

# Tugas01-10 Praktikum Mandiri

SYAHRI GHIFARI MAULIDI 0110222217

<sup>1</sup> Teknik Informatika, STT Terpadu Nurul Fikri, Depok

## 1.1 Persiapan Data

```
▶ import pandas as pd
  import numpy as np
  from sklearn.neighbors import KNeighborsClassifier
  from sklearn.model_selection import cross_val_score

  # Data training
  data = {
    'Temperature': [10, 25, 15, 20, 18, 20, 22, 24],
    'Wind_Speed': [0, 0, 5, 3, 7, 10, 5, 6],
    'Perception': ['Dingin', 'Panas', 'Dingin', 'Panas', 'Dingin', 'Dingin', 'Panas', 'Panas']
  }

  df = pd.DataFrame(data)

  X = df[['Temperature', 'Wind_Speed']]
  y = df['Perception']
```

Data training disiapkan dalam format DataFrame yang terdiri dari fitur temperatur (10-25°C), kecepatan angin (0-10 km/jam), dan variabel target persepsi (Dingin/Panas). Data dibagi menjadi feature set (X) yang berisi kolom Temperature dan Wind\_Speed, serta target variable (y) yang berisi kolom Perception, dengan proses ini memastikan data terstruktur dengan baik untuk proses training model KNN.

## 1.2 Pemilihan Parameter Optimal

```

▶ best_k = 0
best_score = 0
from sklearn.model_selection import LeaveOneOut # Import LeaveOneOut

loo = LeaveOneOut() # Create a LeaveOneOut object
for k in [1, 3, 5, 7]:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, X, y, cv=loo) # Use LeaveOneOut object
    accuracy = np.mean(scores)
    print(f'k={k}, Accuracy: {accuracy:.1%}')

    if accuracy > best_score:
        best_score = accuracy
        best_k = k

print(f'\nNilai k optimal: {best_k} dengan akurasi {best_score:.1%}')


...
k=1, Accuracy: 100.0%
k=3, Accuracy: 75.0%
k=5, Accuracy: 75.0%
k=7, Accuracy: 0.0%


Nilai k optimal: 1 dengan akurasi 100.0%

```

Dilakukan evaluasi empat nilai k berbeda (1, 3, 5, 7) menggunakan Leave-One-Out Cross Validation yang memberikan akurasi masing-masing 100%, 75%, 75%, dan 0%. Hasil tuning mengidentifikasi k=1 sebagai parameter optimal dengan akurasi sempurna 100%, meskipun dalam praktiknya nilai k=3 atau k=5 sering lebih dipilih untuk menghindari overfitting pada data training yang berukuran kecil.

### 1.3 Prediksi Data Baru

```

knn_best = KNeighborsClassifier(n_neighbors=best_k)
knn_best.fit(X, y)
prediction = knn_best.predict([[16, 3]])[0]
print(f'Prediksi untuk [16, 3]: {prediction}')


Prediksi untuk [16, 3]: Dingin
/usr/local/lib/python3.12/dist-packages/sklearn/utils/v
warnings.warn(

```

Model KNN dengan k=1 dilatih menggunakan seluruh data training kemudian dilakukan prediksi pada kondisi baru temperatur 16°C dan kecepatan angin 3 km/jam. Hasil prediksi menunjukkan persepsi "Dingin" yang mengindikasikan bahwa pada kondisi tersebut, model

mengklasifikasikan berdasarkan similarity terdekat dengan data historis yang memiliki karakteristik serupa.