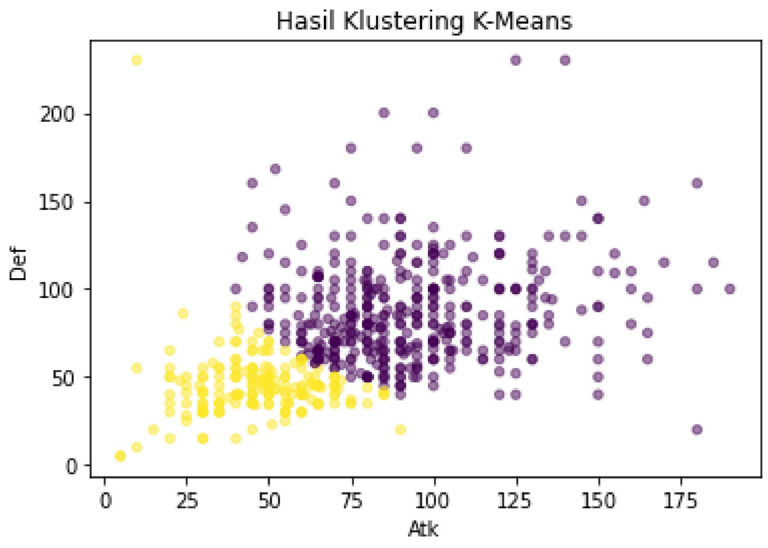
```
import csv
from urllib.request import urlopen, Request
import requests
from bs4 import BeautifulSoup as bs
import pandas as pd
import numpy as np

alamat = "https://pokemondb.net/pokedex/all"
safeAdd = Request(alamat, headers={'User-Agent': 'Mozilla/5.0'})
html = urlopen(safeAdd)
data = bs(html, 'html.parser')

table = data.find("table", {"id":"pokedex"})
rows = data.findAll("tr")

row_data = []
for row in rows:
    cell_data = []

    if row.contents[1].get_text() == "501": #stop function
        break

    for item in row.findAll(["th","td"]): #gathering function
        cell_data.append(item.get_text())
    row_data.append(cell_data)

df = pd.DataFrame(row_data)
df.columns = df.iloc[0]
df = df[1:]
df.to_csv('data_pokemon.csv',index=False)


from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
data = pd.read_csv("data_pokemon.csv")

data["Attack"] = pd.to_numeric(data["Attack"])
data["Defense"] = pd.to_numeric(data["Defense"])

#menambahkan dua kolom tranformasi
data["Alog"] = np.log(data["Attack"])
data["Dlog"] = np.log(data["Defense"])

log_data = data.iloc[:, 10:12]
log_array = np.array(log_data)

kmeans = KMeans(n_clusters=2, random_state=200)
kmeans.fit(log_array)
data['kluster'] = kmeans.labels_
```

```
plt.scatter(data.Attack, data.Defense, s = 20, c = data.kluster, marker = "o", alpha = 0.5)
plt.title("Hasil Klustering K-Means")
plt.xlabel("Atk")
plt.ylabel("Def")
plt.show()

data.to_csv('data_cluster.csv',index=False)
print("Jumlah K terbaik adalah K=2")
```



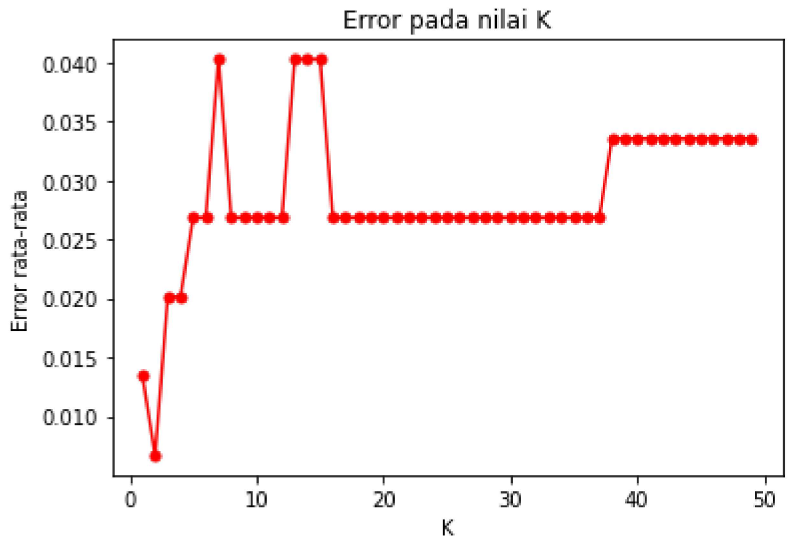
```
Jumlah K terbaik adalah K=2
```

```
df = pd.read_csv('data_cluster.csv')
df.head()
```

	#	Name	Type	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Alog	Dlog
0	1	Bulbasaur	Grass Poison	318	45	49	49	65	65	45	3.891820	3.891820
1	2	Ivysaur	Grass Poison	405	60	62	63	80	80	60	4.127134	4.143135
2	3	Venusaur	Grass Poison	525	80	82	83	100	100	80	4.406719	4.418841

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import accuracy_score

X = df[['Alog', 'Dlog']]
y = df['kluster']

x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.25,random_state=150) #
```

```
error = []
for i in range(1, 50):
    model_k = KNeighborsClassifier(n_neighbors=i)
    model_k.fit(x_train, y_train)
    y_pred = model_k.predict(x_test)
    error.append(np.mean(y_pred != y_test))

plt.figure(1)
plt.plot(range(1, 50), error, color='red', marker='o', markersize=5)
plt.title('Error pada nilai K')
plt.xlabel('K')
plt.ylabel('Error rata-rata')
plt.show()
```

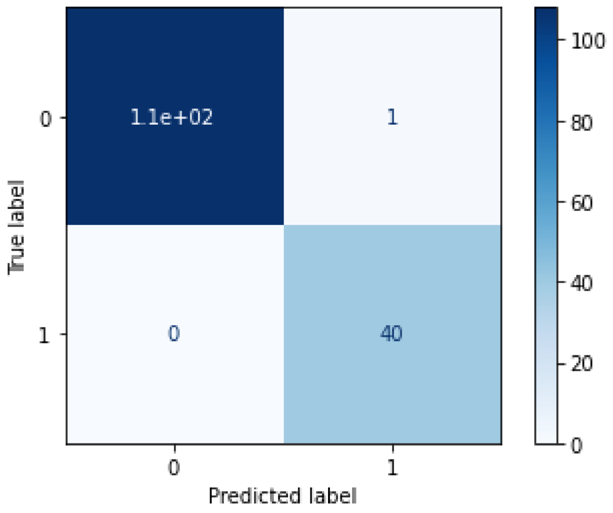


```
model_k = KNeighborsClassifier(n_neighbors=2)
model_k.fit(x_train, y_train)
y_pred = model_k.predict(x_test)
print(classification_report(y_test, y_pred))
print(accuracy_score(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       109
           1       0.98      1.00      0.99        40

    accuracy                           0.99       149
   macro avg       0.99      1.00      0.99       149
weighted avg       0.99      0.99      0.99       149

0.9932885906040269
```

```
disp = plot_confusion_matrix(model_k, x_test, y_test,
                             cmap=plt.cm.Blues,
                             )
print(disp.confusion_matrix)
```

```
[[108   1]
 [  0  40]]
```



```
from sklearn.model_selection import KFold
from sklearn.neighbors import KNeighborsClassifier
import numpy as np

from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier

knn= KNeighborsClassifier(n_neighbors=2)
score= cross_val_score(knn, X, y, cv=5, scoring='accuracy')
print(score)
print(score.mean())
```

```
[0.97478992 1.         0.96638655 0.97478992 0.99152542]
0.9814983620566871
```

Kesimpulan: Hasil tidak jauh beda antara train_test_split dengan prediksi menggunakan k-fold cross validation. Namun hasil k-fold dirasa lebih reliable dibandingkan dengan train test split biasa, terbukti dari hasil k-fild menghasilkan hasil yang beragam mulai dari akurasi 97% hingga 100%