



FINAL PROJECT

NAMA: HAIKAL EFENDI

PROGRAM: DATA SCIENCE – PYTHON

SANBERCODE BATCH 26

A. Latar Belakang

Data Science merupakan ilmu yang sedang trend yang dibangun berdasarkan disiplin ilmu matematika, statistik, dan komputer. Kombinasi disiplin ilmu tersebut membuat data science powerful untuk mengolah big data. Data science dapat membantu proses pengolahan data yang meliputi pengumpulan data, manipulasi data, hingga analisis data dengan melakukan pemodelan pada kumpulan data untuk menghasilkan informasi berupa insight yang berguna dan bisa dijadikan pedoman dalam pengambilan keputusan di masa depan. Data science mengolah big data dimana berisi data terstruktur maupun tidak terstruktur. Jadi tidak hanya data numerik saja, tetapi juga data berupa suara, gambar, teks, dan sebagainya.

Data Science itu sendiri juga tidak lepas dengan istilah Big Data, lalu apa itu Big Data? Big Data adalah sebuah cara untuk mengambil, menyimpan, dan menganalisis data yang sekiranya sebelumnya tidak dapat dilakukan proses simpan hingga yang bersifat analytical. Jika diteruskan maka data tersebut bisa menjadi data yang tidak berguna atau rusak karena diperlukan cara – cara khusus untuk mengendalikannya.

Data Science menjadi sebuah istilah yang sangat penting dalam fenomena Big Data, bagaimana cara mengolah data yang begitu besar yang dihasilkan dalam internet dapat dikelola dan dianalisa dengan baik sehingga dapat menjadi sebuah data prediksi yang lebih akurat dengan data sebenarnya. Data Science merupakan cabang ilmu yang menggabungkan ilmu inferensi data, penggabungan algoritma, dan penggunaan teknologi untuk memecahkan masalah analitik yang kompleks. Dalam Data Science tidak hanya ilmu sains saja yang harus dipahami, namun berbagai ilmu yang mendukung antara lain seperti:

- Machine Learning
- Bahasa Pemrograman Python, R, atau Scala
- Ilmu analisis yang kuat

B. Hasil Kerja

Pada tugas final project kali ini, saya diberikan data 'Data_Negara_Help' untuk mengkategorikan negara mana yang membutuhkan bantuan perkembangan terhadap faktor sosial ekonomi dan kesehatan. 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.

Langkah pertama yang harus dilakukan yaitu mendeklarasikan library yang perlu dipakai, lalu membaca data yang telah diberikan.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import RobustScaler
from sklearn.cluster import KMeans

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 x 10 columns

Dapat dilihat dari gambar diatas terdapat kolom Negara, Kematian_anak, Ekspor, Kesehatan, Impor, Pendapatan, Inflasi, Harapan_hidup, Jumlah_fertiliti, GDPperkapita. Total baris dan kolomnya yaitu 167 x 10.



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
```

Lalu saya mencari tahu info data tersebut menggunakan .info() agar saya tahu apakah data tersebut terdapat nilai null atau tidak, dan ternyata berdasarkan data diatas tidak ada data yang null. Data tersebut juga memiliki data kolom yang bertipe float64 sebanyak 7, int64 sebanyak 2, dan object sebanyak 1.



data.isnull().sum()

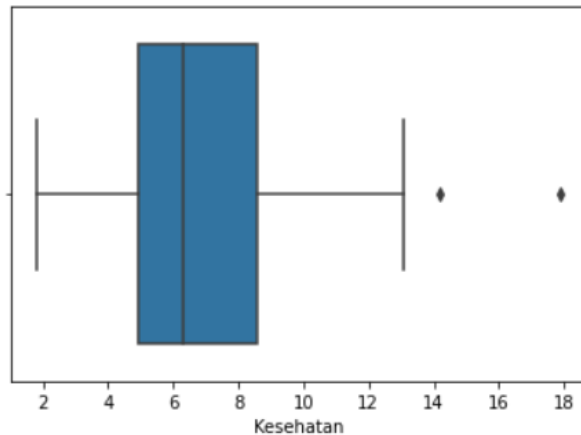


```
Negara                0
Kematian_anak         0
Ekspor                0
Kesehatan             0
Impor                 0
Pendapatan            0
Inflasi               0
Harapan_hidup         0
Jumlah_fertiliti     0
GDPperkapita          0
dtype: int64
```

Setelah itu, untuk lebih jelas apakah data tersebut terdapat nilai yang missing atau missing value (nilai null) maka saya menghitung dengan menggunakan .isnull().sum(). Dapat dilihat bahwa hasil dari masing – masing kolom yaitu 0.

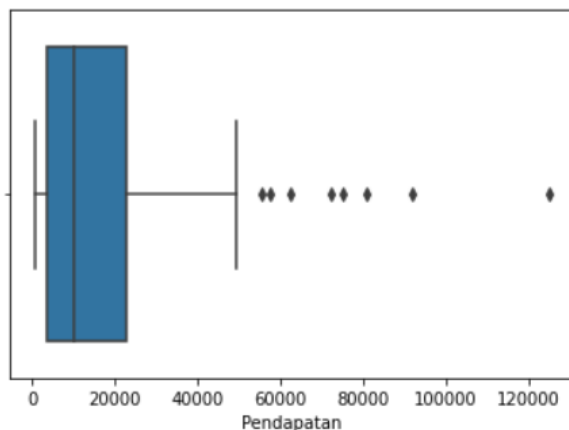
```
sns.boxplot(data['Kesehatan'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas  
FutureWarning  
<matplotlib.axes._subplots.AxesSubplot at 0x7f2a70404190>
```



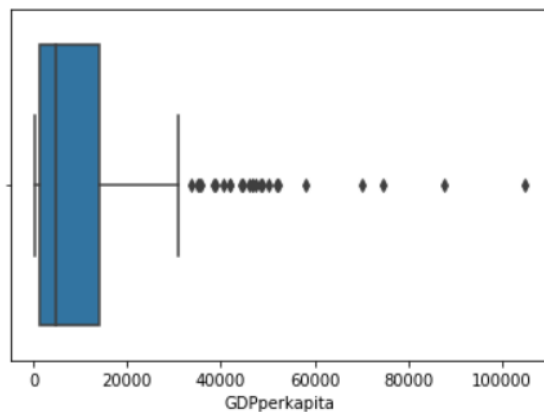
```
[ ] sns.boxplot(data['Pendapatan'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas  
FutureWarning  
<matplotlib.axes._subplots.AxesSubplot at 0x7fc68860a590>
```



```
[ ] sns.boxplot(data['GDPperkapita'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass th  
FutureWarning  
<matplotlib.axes._subplots.AxesSubplot at 0x7fc691cfd0>
```



Dapat dilihat dari 3 boxplot diatas bahwa terdapat pencilan(outliers) pada data, namun saya sengaja tidak menghilangkan outlier tersebut karena pada data terdapat negara maju dan negara yang terpuruk. Ketika divisualisasikan dengan boxplot maka negara – negara ini jadi outlier, lalu jika misalnya negara – negara ini dihapus, seharusnya negara yang terpuruk ini tidak jadi cluster – cluster yang seharusnya dibantu, padahal negara yang terpuruk seharusnya dapat bantuan. Maka dari itu saya membiarkan outlier nya.

```
#drop negara karena negara adalah data dependent
data_baru = data.drop(['Negara'], axis=1)

clusters = []
for i in range(1, 11):
    km = KMeans(n_clusters=i).fit(data_baru)
    clusters.append(km.inertia_)

print('cluster :')
clusters
```

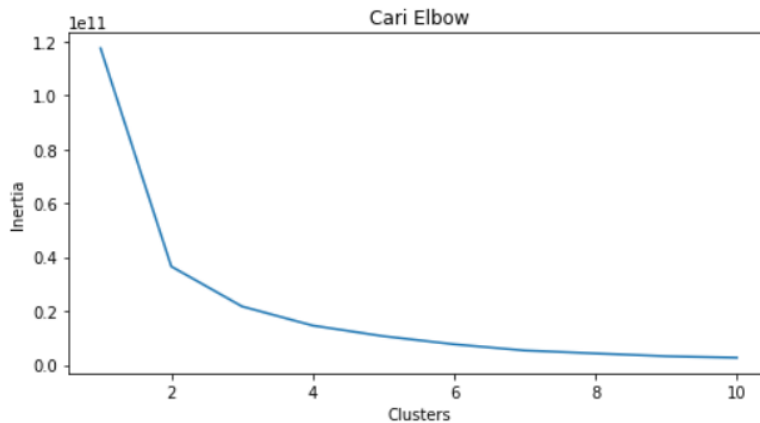
```
↳ cluster :
[117459687469.07819,
 36528387934.3221,
 21710210039.75414,
 14610713446.535133,
 10655378083.862543,
 7655725089.094505,
 5353767457.732321,
 4258897193.7515826,
 3228565265.69808,
 2662664687.013527]
```

Langkah saya mendropkan atau menghapuskan data negara karena data negara adalah data yang dependent, lalu saya membuat list kosong untuk menghitung cluster pada data baru.

```
# membuat plot inertia
# mencari elbow untuk menentukan n_cluster
fig, ax = plt.subplots(figsize=(8, 4))
sns.lineplot(x=list(range(1, 11)), y=clusters, ax=ax)
ax.set_title('Cari Elbow')
ax.set_xlabel('clusters')
ax.set_ylabel('Inertia')

#ambil cluster = 3
```

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



Setelah membuat cluster, saya mencari elbow untuk menentukan `n_cluster` yang akan saya jadikan dengan membuat plot inertia. Dari grafik diatas, yang menunjukkan datanya stabil yaitu mulai dari cluster 3. Maka dari itu yang akan menjadi `n_cluster` nya adalah 3.

```
#scaling data
sc = RobustScaler()
data_sc = sc.fit_transform(data_baru)

#clustering with kmeans
km1 = KMeans(n_clusters=3).fit(data_sc)
data_sc = pd.DataFrame(sc.inverse_transform(data_sc), columns=data_baru.columns)
data_sc['Labels'] = km1.labels_
```

```
[ ] data_sc
```

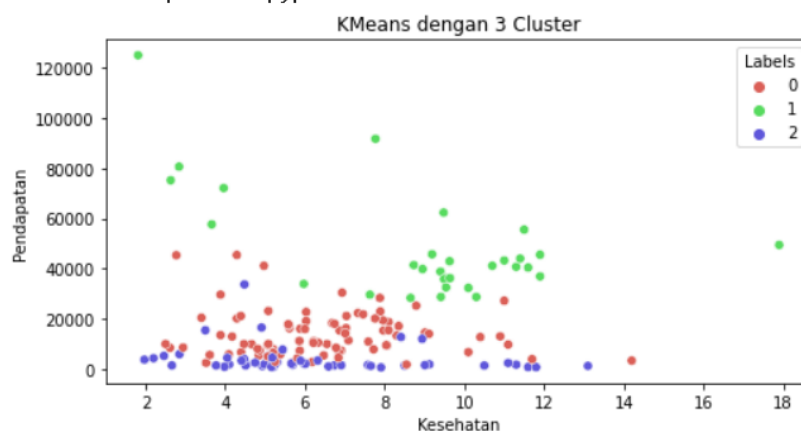
	Kematian_anak	Ekspor	Kesehatan	Inpor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita	Labels
0	90.2	10.0	7.58	44.9	1610.0	9.44	56.2	5.82	553.0	0
1	16.6	28.0	6.55	48.6	9930.0	4.49	76.3	1.65	4090.0	1
2	27.3	38.4	4.17	31.4	12900.0	16.10	76.5	2.89	4460.0	1
3	119.0	62.3	2.85	42.9	5900.0	22.40	60.1	6.16	3530.0	0
4	10.3	45.5	6.03	58.9	19100.0	1.44	76.8	2.13	12200.0	1
...
162	29.2	46.6	5.25	52.7	2950.0	2.62	63.0	3.50	2970.0	1
163	17.1	28.5	4.91	17.6	16500.0	45.90	75.4	2.47	13500.0	0
164	23.3	72.0	6.84	80.2	4490.0	12.10	73.1	1.95	1310.0	1
165	56.3	30.0	5.18	34.4	4480.0	23.60	67.5	4.67	1310.0	0
166	83.1	37.0	5.89	30.9	3280.0	14.00	52.0	5.40	1460.0	0

167 rows × 10 columns

Selanjutnya yaitu tahap menscaling data. Kali ini saya menggunakan robust scaler untuk scaling data. Dapat dilihat dari kodingan diatas tahap – tahapan scaling data. Variabel sc yaitu sebagai fungsi RobustScaler. Lalu saya membuat variabel baru yang diberi nama data_sc yang akan dipelajari oleh computer untuk dinormalisasikan data. Setelah itu saya melakukan tahapan clustering dengan menggunakan k-means. Sebelumnya data_sc merupakan sebuah array, agar menjadi dataframe saya menggunakan data_sc = pd.DataFrame untuk mengubah data tersebut menjadi dataframe, lalu agar mengembalikan nilai data, gunakan sc.inverse_transform(data_sc), karena jika tidak dikembalikan nilai awal maka yang akan muncul yaitu nilai – nilai yang telah dinormalisasikan oleh computer. Agar dapat dilihat oleh user data tersebut kembali menjadi normal yaitu dengan cara inverse transform.

```
plt.figure(figsize=(8, 4))
sns.scatterplot(
    data_sc['Kesehatan'],
    data_sc['Pendapatan'],
    hue=data_sc['Labels'],
    palette=sns.color_palette('hls', 3)
)
plt.title('KMeans dengan 3 Cluster')
plt.show
```

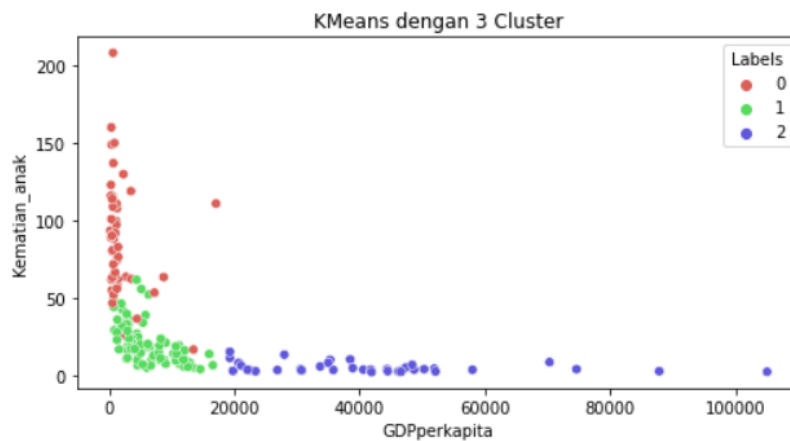
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the FutureWarning
<function matplotlib.pyplot.show>



Dapat dilihat dari grafik diatas, pada labels biru menunjukkan pendapatan yang rendah namun kesehatan nya ada yang rendah dan ada yang tidak terlalu tinggi namun pendapatannya rendah. Pada labels merah menunjukkan terdapat pendapatan yang rendah dan menengah dan kesehatan yang rendah dan ada juga yang tidak rendah. Pada labels hijau pendapatannya tinggi namun kesehatannya juga ada yang rendah dan juga yang kesehatannya tinggi. Dapat disimpulkan bahwa jika ingin mendapatkan kesehatan yang tinggi maka harusnya pendapatannya tidak rendah.


```
plt.figure(figsize=(8, 4))
sns.scatterplot(
    data_sc['GDPperkapita'],
    data_sc['Kematian_anak'],
    hue=data_sc['Labels'],
    palette=sns.color_palette('hls', 3)
)
plt.title('KMeans dengan 3 Cluster')
plt.show
```

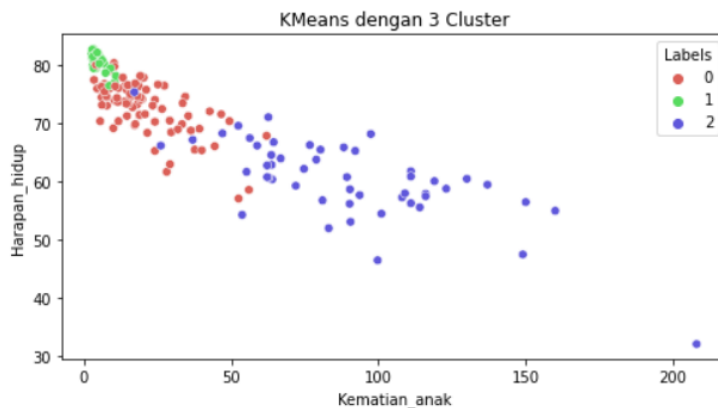
→ /usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword arguments: {'x': 'GDPperkapita', 'y': 'Kematian_anak'}. This warning will be removed in a future version of Seaborn.



Lalu pada grafik yang ini, terdapat labels hijau yang mana GDPperkapita nya rendah kematian anaknya juga rendah. Labels merah menunjukkan GDPperkapita nya rendah dan kematian anak tinggi. Labels biru menunjukkan GDPperkapita tinggi namun kematian anak rendah. Kesimpulannya untuk mendapatkan angka kematian anak yang rendah, seharusnya negara itu harus memiliki GDPperkapita yang tinggi

```
plt.figure(figsize=(8, 4))
sns.scatterplot(
    data_sc['Kematian_anak'],
    data_sc['Harapan_hidup'],
    hue=data_sc['Labels'],
    palette=sns.color_palette('hls', 3)
)
plt.title('KMeans dengan 3 Cluster')
plt.show
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following argument(s) into the function: matplotlib.pyplot.show




Pada grafik yang ini, dapat dilihat bahwa labels hijau menunjukkan bahwa angka kematian anak rendah namun harapan hidup tinggi, sedangkan labels merah menunjukkan angka kematian anak dari rendah menuju menengah namun harapan hidup ada yang tinggi dan sedikit yang menengah. Sedangkan labels biru menunjukkan angka kematian tinggi dan harapan hidup dari menengah menuju ke rendah. Dapat disimpulkan bahwa jika ingin angka kematian anak rendah maka seharusnya harapannya itu tinggi.

```
data[data_sc['Kesehatan'] < 4].where(data_sc['Pendapatan'] < 20000).dropna()
```

	Negara	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
3	Angola	119.0	62.300	2.85	42.9000	5900.0	22.40	60.1	6.16	3530.0
12	Bangladesh	49.4	16.000	3.52	21.8000	2440.0	7.14	70.4	2.33	758.0
31	Central African Republic	149.0	11.800	3.98	26.5000	888.0	2.01	47.5	5.21	446.0
38	Congo, Rep.	63.9	85.100	2.46	54.7000	5190.0	20.70	60.4	4.95	2740.0
50	Eritrea	55.2	4.790	2.66	23.3000	1420.0	11.60	61.7	4.61	482.0
55	Gabon	63.7	57.700	3.50	18.9000	15400.0	16.60	62.9	4.08	8750.0
70	Indonesia	33.3	24.300	2.61	22.4000	8430.0	15.30	69.9	2.48	3110.0
93	Madagascar	62.2	25.000	3.77	43.0000	1390.0	8.79	60.8	4.60	413.0
107	Myanmar	64.4	0.109	1.97	0.0659	3720.0	7.04	66.8	2.41	988.0
116	Pakistan	92.1	13.500	2.20	19.4000	4280.0	10.90	65.3	3.85	1040.0
120	Philippines	31.9	34.800	3.61	36.6000	5600.0	4.22	69.0	3.16	2130.0
140	Sri Lanka	11.2	19.600	2.94	26.8000	8560.0	22.80	74.4	2.20	2810.0
148	Thailand	14.9	66.500	3.88	60.8000	13500.0	4.08	76.6	1.55	5080.0
154	Turkmenistan	62.0	76.300	2.50	44.5000	9940.0	2.31	67.9	2.83	4440.0


Pada data berikut menunjukkan negara yang mana memiliki kesehatan dibawah angka 4, dan pendapatan dibawah 20000. Menurut saya, dari angka ini negara yang harus diberi bantuan oleh HELP International karena negara tersebut memiliki kesehatan yang rendah dan

pendapatan yang rendah. Oleh karena itu, untuk membantu negara – negara tersebut, saya menyarankan HELP International untuk membantu negara – negara yang tertera pada data diatas.

 `data[data_sc['GDPperkapita']<20000].where(data_sc['Kematian_anak']>100).dropna()`

	Negara	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
3	Angola	119.0	62.3	2.85	42.9	5900.0	22.400	60.1	6.16	3530.0
17	Benin	111.0	23.8	4.10	37.2	1820.0	0.885	61.8	5.36	758.0
25	Burkina Faso	116.0	19.2	6.74	29.6	1430.0	6.810	57.9	5.87	575.0
28	Cameroon	108.0	22.2	5.13	27.0	2660.0	1.910	57.3	5.11	1310.0
31	Central African Republic	149.0	11.8	3.98	26.5	888.0	2.010	47.5	5.21	446.0
32	Chad	150.0	36.8	4.53	43.5	1930.0	6.390	56.5	6.59	897.0
37	Congo, Dem. Rep.	116.0	41.1	7.91	49.6	609.0	20.800	57.5	6.54	334.0
40	Cote d'Ivoire	111.0	50.6	5.30	43.3	2690.0	5.390	56.3	5.27	1220.0
49	Equatorial Guinea	111.0	85.8	4.48	58.9	33700.0	24.900	60.9	5.21	17100.0
63	Guinea	109.0	30.3	4.93	43.2	1190.0	16.100	58.0	5.34	648.0
64	Guinea-Bissau	114.0	14.9	8.50	35.2	1390.0	2.970	55.6	5.05	547.0
66	Haiti	208.0	15.3	6.91	64.7	1500.0	5.450	32.1	3.33	662.0
97	Mali	137.0	22.8	4.98	35.1	1870.0	4.370	59.5	6.55	708.0
106	Mozambique	101.0	31.5	5.21	46.2	918.0	7.640	54.5	5.56	419.0
112	Niger	123.0	22.2	5.16	49.1	814.0	2.550	58.8	7.49	348.0
113	Nigeria	130.0	25.3	5.07	17.4	5150.0	104.000	60.5	5.84	2330.0
132	Sierra Leone	160.0	16.8	13.10	34.5	1220.0	17.200	55.0	5.20	399.0

Data diatas menunjukkan negara-negara yang memiliki angka GDPperkapita kurang dari 20000 dan angka kematian anak lebih dari 100. Hal ini menunjukkan bahwa negara tersebut memiliki angka GDPperkapita yang rendah dan angka kematian anak yang tinggi. Oleh karena itu, untuk membantu negara – negara tersebut, saya menyarankan HELP International untuk membantu negara – negara yang tertera pada data diatas.

 `data[data_sc['Kematian_anak']>100].where(data_sc['Harapan_hidup']<50).dropna()`

	Negara	Kematian_anak	Ekspor	Kesehatan	Impor	Pendapatan	Inflasi	Harapan_hidup	Jumlah_fertiliti	GDPperkapita
31	Central African Republic	149.0	11.8	3.98	26.5	888.0	2.01	47.5	5.21	446.0
66	Haiti	208.0	15.3	6.91	64.7	1500.0	5.45	32.1	3.33	662.0

Data diatas menunjukkan negara-negara yang memiliki angka kematian diatas 100 dan harapan hidup dibawah 50. Hal ini menunjukkan negara tersebut memiliki angka kematian anak yang tinggi dan harapan hidup yang rendah. Oleh karena itu, untuk membantu negara – negara tersebut, saya menyarankan HELP International untuk membantu negara – negara yang tertera pada data diatas.

Kesimpulan:

Berdasarkan dari 3 data diatas, saya menyarankan HELP International untuk membantu negara berikut:

- Angola
- Bangladesh
- Central African Republic
- Congo, Rep.
- Congo, Dem. Rep.
- Eritrea
- Gabon
- Indonesia
- Madagascar
- Myanmar
- Pakistan
- Phillipines
- Sri Lanka
- Thailand
- Turkmenistan
- Benin
- Burkina Faso
- Cameroon
- Chad
- Cote d'Ivoire
- Equatorial Guinea
- Guinea
- Guine-Bissau
- Haiti
- Mali
- Mozambique
- Niger
- Nigeria
- Sierra Leone