

## KNN – K-FOLD VALIDATION

Menggunakan data pokemon yang telah diberi label, analisis dilakukan pada model menggunakan metode K-NN dan k-fold cross validation. Bandingkan hasil saat kita langsung menggunakan K-NN dan menggunakan k-nn + k-fold cross validation.

### Langkah 1: Membaca Data CSV dan Mempersiapkan Data Untuk Training

Code:

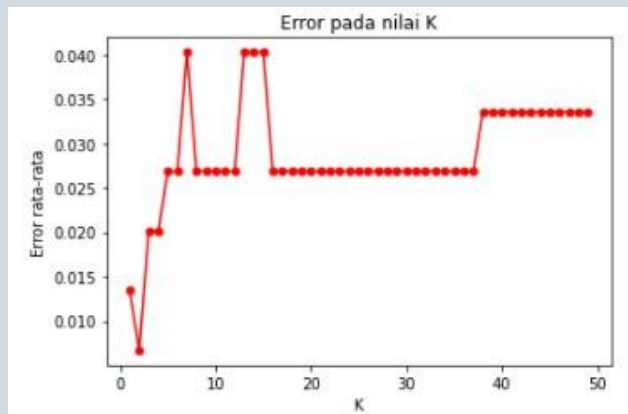
```
#membaca data kluster CSV
knn_data = pd.read_csv("data_kluster_pokemon.csv")
#mengambil 2 atribut yang diperlukan
set_data = knn_data.iloc[:, 10:12]
kluster_data = knn_data.iloc[:, 12]
#persiapan data untuk training
x_train, x_test, y_train, y_test = train_test_split(set_data, kluster_data, test_size=0.25, random_state=150) # 75% training and 25% test
```

### Langkah 2: Mencari K terbaik pada metode K-NN

Code:

```
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()
```

Output:



#### NOTE:

Diagram menunjukkan nilai error terendah terjadi ketika k = 2. Maka diperoleh nilai k terbaik adalah 2

### Langkah 3: Mencari nilai akurasi dengan K-NN (k=2)

Code:

```
model_k = KNeighborsClassifier(n_neighbors=2)
#model belajar
model_k.fit(x_train, y_train)
#test model
y_pred = model_k.predict(x_test)
#evaluasi
print('\n Hasil confussion matrix dengan KNN')
print(confusion_matrix(y_test, y_pred))
print('\n Hasil classification report dengan KNN')
print(classification_report(y_test, y_pred))
```

Output:

```
Hasil confussion matrix dengan KNN
[[108  1]
 [ 0 40]]

Hasil classification report dengan KNN
```

	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

Hasil akurasi sangat baik yaitu 99%.

Hasil confusion matrix menunjukkan hanya metode pembelajaran ini hanya mengalami jumlah prediksi yang salah sebanyak 1 kali yaitu pada bagian false negative.

### Langkah 4: Mencari nilai akurasi dengan K-NN + K-fold cross validation (k=5)

Code:

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

Output:

```
0.9842696629213481
```

Hasil akurasi sangat baik yaitu 98%.

### **ANALISA PERBANDINGAN KNN VS KNN+ K-FOLD CROSS VALIDATION:**

Hasil pada perhitungan akurasi menunjukkan bahwa metode K-Nearest Neighbor lebih unggul dengan nilai 99% tanpa K-Fold Cross Validation dibandingkan dengan K-Fold Cross Validation yang menghasilkan 98%.