ibimbing

"POKEMON" ANALISIS

Digital Skill Fair 24.0 – Data Science

Portofolio by **Khairina Altaf Salsabila**

ABOUT ME

Hai, saya **Rina.**

Saya adalah seorang fresh graduate **Sarjana Matematika** di Universitas Sebelas Maret yang sangat tertarik dalam dunia **data**. Saya mempunyai pribadi yang **berdedikasi** tinggi dan **berkomitmen** penuh di setiap aspek dalam hidup saya dan juga **konsisten** mencari solusi kreatif di setiap tantangan yang saya hadapi.

Dengan background pendidikan dan beberapa pelatihan yang saya ikuti, saya memiliki dasar dan kemampuan dalam **menganalisis** dan **memvisualisasikan** data menggunakan **Microsoft Excel, Python, SQL, dan Tableu.**

SUMMARY

Dataset yang digunakan dalam data analisis ini yaitu dataset **Pokemon**. Data ini diambil dari https://bit.ly/data-pokemon-dsf yang merupakan data study case dari salah satu assignments Dibimbing.

Dataset ini mempunyai informasi mengenai **800 Pokemon** dari total enam Generasi Pokemon, terdapat pula tipe dan beberapa statistik dari masing-masing Pokemon. Selain untuk memenuhi tugas dari Digital Skill Fair – Data Science yang diselenggarakan Dibimbing, data ini sangat menarik untuk dianalisis dan dieksplorasi lebih lanjut.

Hasil analisis berupa **visualisasi data** dan penerapan **K-Nearest Neighbors (KNN)** untuk mengklasifikasikan Legendary Pokemon menggunakan bahasa pemrograman **Python**.

TRY TO ANSWER THIS QUESTION

1. Berapa jumlah Pokemon per Generation?
2. Berapa jumlah Pokemon per Type?
3. Berapa jumlah perbandingan Pokemon yang Legendary dan tidak?
4. Type Pokemon mana yang paling kuat?
5. Bagaimana korelasi antara masing-masing atribut?
6. Mengklasifikasikan Pokemon, apakah termasuk Legendary atau tidak?

DATASET

```
#menampilkan beberapa baris pertama dan terakhir pada dataset
pokemon
```

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...

# = nomor seri pokemon
Name = nama pokemon
Type 1 = tipe utama pokemon
Type 2 = tipe sekunder pokemon
Total = total HP, Attack, Defense, speed
HP = health point
Attack = kekuatan serangan

Defense = kekuatan pertahanan
Sp. Atk = kekuatan serangan khusus
Sp. Def = kekuatan pertahanan khusus
Speed = tingkat kecepatan
Generation = generasi
Legendary = legendaris

DATASET

Dataset Pokemon mempunyai **800 baris** dan **13 kolom** dengan rincian kolomnya yaitu **satu kolom** mempunyai tipe data **boolean**, **sembilan kolom** mempunyai tipe data **integer**, dan **tiga kolom** mempunyai tipe data **string/object**. Ada 800 data yang diinput dan memory yang digunakan sebesar 75.9+ KB.

```
#menampilkan gambaran umum dataset seperti tipe data
pokemon.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   #           800 non-null    int64
 1   Name        800 non-null    object
 2   Type 1      800 non-null    object
 3   Type 2      414 non-null    object
 4   Total       800 non-null    int64
 5   HP          800 non-null    int64
 6   Attack      800 non-null    int64
 7   Defense     800 non-null    int64
 8   Sp. Atk     800 non-null    int64
 9   Sp. Def     800 non-null    int64
 10  Speed       800 non-null    int64
 11  Generation  800 non-null    int64
 12  Legendary   800 non-null    bool
dtypes: bool(1), int64(9), object(3)
memory usage: 75.9+ KB
```

Pre-Processing

DESCRIPTIVE STATISTICS

Pada Gambar di bawah dapat diketahui mengenai count (jumlah data), mean (rata-rata), std (standar deviasi atau simpangan baku), min (nilai terendah), 25% (kuartil bawah), 50% (kuartil tengah atau median), 75% (kuartil atas), dan max (nilai tertinggi) di masing-masing **kolom** dataset **yang mempunyai nilai numerik**.

```
#menampilkan statistik deskriptif dataset (hanya pada kolom yang bernilai numerik)
pokemon.describe()
```

	#	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation
count	800.000000	800.00000	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000	800.00000
mean	362.813750	435.10250	69.258750	79.001250	73.842500	72.820000	71.902500	68.277500	3.32375
std	208.343798	119.96304	25.534669	32.457366	31.183501	32.722294	27.828916	29.060474	1.66129
min	1.000000	180.00000	1.000000	5.000000	5.000000	10.000000	20.000000	5.000000	1.00000
25%	184.750000	330.00000	50.000000	55.000000	50.000000	49.750000	50.000000	45.000000	2.00000
50%	364.500000	450.00000	65.000000	75.000000	70.000000	65.000000	70.000000	65.000000	3.00000
75%	539.250000	515.00000	80.000000	100.000000	90.000000	95.000000	90.000000	90.000000	5.00000
max	721.000000	780.00000	255.000000	190.000000	230.000000	194.000000	230.000000	180.000000	6.00000

Dapat dilihat juga bahwa nilai minimum tidak ada yang bernilai 0. Hal ini berarti **tidak ada data yang bernilai 0 ataupun kosong**.

SHOWING MISSING VALUE

```
#menampilkan jumlah data kosong
print(pokemon.isna().sum())
```

```
#               0
Name            0
Type 1          0
Type 2        386
Total           0
HP              0
Attack          0
Defense         0
Sp. Atk         0
Sp. Def         0
Speed           0
Generation      0
Legendary       0
dtype: int64
```

Pada Gambar tersebut dapat dilihat bahwa kolom **Type 2** mempunyai **missing value** sebanyak **386** data dan kolom lainnya tidak ada missing value.

HANDLING MISSING VALUE

```
pokemon.columns=pokemon.columns.str.replace(' ','') #menghilangkan spasi yang terdapat pada judul kolom
pokemon['Type2'].fillna('None',inplace=True) #mengganti missing value di kolom Type 2 dengan nilai None
pokemon.head()
```

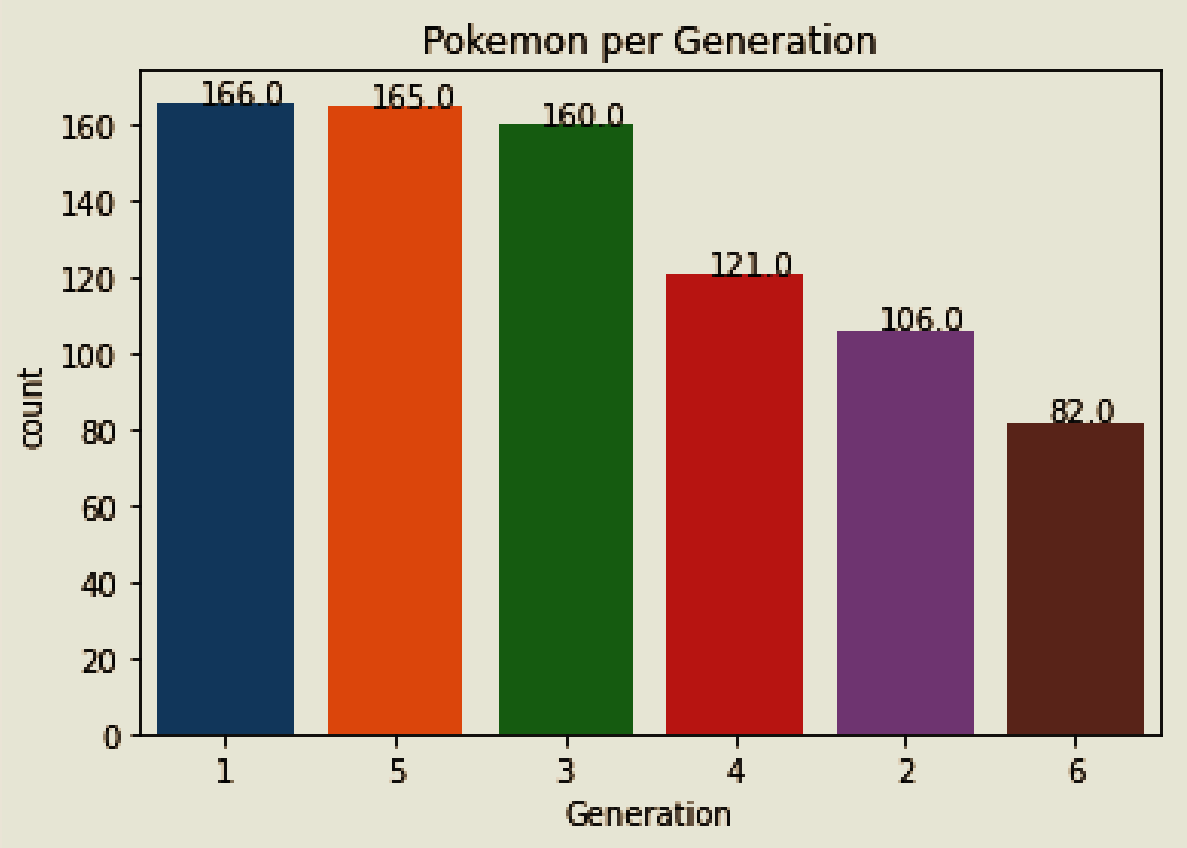
	#	Name	Type1	Type2	Total	HP	Attack	Defense	Sp.Atk	Sp.Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	None	309	39	52	43	60	50	65	1	False

Kolom **Type 1, Type 2, Sp. Atk, dan Sp. Def** sebelumnya terdapat spasi pada format namanya. Untuk mempermudah pemanggilan kolom, hal tersebut harus diganti nama kolomnya atau bisa **dihapus spasinya**.

Sebelumnya diketahui bahwa kolom **Type 2** terdapat 386 missing value. Karena tidak semua pokemon mempunyai Type 2 sehingga tidak bisa diisi dengan nilai yang random. Oleh karena itu **diisi dengan nilai 'None'.**

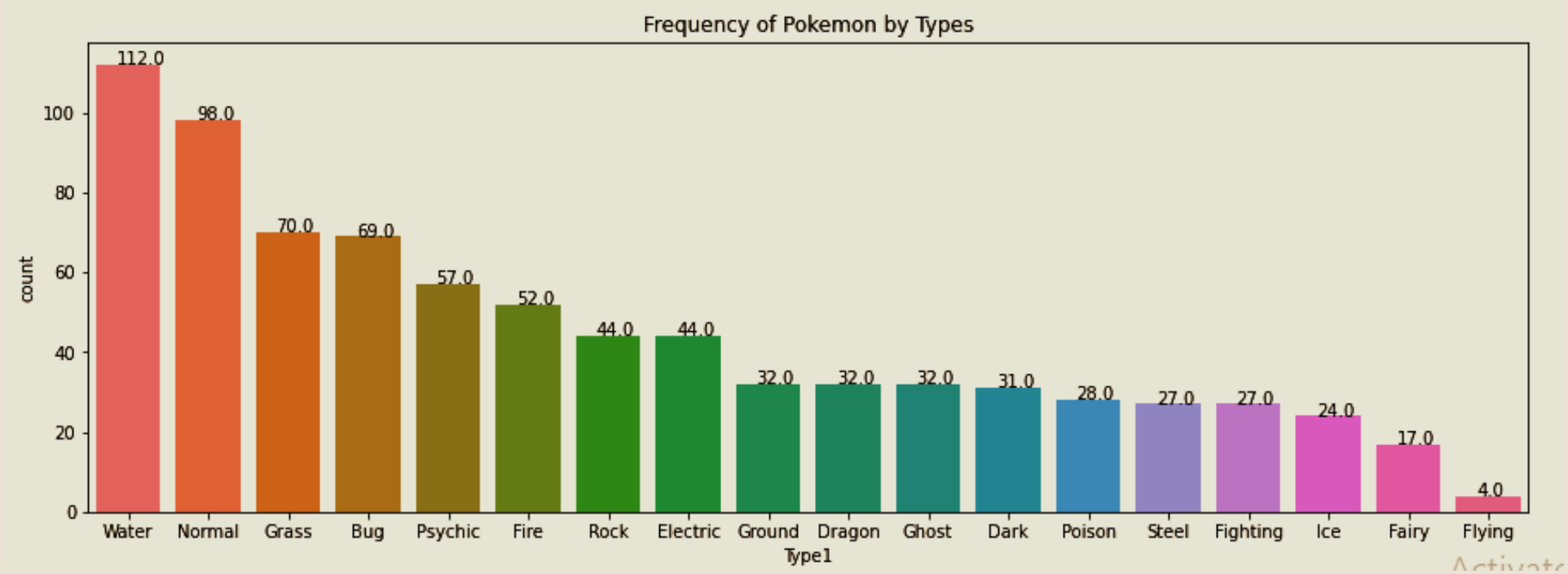
Analysis and Visualization

VISUALIZATION



Terdapat enam generasi pada Pokemon. **Generasi** Pokemon **pertama** mempunyai jumlah Pokemon **paling banyak** yaitu sebesar **166** Pokemon, diikuti Generasi Pokemon kelima yang hanya selisih satu. Sedangkan Generasi pokemon yang mempunyai jumlah **paling sedikit** yaitu **generasi keenam** sebesar **82** pokemon.

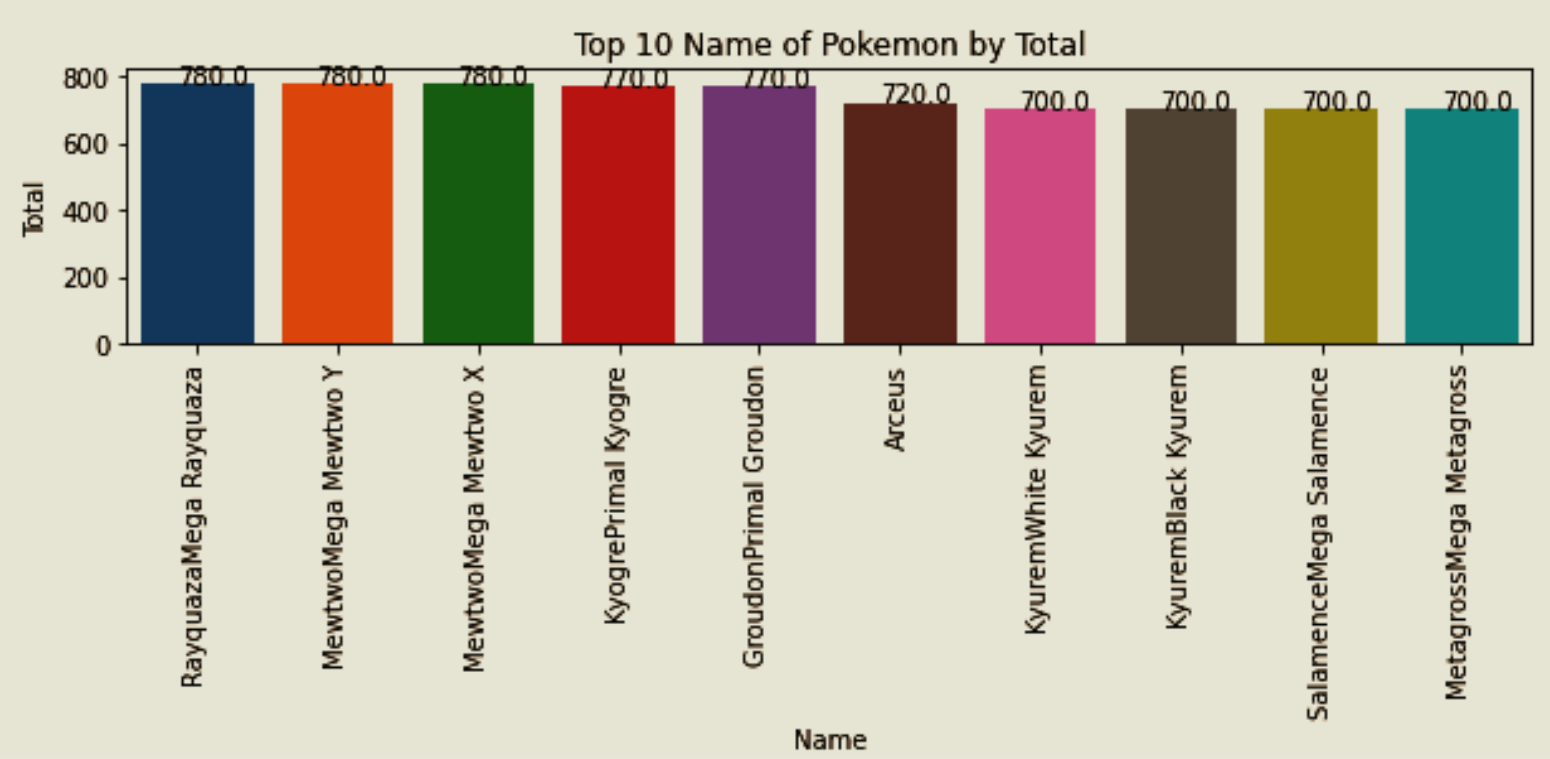
VISUALIZATION



Pokemon yang mempunyai tipe **Water** mempunyai jumlah **paling banyak** yaitu **112** Pokemon.

VISUALIZATION

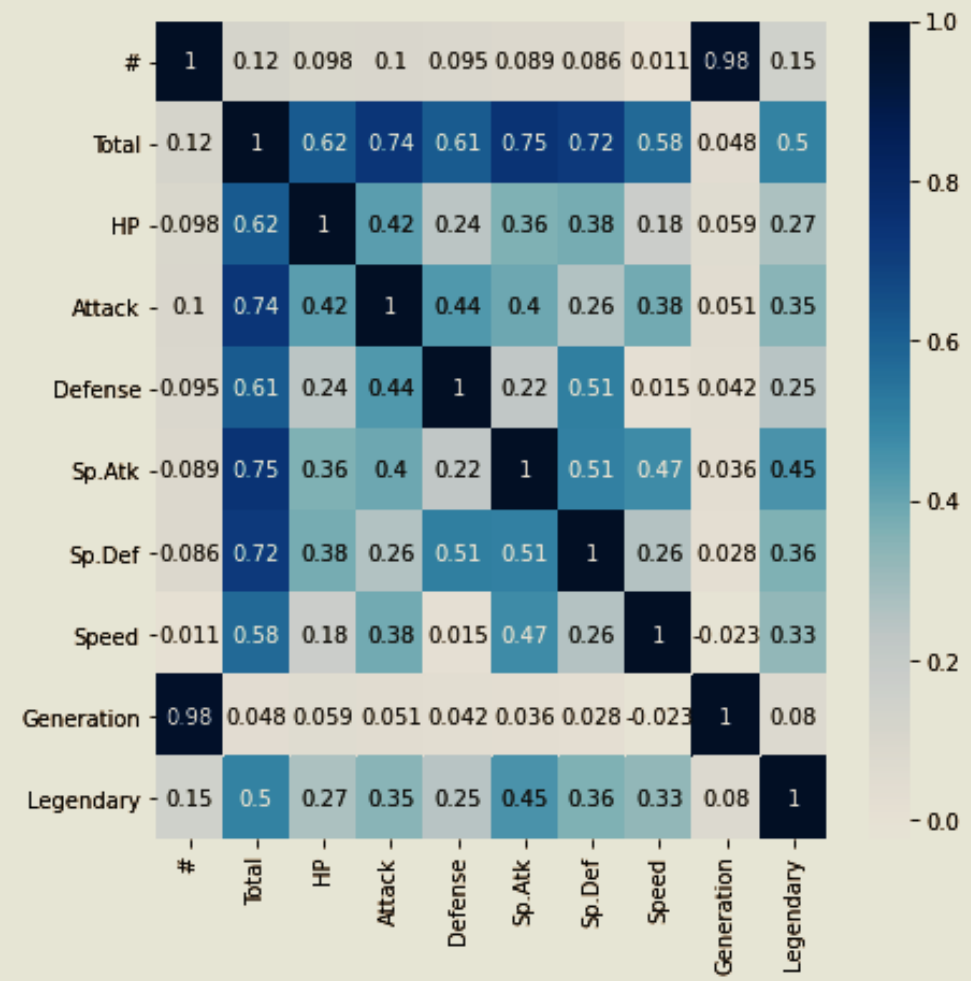
Dibawah ini adalah 10 Pokemon yang mempunyai total tertinggi. Dapat dilihat juga bahwa Pokemon yang bernama Rayquaza Mega Rayquaza, Mewtwo Mega Mewtwo Y, dan Mewtwo Mega Mewtwo X mempunyai total paling banyak yaitu sebesar 780.



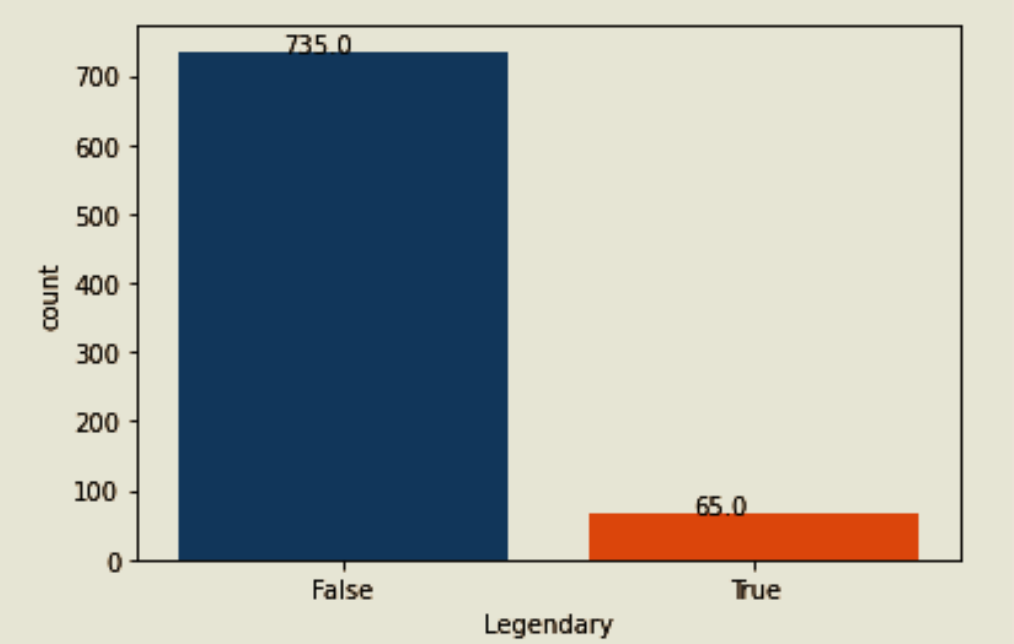
VISUALIZATION

Atribut **Total** mempunyai korelasi yang baik dengan atribut attack dan defense, yaitu:

- Total dengan Attack : 0.74
- Total dengan Defense : 0.61
- Total dengan Sp.Def : 0.72
- **Total dengan Sp.Atk : 0.75**, yang juga merupakan **korelasi tertinggi**



VISUALIZATION



Kolom **Legendary** mempunyai dua output yaitu True dan False. Jumlah **False** jauh **lebih besar** yaitu sebesar **735** daripada nilai **True** yang hanya berjumlah **65**.

PREDICTED ANALYSIS

```
#transformasi data menggunakan MinMaxScaler
from sklearn.preprocessing import MinMaxScaler

scaler=MinMaxScaler(feature_range=(0,1))
x=scaler.fit_transform(x)
x=pd.DataFrame(x)
x.head()
```

	0	1	2	3	4	5	6	7
0	0.230000	0.173228	0.237838	0.195556	0.298913	0.214286	0.228571	0.0
1	0.375000	0.232283	0.308108	0.257778	0.380435	0.285714	0.314286	0.0
2	0.575000	0.311024	0.416216	0.346667	0.489130	0.380952	0.428571	0.0
3	0.741667	0.311024	0.513514	0.524444	0.608696	0.476190	0.428571	0.0
4	0.215000	0.149606	0.254054	0.168889	0.271739	0.142857	0.342857	0.0

Sebelum memproses data menggunakan algoritma KNN perlu dilakukan **normalisasi data**, yaitu menyamakan skala atribut data ke dalam suatu range. Gambar disamping merupakan hasil normalisasi dari **Min Max Normalization**.

PREDICTED ANALYSIS

```
#score dari data training
max_train_score = max (train_score)
train_scores_ind = [i for i, v in enumerate(train_score)
if v == max_train_score]
print('Max train score {} % and k = {}'.format(max_train_score*100, list(map(lambda x:
x+1, train_scores_ind))))
```

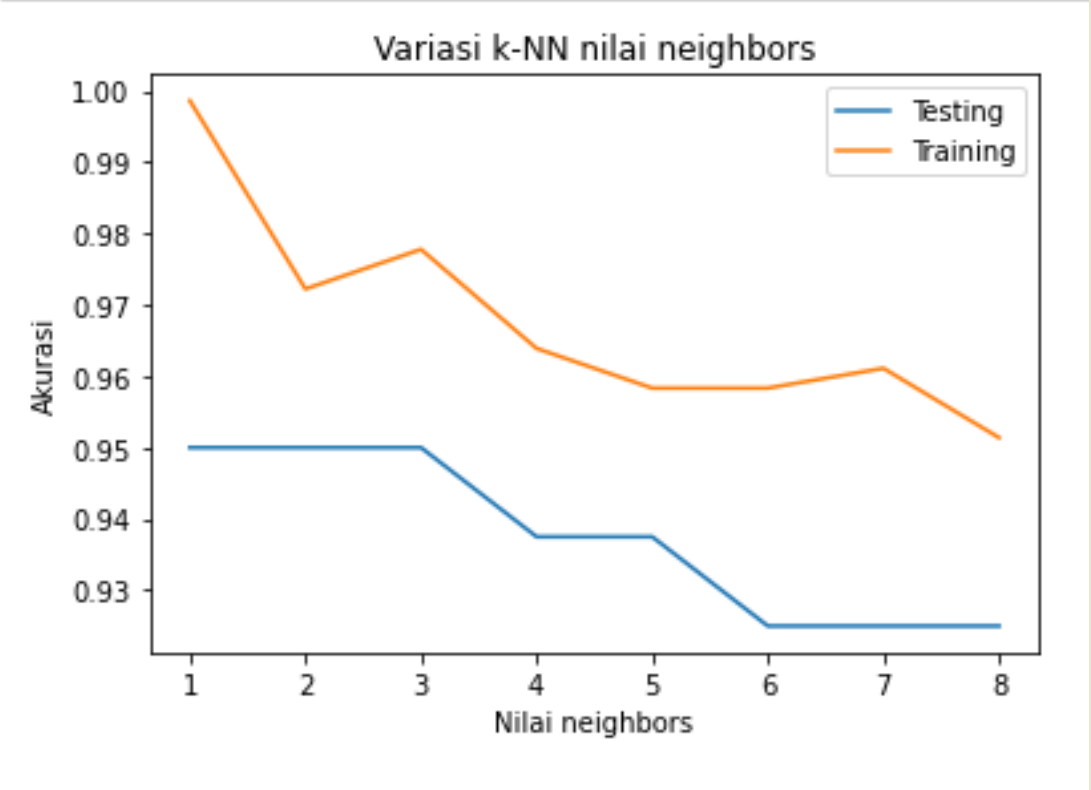
```
Max train score 99.86111111111111 % and k = [1]
```

```
#score dari data testing
max_test_score = max (test_score)
test_scores_ind = [i for i, v in enumerate(test_score)
if v == max_test_score]
print('Max test score {} % and k = {}'.format(max_test_score*100, list(map(lambda x:
x+1, test_scores_ind))))
```

```
Max test score 95.0 % and k = [1, 2, 3]
```

Setelah membagi data menjadi data training dan data testing pada model KNN, dapat diketahui score dari **data training** yaitu **99,861 %** dengan nilai **k=1** dan score **data testing** yaitu **95%** dengan nilai **k=1, k=2, dan k=3**.

PREDICTED ANALYSIS



Gambar di samping merupakan visualisasi **variasi score** dari nilai k pada **data training** (grafik warna oranye) dan **data testing** (grafik warna biru).

PREDICTED ANALYSIS

```
#menyiapkan knn Classifier menggunakan k neighbors
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(x_train,y_train) #Fit modelnya
knn.score(x_test,y_test) #menampilkan knn score
```

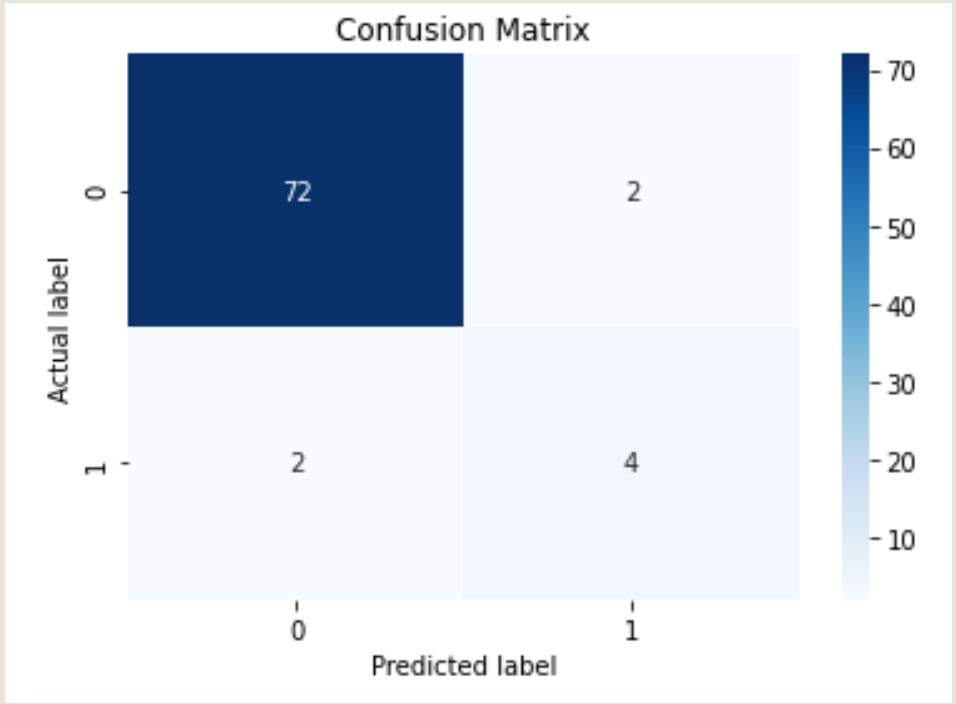
```
0.95
```

Score dari algoritma **KNN** yaitu sebesar **0.95** dengan nilai **k** yaitu **1**. Hal ini juga merupakan representasi akurasi dari nilai algoritma yang digunakan.

PREDICTED ANALYSIS

Berdasarkan hasil akurasi yang diperoleh dari **confusion matrix**, menghasilkan:

a. Terdapat **72** pokemon yang **bukan legendaris** diprediksi **benar**.
b. Terdapat **4** pokemon yang **legendaris** diprediksi **benar**.
c. Terdapat **2** pokemon yang **bukan legendaris** diprediksi **salah**.
d. Terdapat **2** pokemon yang **legendaris** diprediksi **salah**.



THANK YOU


khairinaalsa@gmail.com


linkedin.com/in/khairinaaltaf


github.com/khairinaalsa