

BAB IV HASIL DAN PEMBAHASAN

4.1. Penyajian Data Uji Coba

Bab ini akan menjelaskan hasil penelitian penerapan metode K-nearest neighbor pada klasifikasi profesi di MAN 1 Probolinggo. Langkah-langkah penelitian telah dijelaskan pada bab sebelumnya dan akan diimplementasikan pada bab ini. Tahapan tersebut meliputi hasil pengumpulan data set, implementasi metode k-NN, pengujian, dan penarikan kesimpulan dari hasil implementasi metode.

4.2. Pengumpulan Dataset

Pengumpulan data di dalam penelitian ini memerlukan data nilai ijazah dan data nilai hasil penjurusan kelas siswa yang berjumlah 500. Dari 500 data siswa terdapat 3 kelas yaitu kelas Bahasa (IBB), MIPA(IPA), dan IPS. Data siswa yang diambil ialah data siswa pada tahun 2018 sampai 2020. Sedangkan dari 500 data tersebut dikelompokkan menjadi 2 kelompok data yaitu 400 data *training* dan 100 data *testing*.

4.3. Implementasi Metode *K-Nearest Neighbor*

Implementasi menggunakan metode k-nearest neighbor dilakukan dengan membuat skrip Python. Pelaksanaannya memiliki beberapa tahapan :

a. Pada tahap ini, kumpulan data siap untuk diproses menggunakan k-NN. Hasil konversi data pada tabel excel menjadi tabel csv akan dipanggil menggunakan script Python menggunakan k-NN pada tahap klasifikasi, yang dapat dilihat pada bagian program 4.2. Sebelum memanggil data csv, pertama-tama impor pustaka yang terlihat di segmen program 4.1 berikut:

Segmen Program 4.1 Library

```
import numpy as np #Array
import pandas as pd #membaca file csv
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV
```

Menginput data csv dengan menggunakan pandas sebagai librarynya.

Segmen Program 4.2 Input data csv

```
nilai = pd.read_csv('Data siswa.csv')
nilai.head()
```

Selanjutnya menuju proses pemisahan data yaitu data training dan data testing, yang dilakukan pada Segmen Program 4.3.

Segmen Program 4.3 Pemisah Data

```
X_train = nilai[['X2', 'X3', 'X4', 'X5']]
y_train= nilai['Y']
print(X_train)
print(y_train)
```

Menggunakan 80% data training dan 20% data testing.

Segmen Program 4.3 Penggunaan Dataset

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_train,
                                                    y_train, test_size=0.20, random_state=0)

print ('Train set:', X_train.shape, y_train.shape)
print ('Test set:', X_test.shape, y_test.shape)
```

Kemudian dilakukan klasifikasi k-NN, terlihat pada segmen program 4.4.

Segmen Program 4.4 Klasifikasi k-NN

```
from sklearn.neighbors import KNeighborsClassifier #library
KNN

k =4

knn = KNeighborsClassifier(n_neighbors = k, metric='euclidean').fit(X_train,y_train)
```

Tahap uji coba merupakan tahapan hasil menguji cobakan data testing dengan metode k-NN yang sudah diimplementasikan. Pada bagian tahapan ini akan mengetahui tingkat akurasi dengan menggunakan metode k-Nearest Neighbor (KNN) di segmen program 4.5.

Segmen Program 4.5 Hasil Akurasi Prediksi

```
from sklearn import metrics

print("Train set accuracy: ", metrics.accuracy_score(y_train, knn.predict(X_train)))

print("Test set accuracy: ", metrics.accuracy_score(y_test, ypred))
```

4.4. Uji Coba

Hasil uji coba dengan 100 data. Mencari parameter k terbaik dengan memanfaatkan GridSearchCV

Segmen Program 4.5 Mencari k terbaik

```
param_grid= {'n_neighbors': np.arange(1, 100)}

knn = GridSearchCV(KNeighborsClassifier(), param_grid, scoring='accuracy')

knn.fit(X_train, y_train)

knn.best_params_
```

Segmen Program 4.6 Lanjutan(Mencari k terbaik)

```
from sklearn.neighbors import KNeighborsClassifier #library
KNN

k = 46

knn = KNeighborsClassifier(n_neighbors = k, metric='euclidean').fit(X_train,y_train)

from sklearn import metrics

print("Train set accuracy: ", metrics.accuracy_score(y_train, knn.predict(X_train)))
```

Hasil uji coba dengan 100 data pada tabel 4.1 berikut

Tabel 4.1 Hasil Uji Coba 100 Data

| Data Uji Coba | Banyak Data IPA | Banyak Data IPS | Banyak Data Bahasa |
|---------------|--------------------|--------------------|-----------------------|
| 100 | 39 | 18 | 43 |