

Nama : Alif Adhitya
NIM : 202131126
Mata Kuliah : Pengantar Big Data (A)

BAB 1 – Pengumpulan Data

```
In [1]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score

In [2]: dataset = pd.read_csv('2015.csv')
```

Langkah pertama yang dilakukan adalah mengimpor library dan fungsi yang dibutuhkan. Disini saya mengimpor pandas, seaborn, matplotlib, numpy, scikit-learn, dengan 3 fungsi yaitu `train_test_split`, `accuracy_score`, dan `precision_score`.

Berikut masing – masing fungsi :

1. **Pandas (import pandas as pd):**
 - o Berfungsi menyediakan struktur data dan alat analisis data yang efisien. DataFrames.
2. **Seaborn (import seaborn as sns):**
 - o Berfungsi untuk visualisasi data berbasis Matplotlib. Ini menyediakan antarmuka tingkat tinggi untuk membuat grafik statistik informatif dan menarik.
3. **Matplotlib (import matplotlib.pyplot as plt):**
 - o Bergungsi untuk membuat visualisasi statis, animasi, dan interaktif dalam Python. pyplot, modul yang diimpor di sini, menyediakan antarmuka untuk membuat berbagai jenis grafik dan plot.
4. **NumPy (import numpy as np):**
 - o **Library ini** menyediakan objek array multidimensi, fungsi matematika tingkat tinggi, dan alat untuk bekerja dengan data array.
5. **Scikit-Learn (from sklearn.model_selection import train_test_split):**
 - o `train_test_split` adalah fungsi yang digunakan untuk membagi dataset menjadi subset pelatihan dan pengujian, yang diperlukan dalam proses evaluasi model.
6. **Scikit-Learn (from sklearn.metrics import accuracy_score):**
 - o `accuracy_score` mengukur sejauh mana prediksi model sesuai dengan nilai sebenarnya dan memberikan akurasi sebagai persentase prediksi yang benar.
7. **Scikit-Learn (from sklearn.metrics import precision_score):**
 - o `precision_score` mengukur sejauh mana model memberikan prediksi positif yang benar di antara semua instance yang diprediksi positif. Ini berguna dalam kasus di mana penting untuk mengurangi jumlah false positive.

Lalu membuat variabel untuk menampung dataset. Dimana dataset didapat dari kaggle. Disini untuk membaca data nya menggunakan fungsi pandas.

```
In [3]: dataset
```

```
Out[3]:
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176
...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.67042
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260	1.63328
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.32858
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.83302
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681	1.56726

158 rows × 12 columns

Lalu berikutnya menampilkan dataset yang ada.

BAB 2 – Menelaah Data

Pada BAB 2, dilakukan penelaahan data untuk mencari informasi tentang karakteristik data yang dipunya.

```
In [4]: dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Country                               158 non-null    object
1   Region                               158 non-null    object
2   Happiness Rank                        158 non-null    int64
3   Happiness Score                       158 non-null    float64
4   Standard Error                       158 non-null    float64
5   Economy (GDP per Capita)              158 non-null    float64
6   Family                               158 non-null    float64
7   Health (Life Expectancy)              158 non-null    float64
8   Freedom                              158 non-null    float64
9   Trust (Government Corruption)         158 non-null    float64
10  Generosity                            158 non-null    float64
11  Dystopia Residual                     158 non-null    float64
dtypes: float64(9), int64(1), object(2)
memory usage: 14.9+ KB
```

Disini saya menampilkan informasi mengenai dataset yang ada.

```
In [5]: categorical_col = []
for column in dataset.columns:
    if dataset[column].dtype == object and len(dataset[column].unique()) <= 50:
        categorical_col.append(column)
        print(f"{column} : {dataset[column].unique()}")
        print("=====")
```

```
Region : ['Western Europe' 'North America' 'Australia and New Zealand'
'Middle East and Northern Africa' 'Latin America and Caribbean'
'Southeastern Asia' 'Central and Eastern Europe' 'Eastern Asia'
'Sub-Saharan Africa' 'Southern Asia']
=====
```

Disini saya ingin melihat nama kolom dan nilai – nilai uniknya untuk setiap kolom yang dianggap sebagai kolom kategorikal dalam dataset. Hasilnya adalah informasi mengenai karakter dan variasi dalam kolom kolom tersebut.

```
In [6]: print("\nJumlah Missing Values per Kolom:")
print(dataset.isnull().sum())
```

```
Jumlah Missing Values per Kolom:
Country                0
Region                 0
Happiness Rank         0
Happiness Score        0
Standard Error         0
Economy (GDP per Capita) 0
Family                0
Health (Life Expectancy) 0
Freedom                0
Trust (Government Corruption) 0
Generosity             0
Dystopia Residual       0
dtype: int64
```

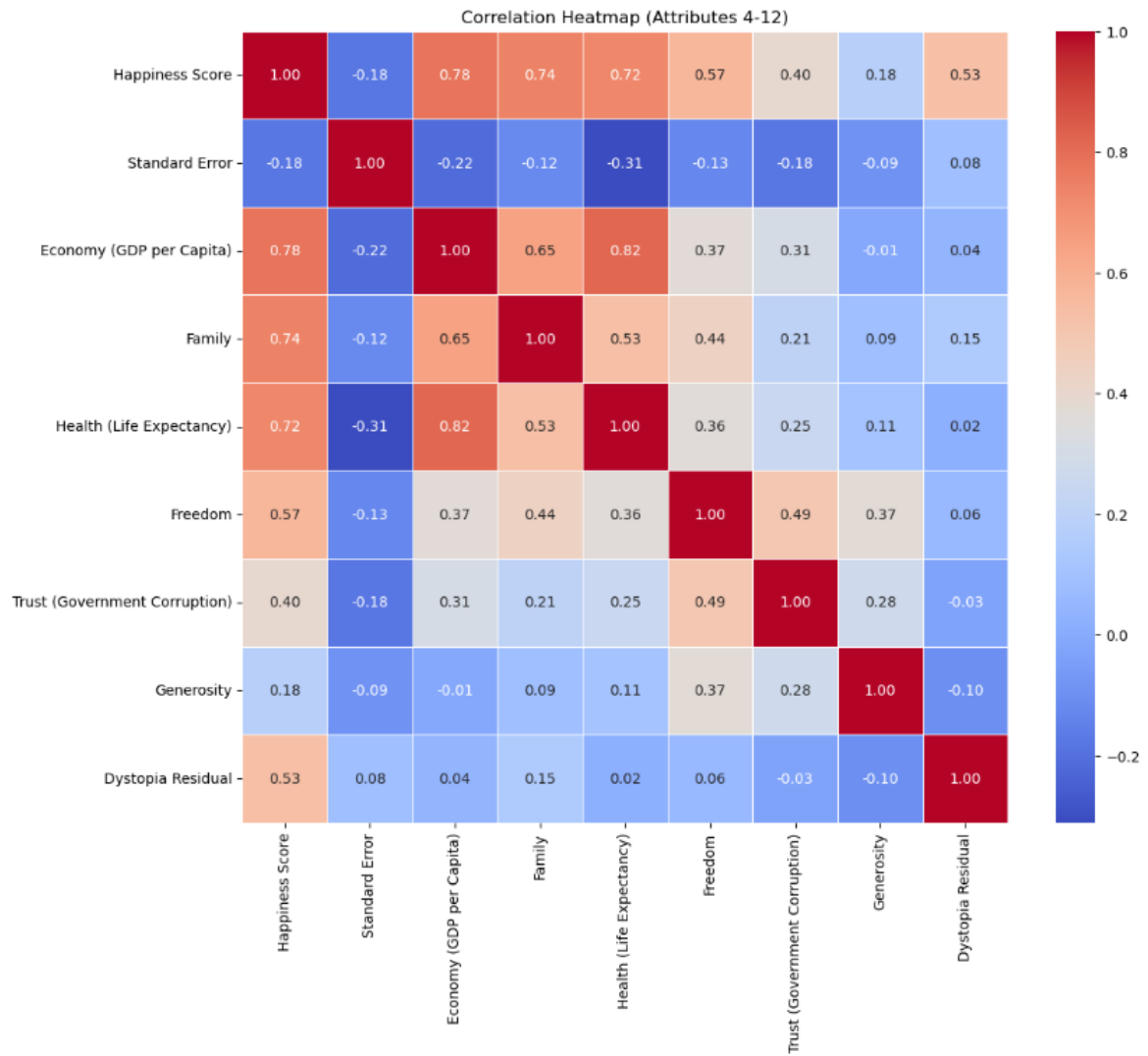
Kode diatas berfungsi untuk menampilkan jumlah jumlah missing value di tiap tiap kolom.

```
In [7]: # Select columns 4 to 12
selected_columns = dataset.iloc[:, 3:12]

# Generate a correlation matrix
correlation_matrix = selected_columns.corr()

# Create a heatmap using seaborn
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Heatmap (Attributes 4-12)')
plt.show()
```

Lalu berikutnya membuat heatmap. Heatmap sendiri adalah bentuk visualisasi data yang menggunakan warna untuk menggambarkan intensitas nilai dalam suatu matriks. Fungsi utama dari heatmap adalah memberikan gambaran visual tentang distribusi dan pola hubungan antara dua variabel atau lebih dalam bentuk matriks.



```
In [8]: descriptive_stats = dataset.describe()
print(descriptive_stats)
```

	Happiness Rank	Happiness Score	Standard Error \
count	158.000000	158.000000	158.000000
mean	79.493671	5.375734	0.047885
std	45.754363	1.145010	0.017146
min	1.000000	2.839000	0.018480
25%	40.250000	4.526000	0.037268
50%	79.500000	5.232500	0.043940
75%	118.750000	6.243750	0.052300
max	158.000000	7.587000	0.136930

	Economy (GDP per Capita)	Family Health (Life Expectancy) \
count	158.000000	158.000000
mean	0.846137	0.991046
std	0.403121	0.272369
min	0.000000	0.000000
25%	0.545808	0.856823
50%	0.910245	1.029510
75%	1.158448	1.214405
max	1.690420	1.402230

	Freedom Trust (Government Corruption)	Generosity \
count	158.000000	158.000000
mean	0.428615	0.143422
std	0.150693	0.120034
min	0.000000	0.000000
25%	0.328330	0.061675
50%	0.435515	0.107220
75%	0.549092	0.180255
max	0.669730	0.551910

	Dystopia Residual
count	158.000000
mean	2.098977
std	0.553550
min	0.328580
25%	1.759410
50%	2.095415
75%	2.462415
max	3.602140

Dan yang terakhir saya menampilkan rata – rata, standar deviasi, nilai minimum, dan nilai maksimum. Kode ini menghitung masing – masing nilai dari attribut.

BAB 3 – Memvalidasi Data

Temuan :

1. Dalam dataset terdapat 3 tipe data, yaitu integer, float, dan object
2. Jumlah tiap - tiap missing value pada kolom adalah 0 atau tidak ada

Kedepannya harus menentukan objek data dan menentukan tipe data seperti apa yang dibutuhkan untuk membangun model kedepannya

BAB 4 – Menentukan Objek Data

Untuk menentukan objek data, saya menggunakan generate table yang telah diberikan di contoh modul yaitu https://www.tablesgenerator.com/markdown_tables Berikut hasil tabel yang saya buat berdasarkan tujuan dari pengembangan dataset ini :

	A	B
1	Country	
2	Region	
3	Happiness Rank	
4	Happiness Score	Class
5	Standard Error	
6	Economy (GDP per Capita)	Attribute
7	Health (life Expectancy)	Attribute
8	Freedom	Attribute
9	Trust (Government Corruption)	Attribute
10	Generosity	Attribute
11	Dystopia Residual	Attribute

1	Country		
2	-----		
3	Region		
4	Happiness Rank		
5	Happiness Score	Class	
6	Standard Error		
7	Economy (GDP per Capita)	Attribute	
8	Health (life Expectancy)	Attribute	
9	Freedom	Attribute	
10	Trust (Government Corruption)	Attribute	
11	Generosity	Attribute	
12	Dystopia Residual	Attribute	

Total terdapat 6 atribut yaitu Economy (GDP per Capita), Health (life Expectancy), Freedom, Trust (Government Corruption), Generosity, Dytopia Residual dan 1 class yaitu Happiness score. Keenam atribut tersebut akan dijadikan sebagai independent variabel dan class happiness score akan menjadi dependent variabel.

Terlihat bahwa terdapat 3 field yang tidak diisi keterangannya yaitu Region, Happiness Rank, dan Standard Error. Ketiga field ini tidak diisi karena nanti nya field ini akan di hapus karena tidak sesuai dengan tujuan regresi linear.

Bab 5 – Membersihkan Data

Untuk membersihkan data, langkah pertama yang saya lakukan adalah menghilangkan data yang masih ada tanda tanya. Untuk melakukannya saya menggunakan fungsi 'replace'. Lalu disini saya meng import satu library yaitu 'missingno' dan diinisialisasi ms yang berfungsi mengecek distribusi data yang hilang nanti.

```
In [25]: import missingno as ms
```

```
In [26]: dataset.replace("?", np.nan, inplace= True)
```

Lalu yang berikutnya saya memanggil 5 data teratas dan 5 data terbawah menggunakan fungsi head dan tail.

```
In [27]: dataset.head()
```

```
Out[27]:
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176

```
In [28]: dataset.tail()
```

```
Out[28]:
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.67042
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28865	0.35386	0.31910	0.48450	0.08010	0.18260	1.63328
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.32858
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.83302

Berikutnya memeriksa apakah ada data yang 'null' menggunakan fungsi isnull dan disini jika hasilnya True, maka terdapat data yang 'null'. Tetapi apabila terdapat hasilnya adalah False, maka dalam dataset tidak ditemukan data yang 'null'.

Melihat apakah ada data yang null

```
In [14]: dataset.isnull().values.any()
```

```
Out[14]: False
```

Berikutnya memeriksa attribut yang terdiri dari data hilang atau kosong. Hasilnya akan menunjukkan index beberapa attribut yang terdapat data yang hilang. Dan disini hasilnya adalah kosong atau tidak terdapat attribut yang memiliki data yang hilang.

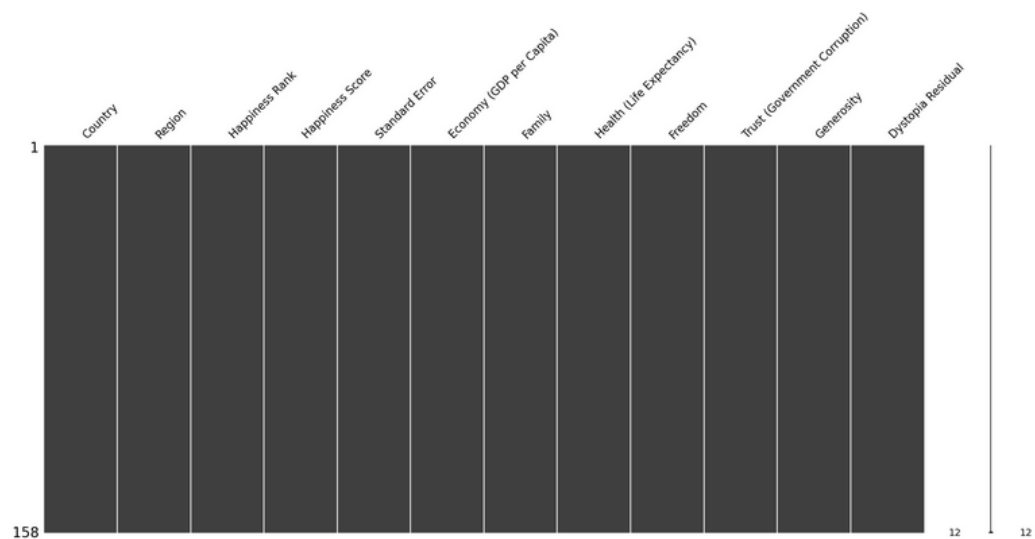
```
In [23]: dataset.loc[:, dataset.isnull().any()].columns
```

```
Out[23]: Index([], dtype='object')
```

Lalu berikutnya mengecek distribusi data yang hilang menggunakan library 'missingno'. Apabila terdapat data yang hilang dalam satu field, maka akan terdapat garis putih di kolom tersebut. Pada dataset saya, tidak ditemukan data yang hilang.

```
In [24]: ms.matrix(dataset)

plt.show()
```



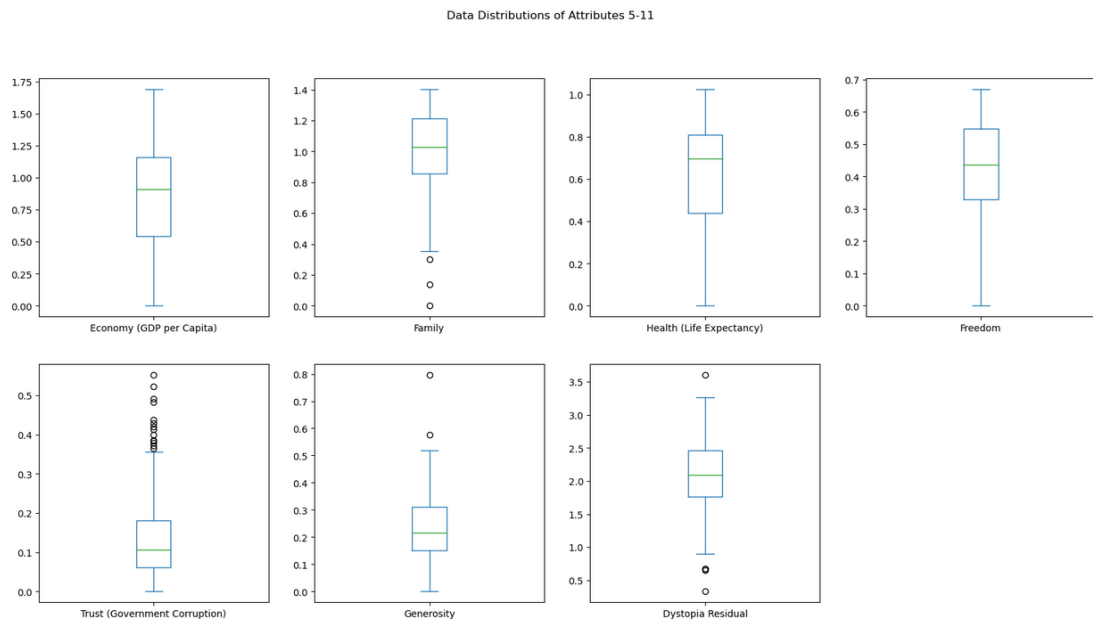
Bab 6 – Mengkonstruksi Data

Dari info data yang telah ada, ditemukan bahwa dari 12 kolom, terdapat 158 non-null records dan juga 9 float tipe data, 1 integer, dan 2 object.

Pada tujuan yang ingin dicapai, saya membutuhkan kolom 4 - 11 sebagai independent variable dan kolom ke 3 sebagai dependent variable. Tipe data yang dibutuhkan sudah sesuai yaitu float.

Disini saya membuat box plots tetapi hanya kolom yang menjadi attribut atau feature (variabel independent) karena diperlukan untuk melihat apakah ada outliers dalam penyebaran data dari masing – masing kolom.


```
In [26]: attributes_to_plot = dataset.columns[5:12]
dataset[attributes_to_plot].plot(kind='box', subplots=True, layout=(2, 4), sharex=False, figsize=(20, 10),
                                title='Data Distributions of Attributes 5-11')
plt.show()
```



Berdasarkan hasil box plots, tidak ditemukakn outliers.

Bab 7 – Menentukan Label Data

Dikarenakan tujuannya adalah regresi linear, attribut harus dibagi menjadi independent variabel dan dependent variabel. Disini variabel dependent nya (y) adalah Score Happiness.

Disini saya membuat variabel y yang akan menampung kolom dependent variable yaitu Happiness Score.

```
In [27]: y = dataset['Happiness Score']
```

```
In [28]: y
```

```
Out[28]: 0      7.587
         1      7.561
         2      7.527
         3      7.522
         4      7.427
         ...
        153     3.465
        154     3.340
        155     3.006
        156     2.905
        157     2.839
         Name: Happiness Score, Length: 158, dtype: float64
```

Lalu berikutnya saya membuat label atau kolom baru yang berfungsi untuk mengelompokkan tingkat kebahagiaan seseorang. Disini saya membaginya secara mandiri yaitu Low, Medium, dan High dengan keterangan sebagai berikut :

Low	< 3
Medium	$3 \leq Y < 6,5$
High	$Y \geq 6,5$

Berikut source code nya :

```
In [29]: bins = [0, 3, 6.5, float('inf')]
labels = ['Low', 'Medium', 'High']

dataset['Happiness Level'] = pd.cut(dataset['Happiness Score'], bins=bins, labels=labels)
```

Bins dan labels digunakan untuk menentukan level kebahagiaan dalam bentuk float. Lalu labels berisi label atau class seperti apa yang dijadikan level. Lalu disini saya menggunakan fungsi 'cut' untuk menambahkan kolom baru tersebut dengan nama 'Happiness Level' ke dataset dan masing masing record disesuaikan dengan nilai pada 'Happiness Score'.

Berikut merupakan hasil dari akhir dataset setelah ditambah kolom baru :

Out[30]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual	Happiness Level
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738	High
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201	High
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204	High
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531	High
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176	High
...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.67042	Medium
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260	1.63328	Medium
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.32858	Medium
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.83302	Low
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681	1.56726	Low

158 rows x 13 columns

Bab 8 – Membuat Model

Model yang akan dibangun adalah regresi linear. Regresi linear adalah algoritma dalam machine learning yang termasuk merupakan klasifikasi, dimana disini program akan mencari hubungan antara variabel independen dan variabel dependen.

Bab 8 - Membangun Model

Algoritma yang akan digunakan adalah Regresi Linear. Fungsi regresi linear disini adalah untuk memprediksi score kebahagiaan (Happiness Score)

```
In [23]: from sklearn.linear_model import LinearRegression
        from sklearn.metrics import mean_squared_error, r2_score
```

```
In [25]: dataset = dataset.drop(columns=['Country', 'Region', 'Happiness Rank', 'Standard Error', 'Happiness Level'])
```

```
In [26]: dataset
```

```
Out[26]:
```

	Happiness Score	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	7.587	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
1	7.561	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
2	7.527	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
3	7.522	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
4	7.427	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176
...
153	3.465	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.67042
154	3.340	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260	1.63328
155	3.006	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.32858
156	2.905	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.83302
157	2.839	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681	1.56726

158 rows x 8 columns

Yang pertama dilakukan adalah mengimport lagi beberapa library yang dibutuhkan diantara adalah sklearn LinearRegression yang akan digunakan untuk membangun model regresi linear. Lalu yang kedua adalah meng import dua metriks yaitu mean_squared_error dan r2_score yang digunakan untuk mengukur hubungan antara variabel dependen dan variabel independen.

Lalu berikutnya menghilangkan kolom kolom yang tidak digunakan dalam proses regresi lienar menggunakan fungsi drop.

```
In [29]: x = dataset[['Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)',
                    'Generosity', 'Dystopia Residual']]
        y = dataset['Happiness Score']
```

Lalu berikutnya adalah menentukan kolom mana saja yang dijadikan variabel independen dan variabel dependen dan menyimpannya dalam variabel x untuk variabel independen dan variabel y untuk menyimpan variabel dependen.

```
In [30]: X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

Lalu membagi data menjadi data menjadi data latih dan data tes menggunakan fungsi dari library train_test_split. Dan juga mennetukan berapakah persentase yang akan dijadikan data tes. Disini yang akan dijadikan data tes sebesar 20% dari total data.

```
In [31]: model = LinearRegression()
```

```
In [32]: model.fit(X_train, y_train)
```

```
Out[32]:  
▼ LinearRegression  
LinearRegression()
```

```
In [33]: y_pred = model.predict(X_test)
```

Berikutnya adalah membangun model regresi untuk pemrosesan algoritmanya. Lalu setelahnya melatih data dan juga membuat variabel untuk menampung nilai prediksi menggunakan fungsi predict yang menerima parameter X_test (data tes).

```
In [34]: mse = mean_squared_error(y_test, y_pred)  
r2 = r2_score(y_test, y_pred)
```

```
In [35]: print(f'Mean Squared Error: {mse}')  
print(f'R-squared (R2): {r2}')
```

```
Mean Squared Error: 7.313926609479024e-08  
R-squared (R2): 0.9999999484481188
```

```
In [36]: print("Koefisien model:")  
for feature, coef in zip(X.columns, model.coef_):  
    print(f'{feature}: {coef}')
```

```
print(f'Intercept: {model.intercept_}')
```

```
Koefisien model:  
Economy (GDP per Capita): 1.0001182677311922  
Family: 0.9999683155738956  
Health (Life Expectancy): 0.9999108990634991  
Freedom: 0.9998357429785737  
Trust (Government Corruption): 0.9999902711128004  
Generosity: 0.9999088001614574  
Dystopia Residual: 1.000027789502709  
Intercept: 9.670242957504627e-06
```

Lalu yang terakhir melihat hasil dari regresi linear. Disini saya mencari nilai MSE dan R² nya. MSE mengukur rata-rata dari kuadrat selisih antara nilai prediksi dan nilai sebenarnya. Semakin kecil MSE, semakin baik modelnya dalam menyesuaikan data. Nilai 7,31392....menunjukkan bahwa model regresi linear secara rata-rata memiliki kesalahan prediksi yang cukup tinggi.

Lalu R² sendiri mengukur sejauh mana variabilitas dalam variabel dependen dapat dijelaskan oleh model. Nilai R-squared berkisar antara 0 dan 1. Nilai mendekati 1 menunjukkan bahwa model mampu menjelaskan sebagian besar variasi dalam data. Dalam konteks ini, nilai yang sangat mendekati 1 (0.9999999484481188) menunjukkan bahwa model regresi linear secara sangat baik menjelaskan variasi dalam data.

Lalu berikutnya adalah saya menampilkan koefisien model yang menggambarkan seberapa besar variabel independen mempengaruhi variabel dependen dalam model regresi linear. Berikut kesimpulannya :

1. **Economy (GDP per Capita):** Sebuah peningkatan sebesar 1 unit dalam variabel ini akan menyebabkan peningkatan sekitar 1.0001182677311922 unit dalam variabel dependen.
2. **Family:** Sebuah peningkatan sebesar 1 unit dalam variabel ini akan menyebabkan peningkatan sekitar 0.9999683155738956 unit dalam variabel dependen.

3. **Health (Life Expectancy):** Sebuah peningkatan sebesar 1 unit dalam variabel ini akan menyebabkan peningkatan sekitar 0.9999108990634991 unit dalam variabel dependen.
4. **Freedom:** Sebuah peningkatan sebesar 1 unit dalam variabel ini akan menyebabkan peningkatan sekitar 0.9998357429785737 unit dalam variabel dependen.
5. **Trust (Government Corruption):** Sebuah peningkatan sebesar 1 unit dalam variabel ini akan menyebabkan peningkatan sekitar 0.9999902711128004 unit dalam variabel dependen.
6. **Generosity:** Sebuah peningkatan sebesar 1 unit dalam variabel ini akan menyebabkan peningkatan sekitar 0.9999088001614574 unit dalam variabel dependen.
7. **Dystopia Residual:** Sebuah peningkatan sebesar 1 unit dalam variabel ini akan menyebabkan peningkatan sekitar 1.000027789502709 unit dalam variabel dependen.
8. **Intercept:** Ini adalah nilai yang ditambahkan ke hasil prediksi dan merupakan nilai prediksi saat semua variabel independen diatur ke nol.