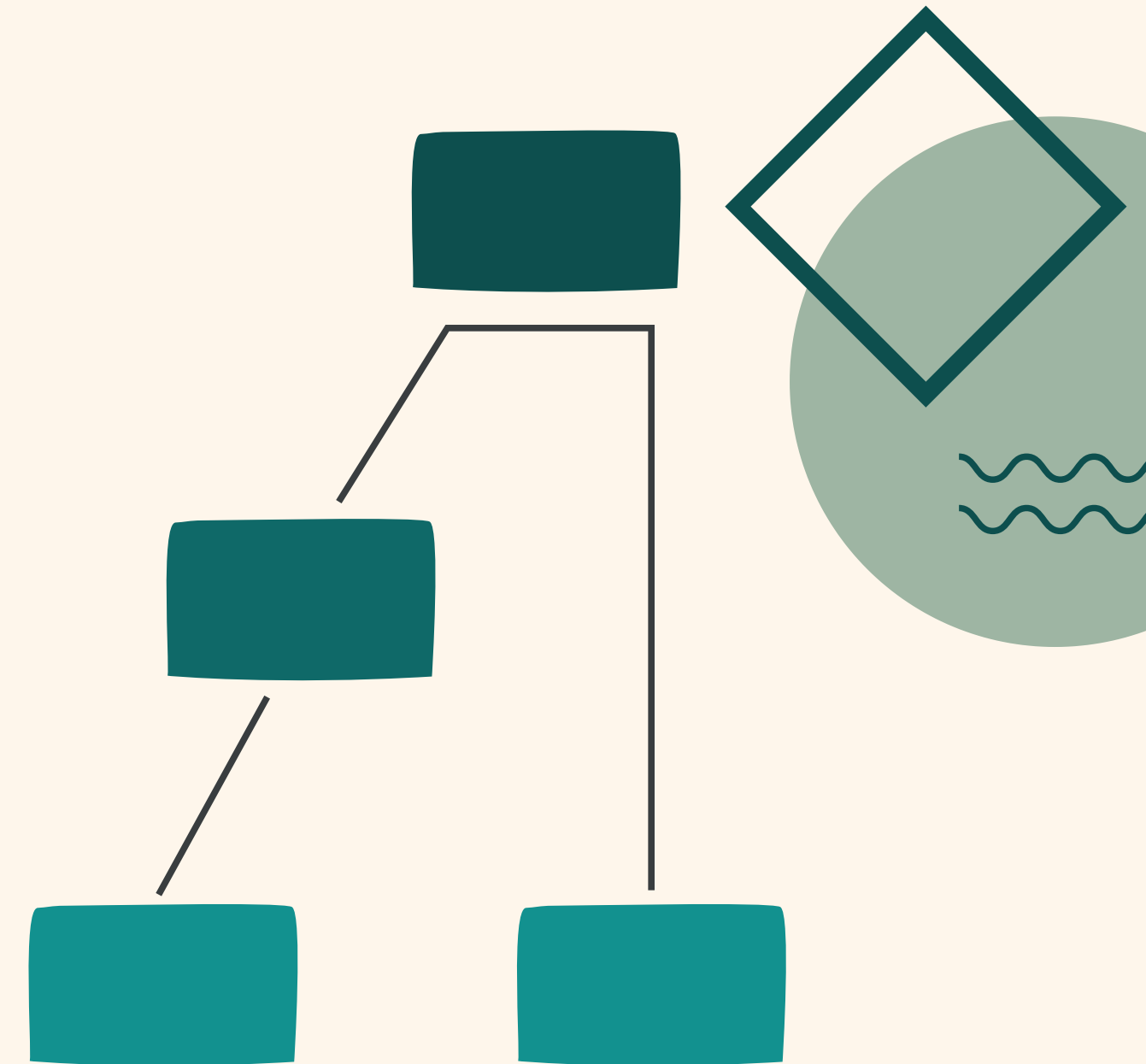
02

Mengambil column  
kematian anak yang dimana  
kematian anak merupakan  
jumlah kematian anak  
dibawah usia 5 tahun per  
1000 kelahiran.

# DATA CLEANING

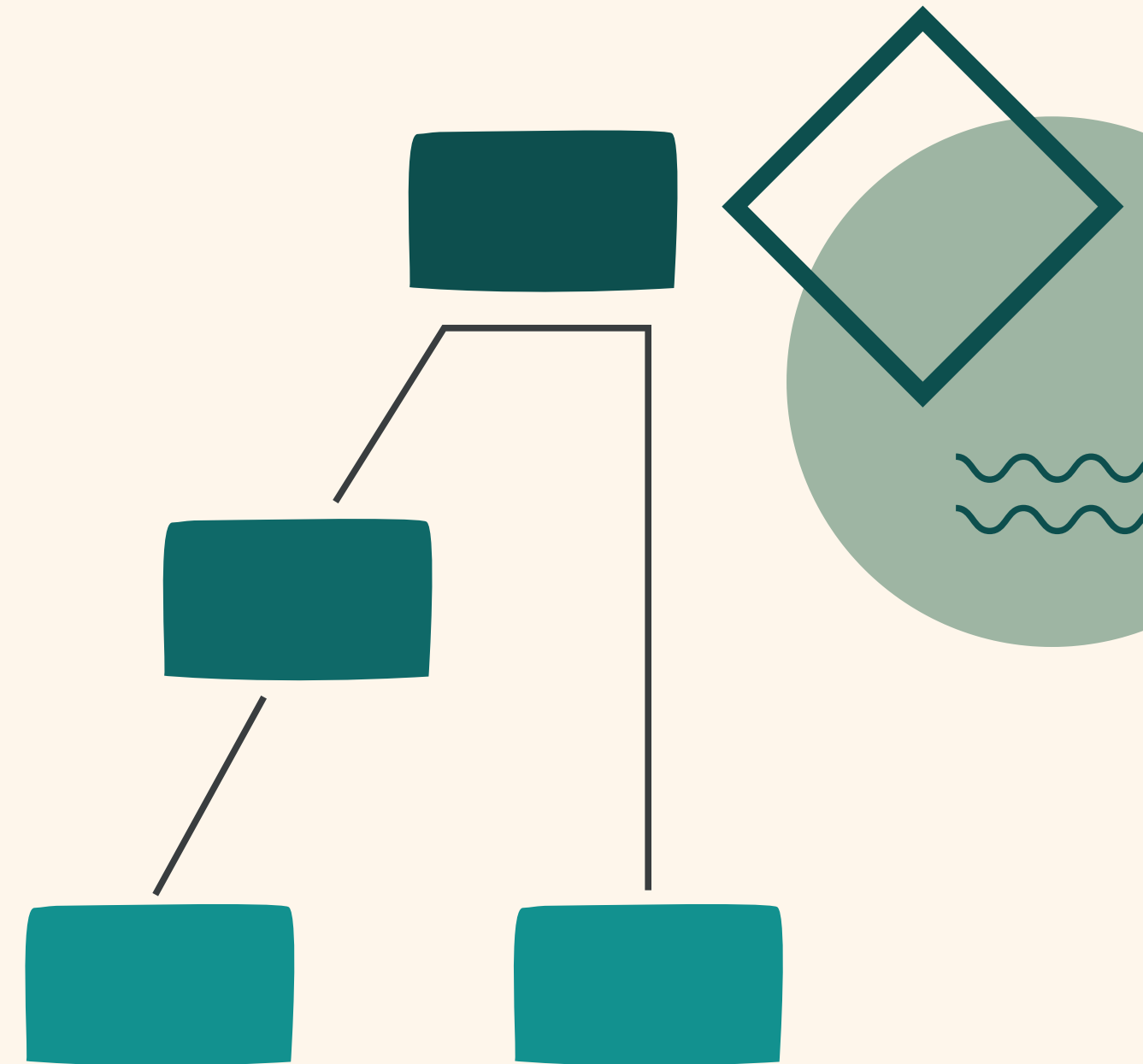
Melakukan pengecekan missing values pada dataset dengan menggunakan `df.info()` dan `df.describe()` untuk mendapatkan data dari dataset dengan lebih baik. sehingga dapat menghasilkan data seperti berikut

```
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
```

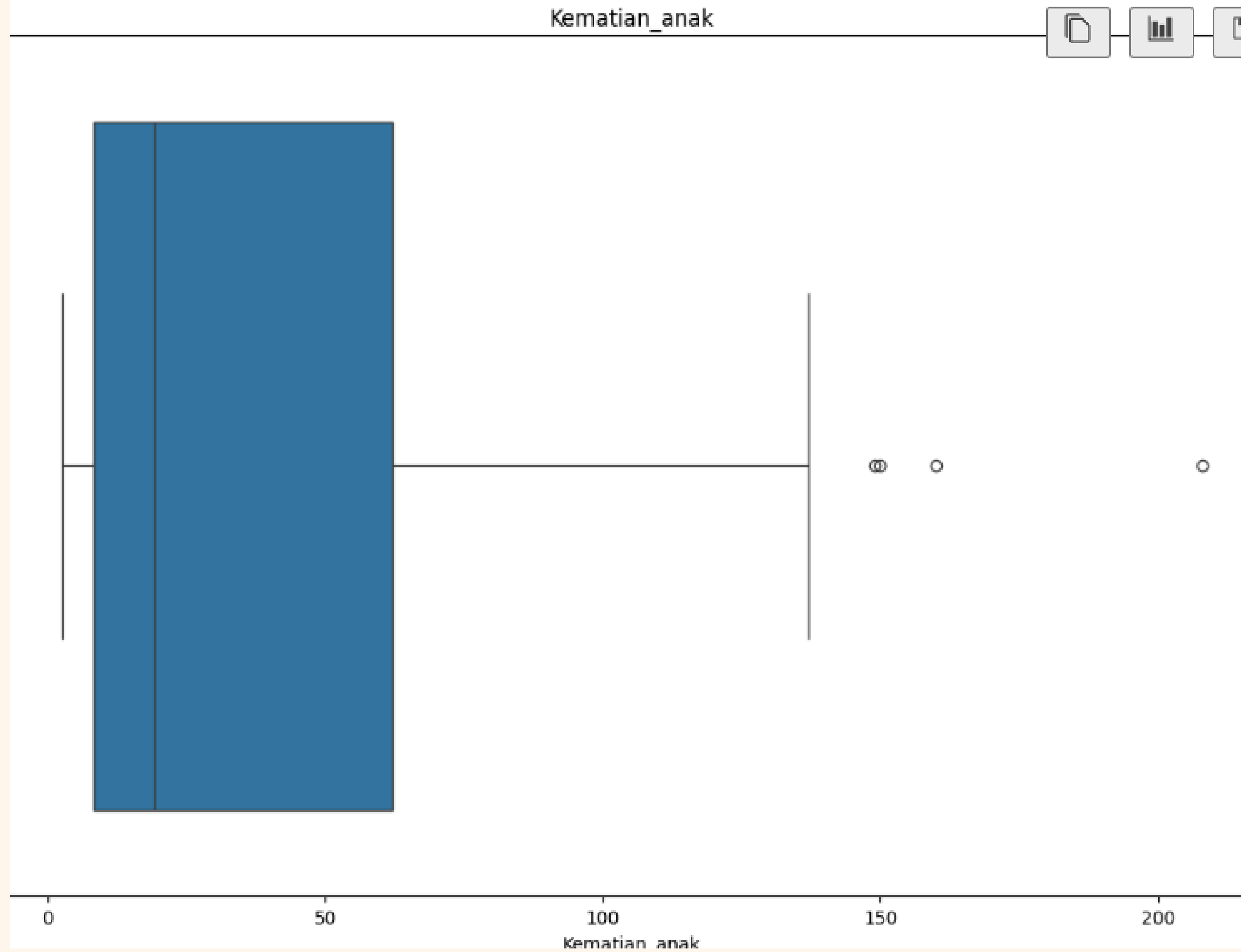


# DATA CLEANING

*Melakukan Pengecekan apakah terdapat outlier pada data yang telah dipilih didalam dataset. outlier adalah adanya titik data yang tidak sesuai dengan data lainnya seperti adanya nilai yang berbeda jauh dibandingkan data lainnya sehingga akan merusak analisis statistik dalam visualisasi.*

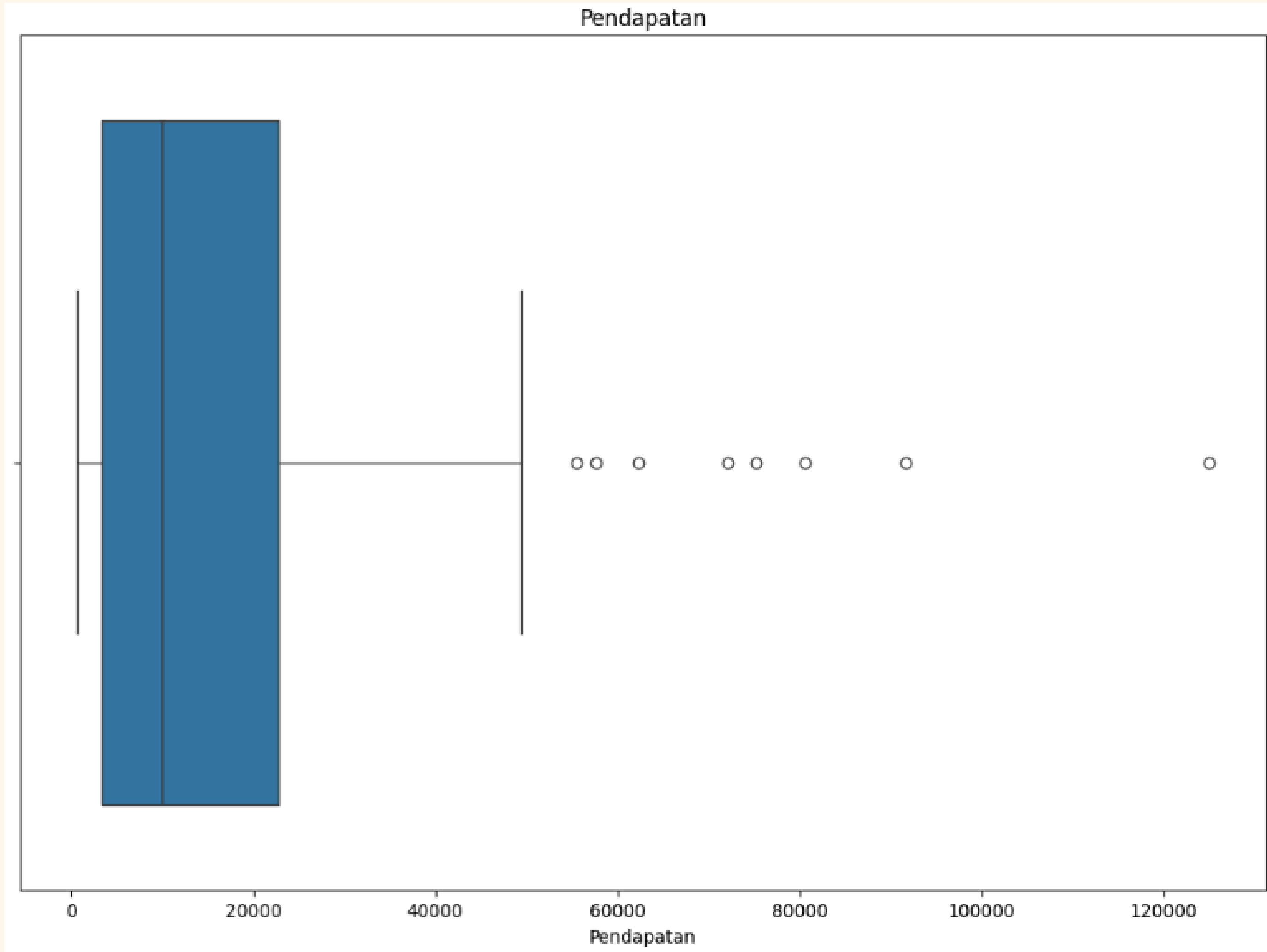


CHECK OUTLIER





CHECK OUTLIER

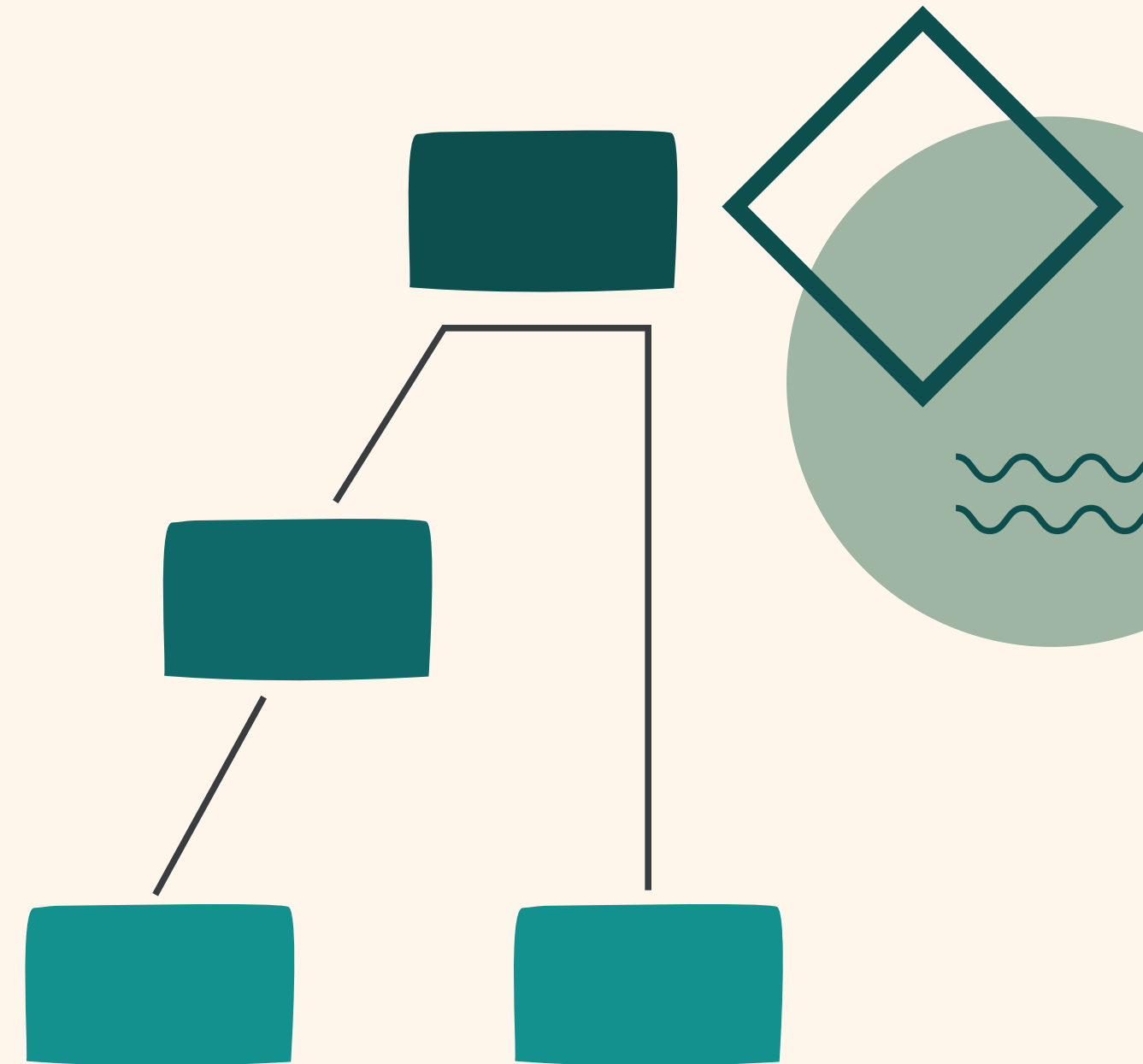


# HANDLING OUTLIER

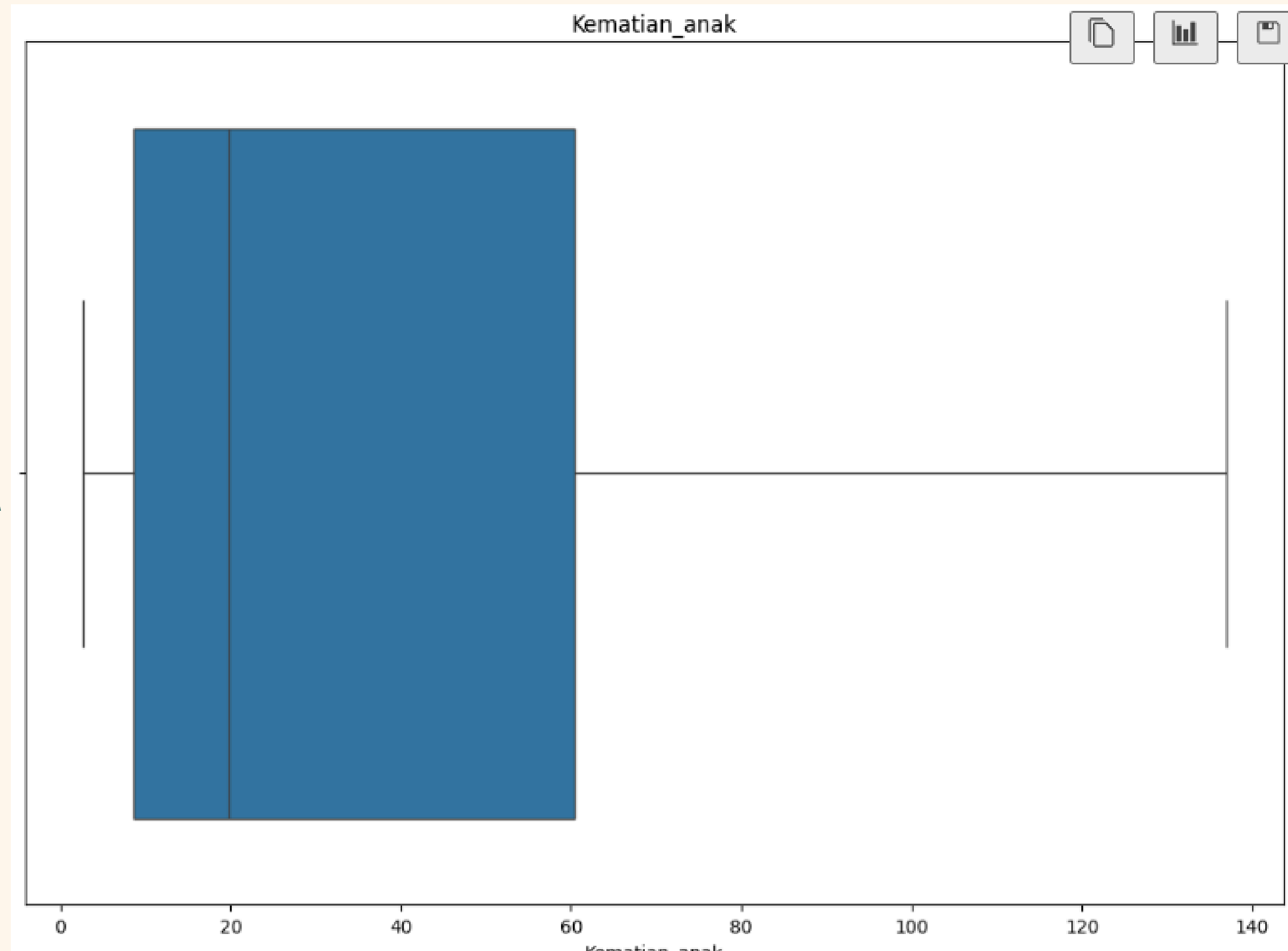
Melakukan Handling pada Outlier dengan Function Remove Outlier



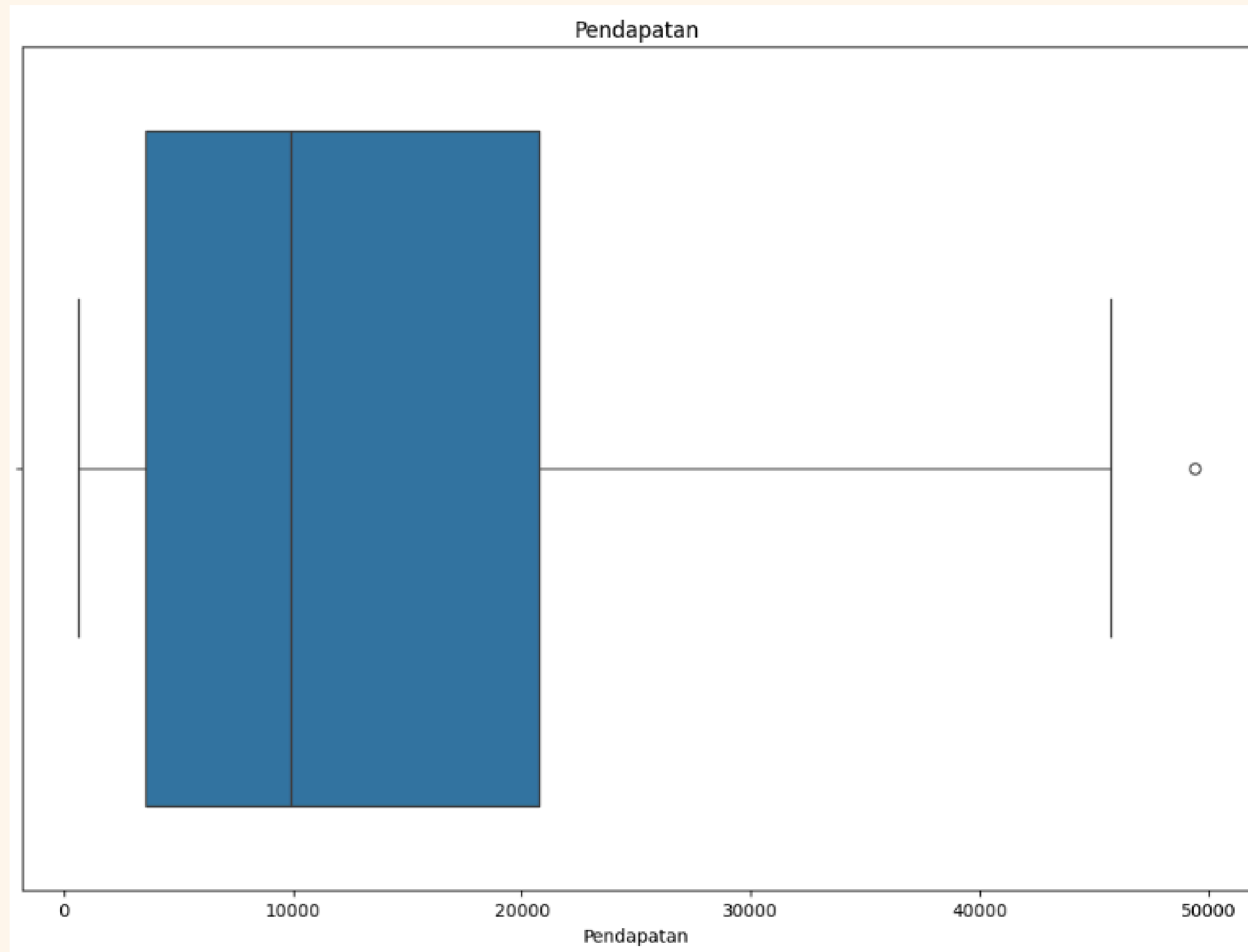
```
1 def remove_outlier(df):  
2     Q1 = df.quantile(0.25)  
3     Q3 = df.quantile(0.75)  
4     IQR = Q3 - Q1  
5     lower_bound = Q1 - 1.5 * IQR  
6     upper_bound = Q3 + 1.5 * IQR  
7     df = df[(df > lower_bound) & (df < upper_bound)]  
8     return df
```



# HANDLING OUTLIER

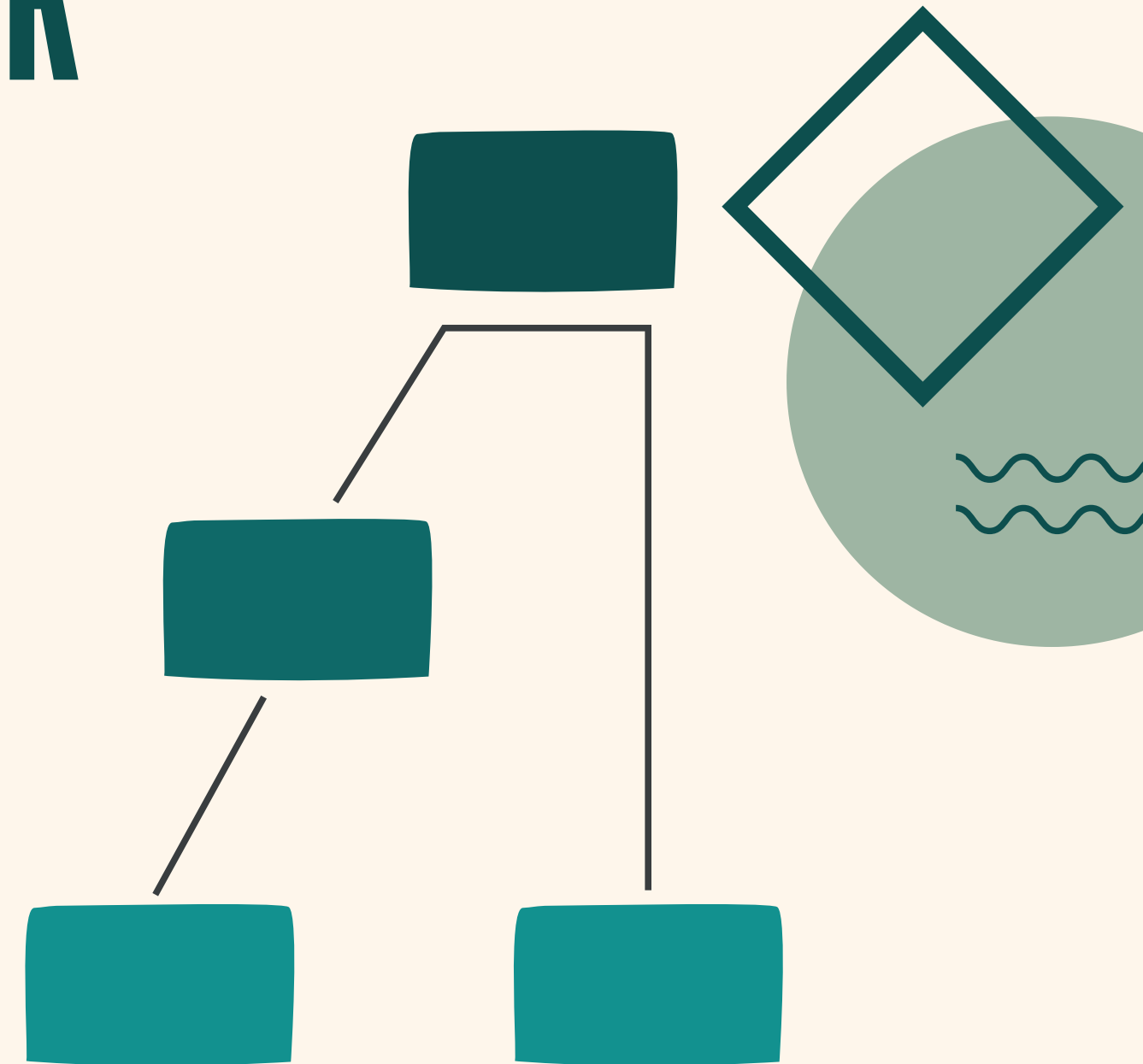


# HANDLING OUTLIER



# HANDLING OUTLIER

Handling outlier didalam dataset dengan menggunakan metode IQR, penggunaan metode ini adalah dengan cara menghitung Q1 dan Q3. Q1 merupakan kuartil bawah dari data, Q3 merupakan kuartil atas dari data. setelah mendapatkan nilai Q1 dan Q3 tahapan selanjutnya dalam metode ini adalah dengan menghitung IQR. jika data melebihi batas atas dan kurang dari batas bawah maka data akan dianggap sebagai outlier dan menghapus data tersebut dari dataframe.



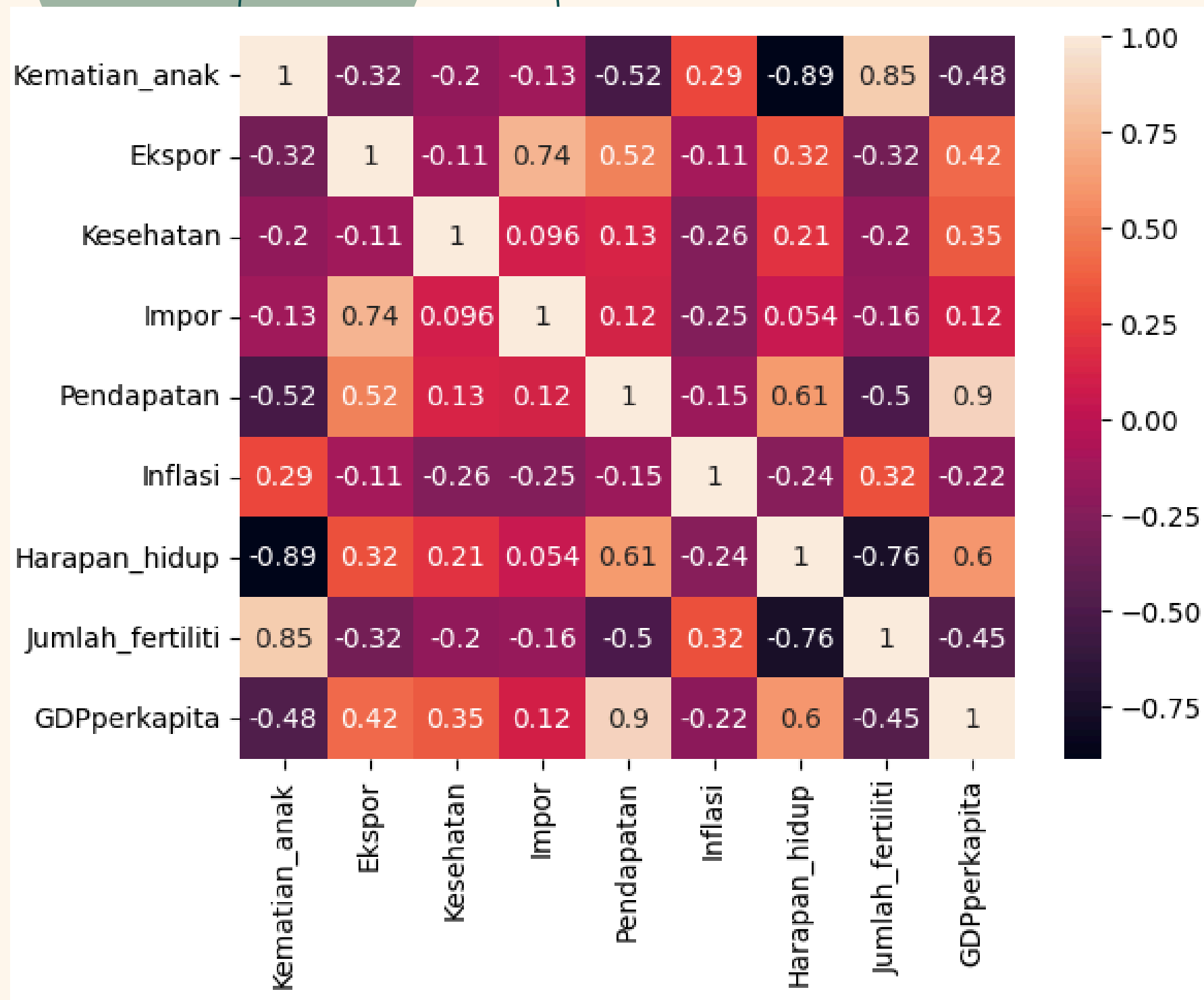
# UNIVARIATE ANALYSIS

## Describe

Menggunakan Function Describe terhadap dataset untuk mendapatkan informasi yang lebih detail mengenai nilai nilai dari setiap column seperti central tendency dan juga non null value.

	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

# BIVARIATE ANALYSIS



## Heatmap

Menggunakan heatmap sebagai visualisasi untuk mendapatkan korelasi antar data didalam dataset, penggunaan heatmap lebih mudah untuk dimengerti untuk mendapatkan nilai korelasi antar data.

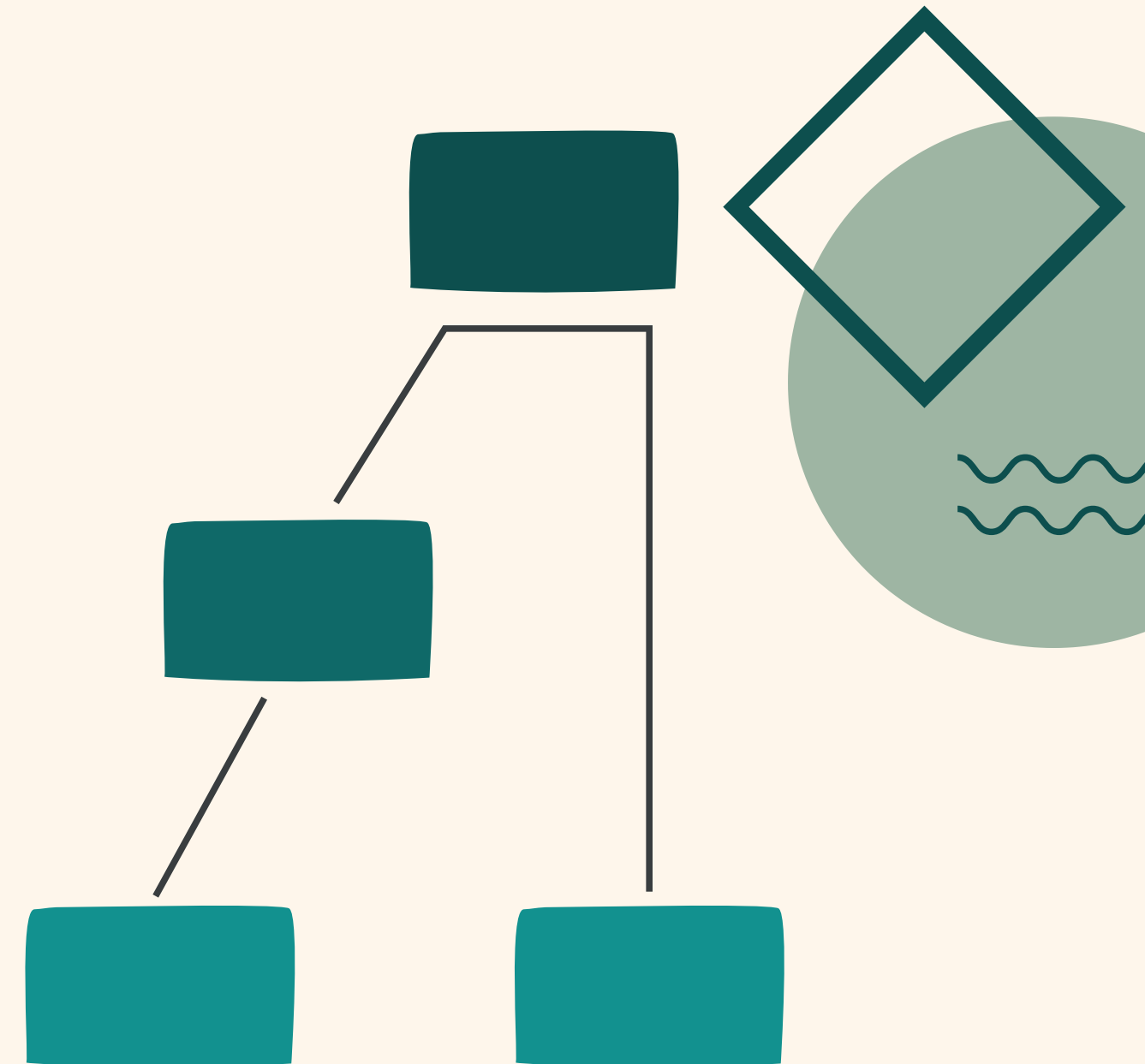
dari heatmap ini dapat didapatkan data korelasi antara pendapatan dan kematian anak memiliki nilai -0.52 sehingga data tersebut memiliki korelasi.

# SCALE DATA

Melakukan scaling data yang sudah dilakukan remove outlier sehingga data dapat dilakukan cluster.



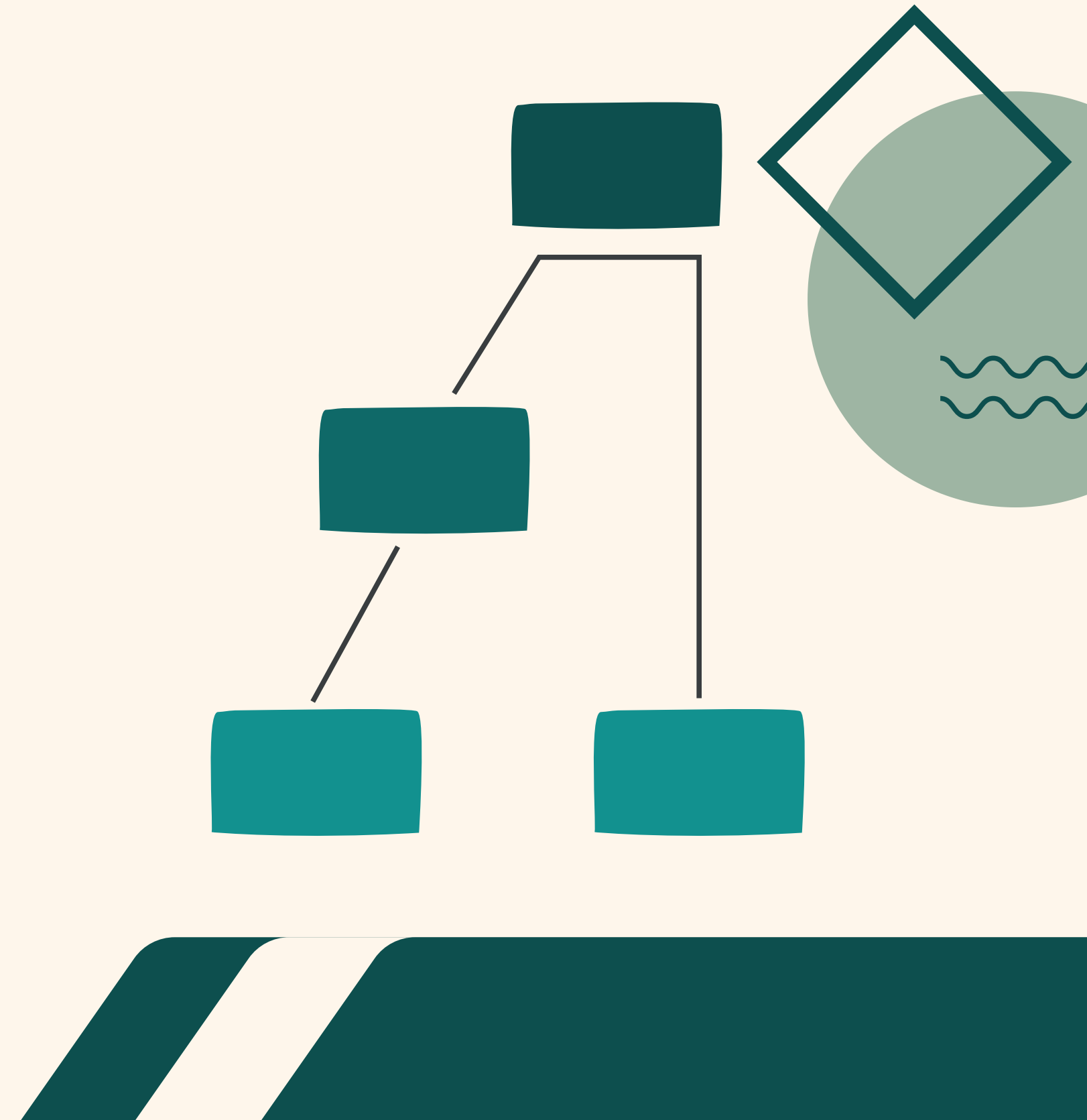
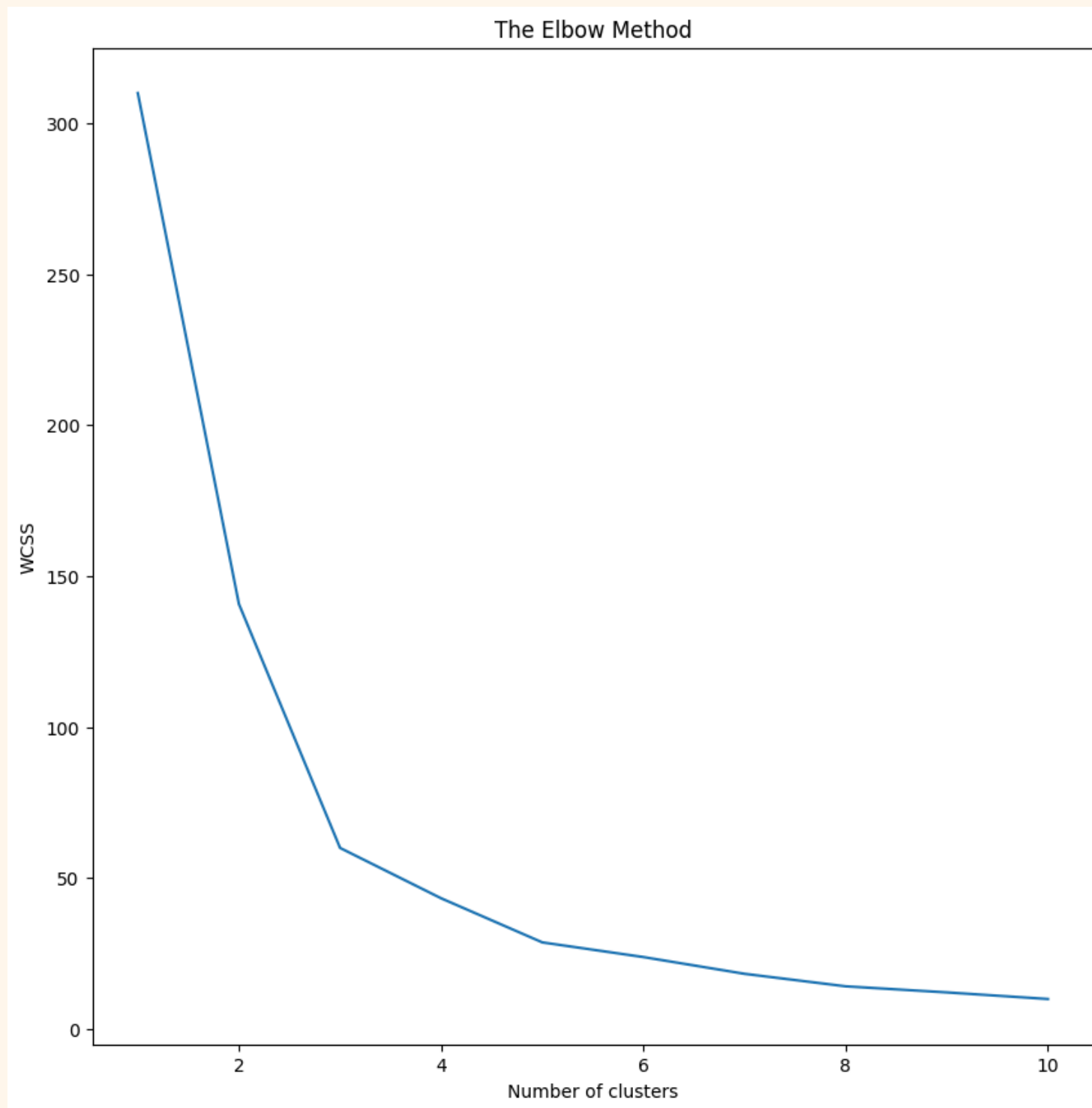
```
1 from sklearn.preprocessing import StandardScaler
2 from sklearn.cluster import KMeans
3
4 sc = StandardScaler()
5 df_outlierScaled = sc.fit_transform(df_outlier.astype(float))
```





# ELBOW METHOD

Menentukan nilai cluster dengan menggunakan elbow method.  
memilih nilai 3 karena nilai siku berada pada angka 3.



# CLUSTERING

Clustering dengan menggunakan K-Means



```
1 kmeans1 = KMeans(n_clusters=3, random_state=42).fit(df_outlierScaled)
2 labels1 = kmeans1.labels_
```

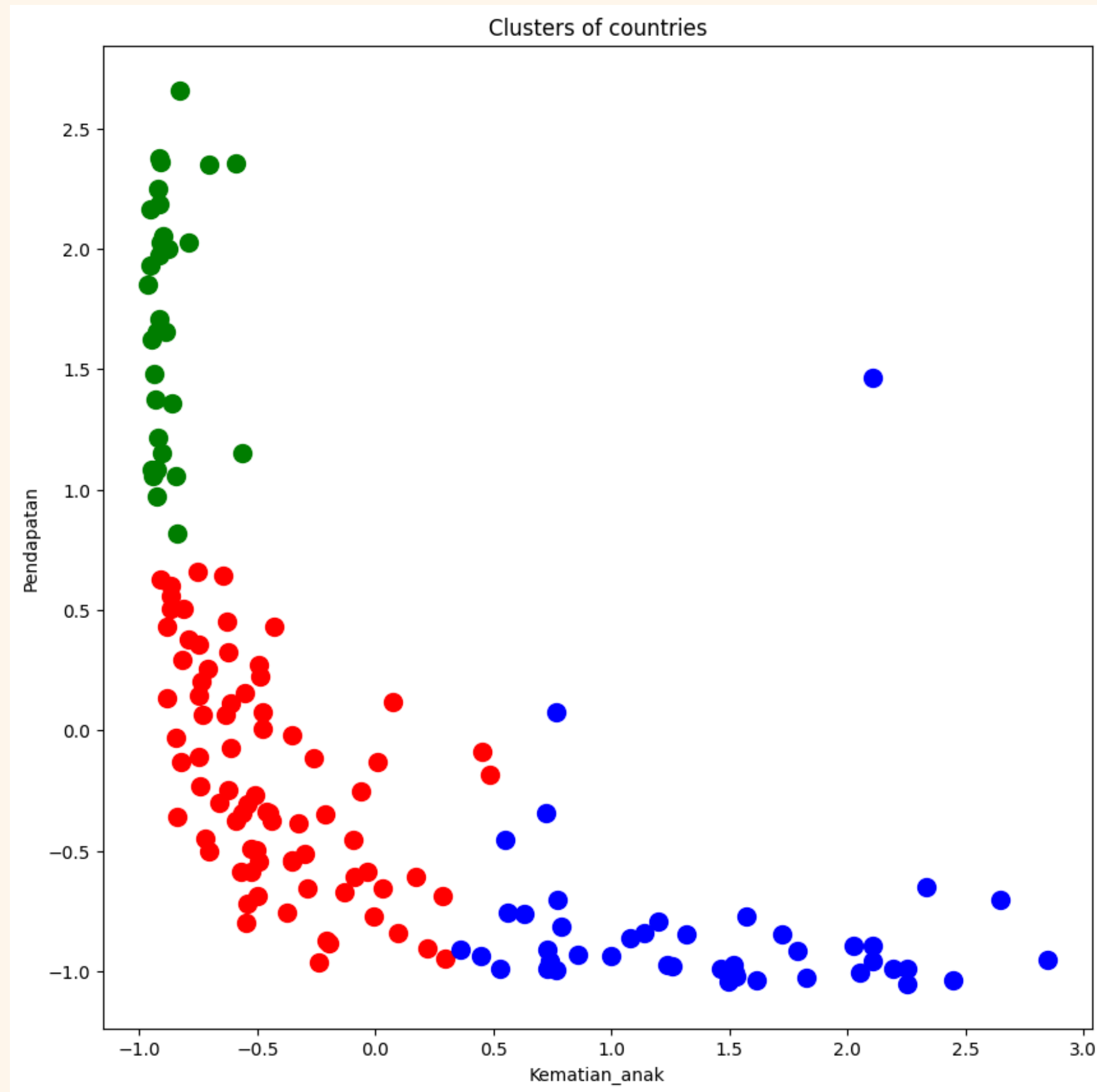


```
1 new_df = pd.DataFrame(data=df_outlierScaled, columns=['Kematian_anak', 'Pendapatan'])
2 new_df['label'] = labels1
```



```
1 plt.figure(figsize=(10, 10))
2 plt.scatter(new_df.Kematian_anak[new_df.label == 0], new_df.Pendapatan[new_df.label == 0], s = 100, c = 'red', label = 'Cluster 1')
3 plt.scatter(new_df.Kematian_anak[new_df.label == 1], new_df.Pendapatan[new_df.label == 1], s = 100, c = 'blue', label = 'Cluster 2')
4 plt.scatter(new_df.Kematian_anak[new_df.label == 2], new_df.Pendapatan[new_df.label == 2], s = 100, c = 'green', label = 'Cluster 3')
5 plt.title('Clusters of countries')
6 plt.xlabel('Kematian_anak')
7 plt.ylabel('Pendapatan')
8 plt.show()
```

# CLUSTERING



# CLUSTERING

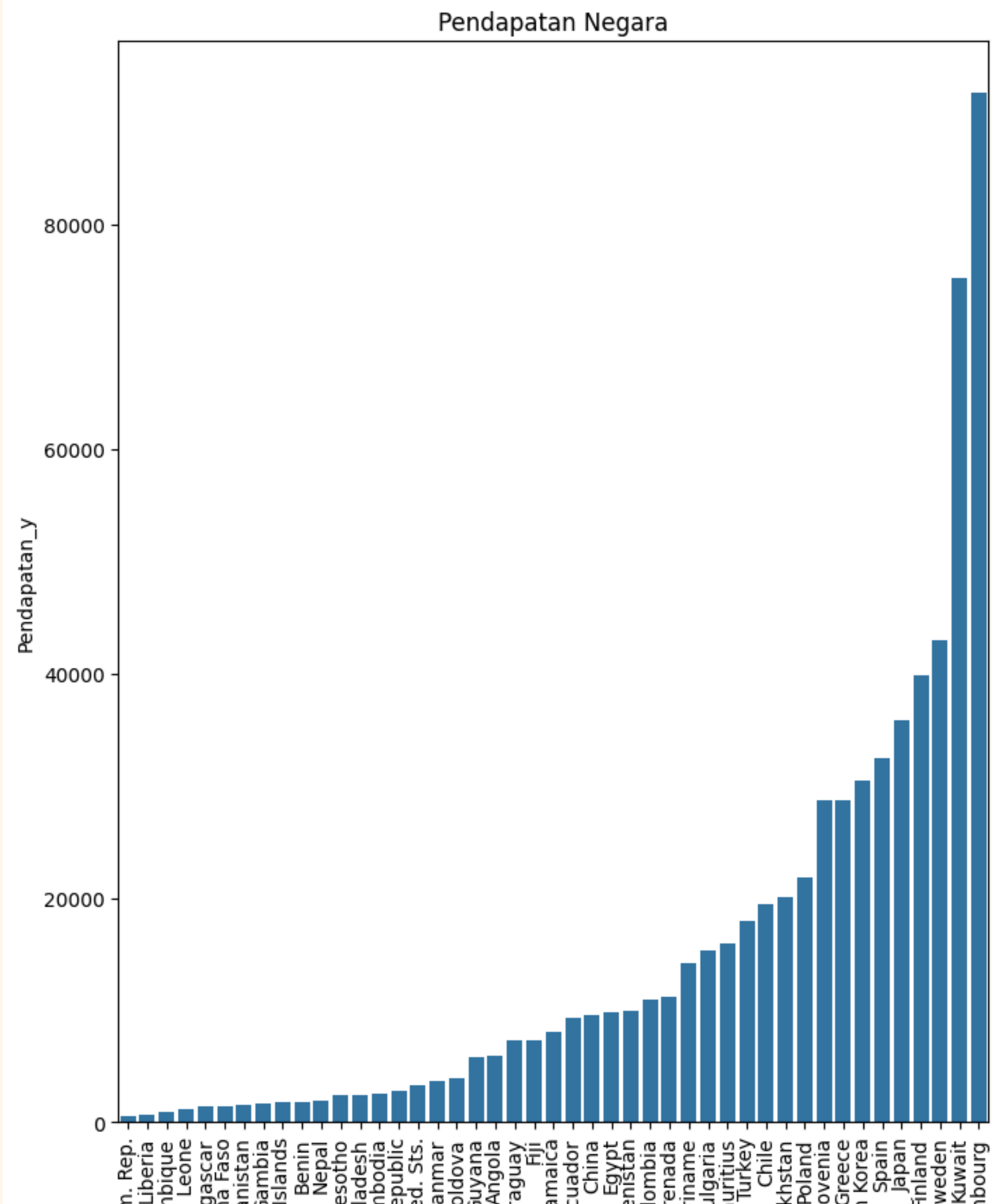
## Cluster Country

Cluster yang difokuskan adalah cluster dengan label 1 berwarna biru karena dari visualisasi sebelumnya. dapat disimpulkan bahwa semakin kecil pendapatan maka semakin besar angka kematian anak di negara tersebut.

## Cluster Country List

Afganistan  
Angola  
Bangladesh  
Benin  
Bulgaria  
Burkino Faso  
Cambodia  
Chile  
China

HASIL



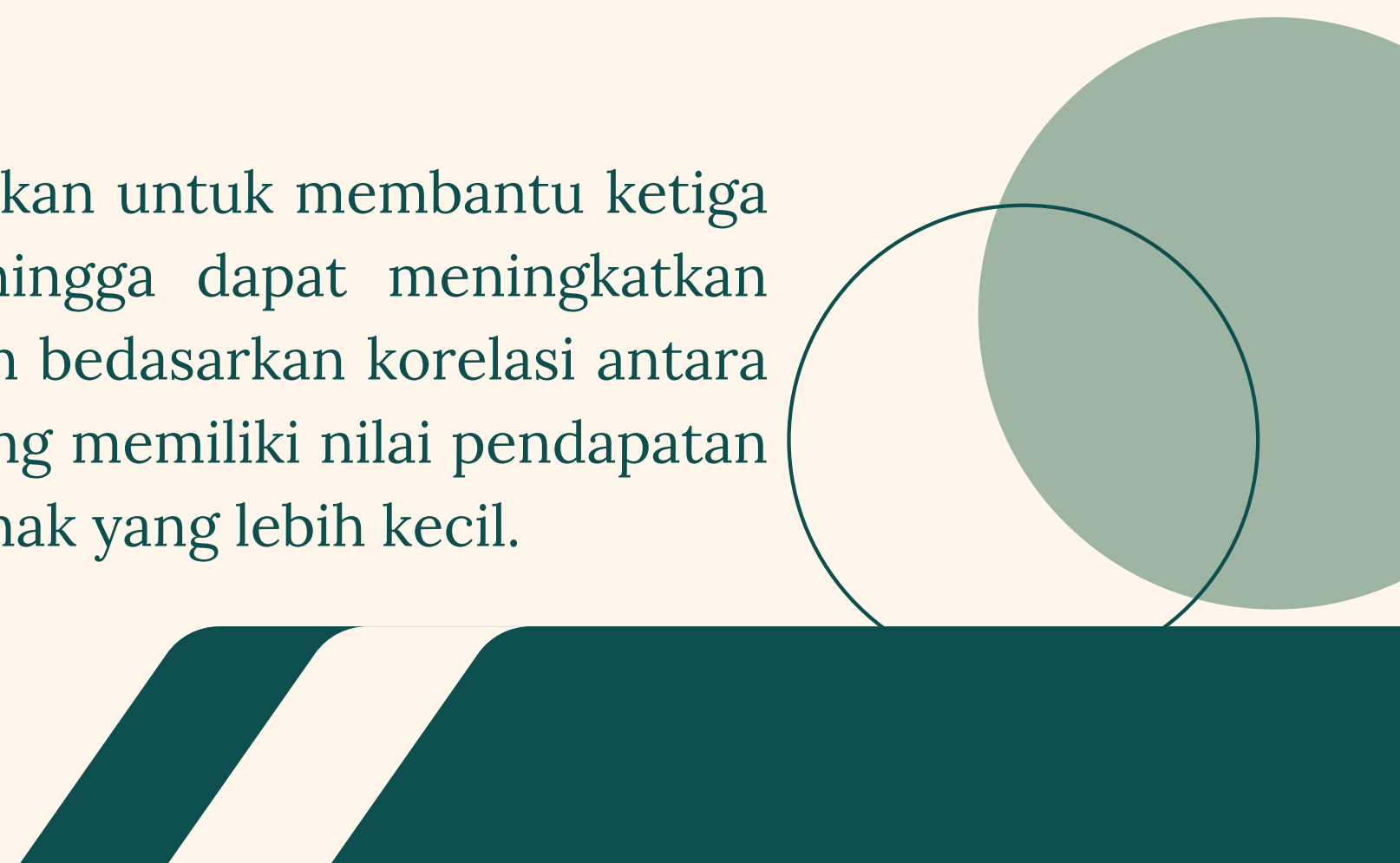


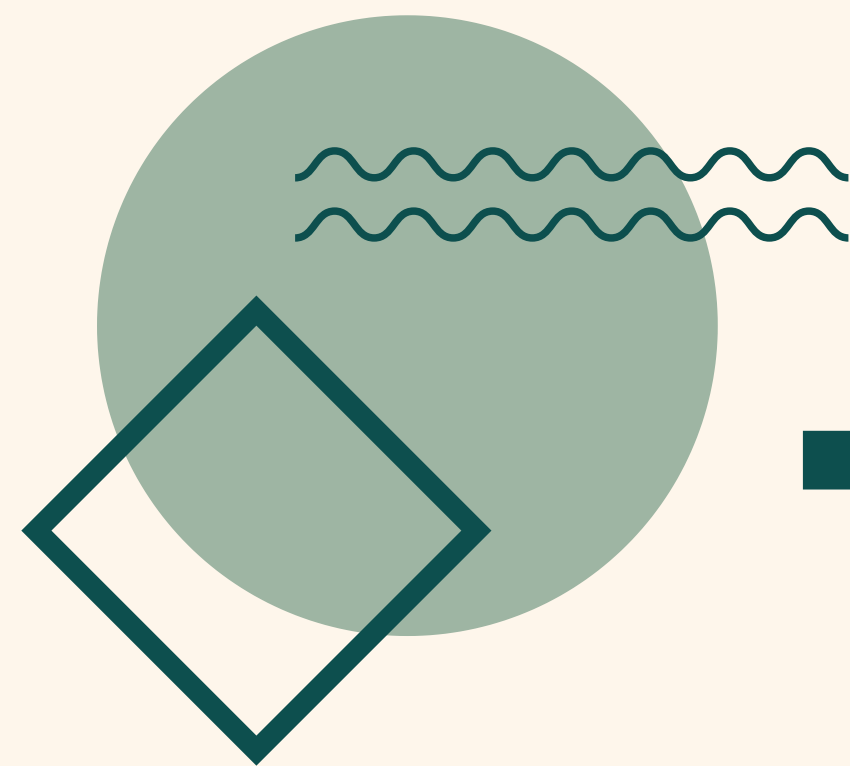
# KESIMPULAN

Bedasarkan Kesimpulan dari clustering mengambil data dari cluster berwarna biru dengan melakukan visualisasi pendapatan per orang. didapatkan bahwa pendapatan terendah dimiliki oleh negara

- Congo, Dem. Rep.
- Liberia
- Mozambique

Organisasi HELP International dapat memfokuskan untuk membantu ketiga negara ini untuk memerangi kemiskinan sehingga dapat meningkatkan pendapatan setiap orangnya. hal ini disimpulkan berdasarkan korelasi antara data pendapatan dan kematian anak. Negara yang memiliki nilai pendapatan lebih besar cenderung memiliki nilai kematian anak yang lebih kecil.





# TERIMA KASIH

