

DATA MINING



Disusun Oleh:

I Putu Juliarta Arya Utama	1705551031
I Made Widiananda Putra Winarta	1705551034
I Gede Angga Kusuma Putra	1705551051
Ketut Nila Arta	1705551059

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS UDAYANA

2020

PENJELASAN KLASIFIKASI DATA MENGGUNAKAN PYTHON

Program python ini menggunakan tiga algoritma untuk melakukan klasifikasi data yaitu algoritma KNN (K-Nearest Neighbor), Naïve Bayes dan Decision Tree. Data yang digunakan untuk klasifikasi adalah data iris yang berformat csv. Berikut merupakan penjelasan dari program python untuk klasifikasi data iris.

```
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import _pickle as pickle
import requests
import json
import numpy as np
import pandas as pd
from os import system, name
```

Kode Program 1 Import library

Kode program 1 merupakan library yang diimport menggunakan Bahasa pemrograman python yang berfungsi untuk membantu proses klasifikasi data iris yang dimana library tersebut akan digunakan sebagai fungsi untuk mengolah data iris agar dapat diklasifikasi dengan efektif.

```
# fungsi utama dalam program
def main():
# import dataset iris dari csv menggunakan pandas
dataset = pd.read_csv('iris.csv')
pd.DataFrame(dataset)
```

Kode Program 2 Import dataset iris

Kode program 2 adalah penjelasan mengenai menggunakan library pandas yang berfungsi untuk memasukan data dari suatu sumber ke data frame sehingga data lebih mudah diolah. Kode

program diatas menjelaskan dimana library panda membaca data iris.csv dan dimasukan kedalam variabel dataset yang selanjutnya data tersebut dibuat menjadi data frame. Semua kode program dimasukan kedalam fungsi utama yaitu def main() agar program berjalan lebih efisien.

```
# mengurangi 1 kolom terakhir
X = dataset.iloc[:, :-1]
# menampilkan data variabel dependen
Y = dataset.iloc[:, 4]
```

Kode Program 3 Memilih data yang diperlukan

Kode program 3 merupakan fungsi untuk membaca dataset yang sebelumnya sudah dimasukan ke dalam data frame menggunakan library panda. Fungsi dari kode program diatas adalah untuk memilih data yang akan diolah dari dataset iris yang sebelumnya sudah di load menggunakan pandas. Variabel x berfungsi untuk mengurangi 1 kolom terakhir pada data iris yang mengambil data panjang dan lebar sepal serta panjang dan lebar petal pada dataset bunga iris. Sedangkan variabel Y berfungsi untuk mengambil kolom terakhir yaitu kolom spesies dari bunga sesuai data irisnya.

```
label_encoder = LabelEncoder()
dataset['species'] = label_encoder.fit_transform(dataset['species'])
dataset['species'].unique()
label_encoder_y = LabelEncoder()
Y = label_encoder_y.fit_transform(Y)
```

Kode Program 4 Koversi label menjadi angka

Kode program 4 berfungsi untuk melakukan konversi label spesies data bunga iris yang sebelumnya sudah dipilih. Konversi tersebut berguna untuk memudahkan pengolah data dan konversi yang dilakukan adalah mengubah spesies setosa menjadi 0, spesies iris-color menjadi 1 dan virginca menjadi 2.

```
# membagi data training dan testing
X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, test_size=0.3, random state=42)
```

Kode Program 5 Membagi data testing dan training

Kode program 5 berfungsi untuk melakukan pembagian atau split data yang akan digunakan dalam melakukan klasifikasi menggunakan tiga algoritma berbeda. Dimana yang digunakan sebagai data training sebesar 0.7 dari keseluruhan data dan data testing sebesar 0.3 dari keseluruhan data.

```
# mengubah skala data.
scale_X = StandardScaler()
X_train = scale_X.fit_transform(X_train)
X_test = scale_X.transform(X_test)
```

Kode Program 6 Normalisasi data

Kode program 6 merupakan kode program yang berfungsi untuk melakukan konversi data panjang dan lebar sepal serta panjang dan lebar petal menjadi satuan yang memiliki jarak yang berdekatan sehingga hasil klasifikasi data yang diharapkan menjadi semakin tinggi akurasi dari prediksinya.

```
# menu input untuk memilih algoritma klasifikasi data iris
print('Pilih Algoritma untuk klasifikasi data iris : \n')
print('1. Naive Bayes\n' +
      '2. K-Nearest Neighbor\n' +
      '3. Decision Tree')
```

Kode Program 7 Menu input algoritma

Kode program 7 merupakan kode program yang berfungsi untuk menampilkan menu input algoritma yang akan digunakan untuk mengolah data iris yang terdiri dari tiga algoritma yaitu Naïve Bayes, K-nearest neighbor dan decision tree.

```
# memanggil fungsi inputNumber
var = inputNumber("Silahkan Masukan Angka : ")
if(var == 1):
    model = GaussianNB()
    name = 'Naive Bayes'
elif(var == 2):
    model = KNeighborsClassifier(n_neighbors=3)
    name = 'K-Nearest Neighbor'
elif(var == 3):
    model = DecisionTreeClassifier()
    name = 'Decision Tree'
else:
    system('cls')
    print('Masukan Angka sesuai pilihan.')
```

```
main()
```

Kode Program 8 Input data pilihan algoritma

Kode program 8 merupakan kode program yang memanggil fungsi validasi inputan angka. Setelah mendapatkan inputan dari user setelah itu inputan tersebut akan masuk ke dalam kondisi dimana jika 1 maka akan menjalankan model naïve bayes , jika 2 akan menjalankan model k-nearest neighbor dan jika 3 maka akan menjalankan model decision tree serta jika tidak sesuai inputan maka user akan dikembalikan ke menu awal karena inputan tidak sesuai.

```
# fungsi validasi untuk input menu klasifikasi
def inputNumber(message):
    while True:
        try:
            userInput = int(input(message))
        except ValueError:
            print("Not an integer! Try again.")
            continue
        else:
            return userInput
        break
```

Kode Program 9 Fungsi validasi menu input

Kode program 9 berfungsi untuk melakukan validasi untuk menu pilihan klasifikasi dimana jika user memasukan input tidak berupa integer atau angka maka fungsi ini akan mengeluarkan eror bahwa inputan yang dimasukan user tidak interger maka harus memasukan inputan kembali. Jika inputan sudah benar berupa angka maka fungsi tersebut akan menghasilkan output berupa variabel userInput yang berisi angka yang diinput user.

```
# Memasukkan data training pada fungsi klasifikasi
train = model.fit(X_train, Y_train)
```

Kode Program 10 Memasukan data traning ke Model

Kode program 10 berfungsi untuk melakukan input data ke dalam masing – masing model yang akan digunakan untuk melakukan klasifikasi data iris dimana menggunakan dua parameter yaitu data yang sudah dinormalisasi sebelumnya.

```
# Menentukan hasil prediksi dari x_test
y_pred = train.predict(X_test)

# Menentukan probabilitas hasil prediksi
```

```
train.predict_proba(X_test)
```

Kode Program 11 Hasil prediksi dan probabilitasnya

Kode program 11 berfungsi untuk menampilkan hasil prediksi dari model algoritma yang digunakan dalam melakukan klasifikasi data iris dan juga untuk menampilkan tingkat probabilitas dari hasil prediksi tersebut.

```
# tampilkan hasil ketepatan nilai precision
print(classification_report(Y_test, y_pred))

# menampilkan akurasi model knn
accuracy = accuracy_score(Y_test, y_pred)*100
print('Accuracy of our model is equal ' + str(round(accuracy, 2)) + '%.')
# memanggil fungsi untuk menampilkan pilihan kembali ke menu utama atau keluar
dari program
back()
```

Kode Program 12 Menampilkan hasil dan tingkat akurasi

Kode program 12 merupakan fungsi untuk menampilkan dan menghitung akurasi dari model yang digunakan dalam melakukan klasifikasi data iris . Precision berfungsi untuk menampilkan array dari data yang diprediksi oleh model dan Accuracy score berfungsi untuk menghitung tingkat akurasi suatu model sesuai dengan hasil prediksinya serta selanjutnya setelah menampilkan hasil akan diarahkan kepada tampilan pilihan yang berisi menu kembali ke menu awal atau keluar dari program.

```
# fungsi untuk menu pilihan kembali ke menu awal atau keluar dari program
def back():
    var = input('Klasifikasi dengan metode lain ? [yes/no] : ')

    if var == 'yes':
        system('cls')
        main()

    elif var == 'no':
        exit(0)

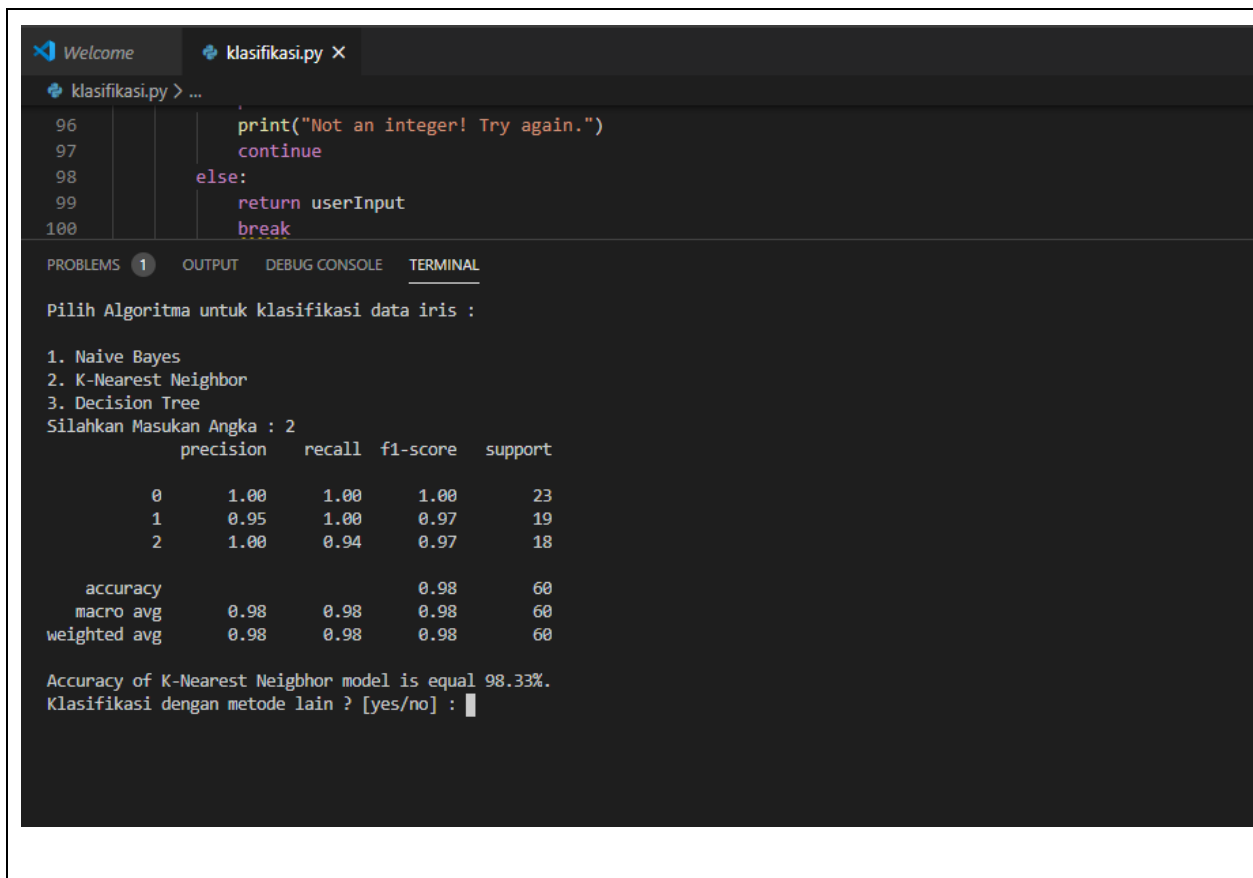
    else:
        print('inputan salah!!!')
```

```
back()
```

Kode Program 13 Fungsi Menu back

Kode program 13 berfungsi untuk menampilkan menu dan melakukan validasi terhadap inputan dari user. Jika user memasukkan inputan yes maka user akan diarahkan kembali ke tampilan awal program dan jika no maka user akan keluar dari program. Sedangkan jika user tidak menginputkan antara kedua kondisi diatas maka user akan mendapat eror yaitu inputan salah dan akan melakukan input kembali.

1. HASIL ALGORITMA KNN (K-Nearest Neighbor)



```
96         print("Not an integer! Try again.")
97         continue
98     else:
99         return userInput
100    break
```

Pilih Algoritma untuk klasifikasi data iris :

1. Naive Bayes
2. K-Nearest Neighbor
3. Decision Tree

Silahkan Masukan Angka : 2

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	0.95	1.00	0.97	19
2	1.00	0.94	0.97	18
accuracy			0.98	60
macro avg	0.98	0.98	0.98	60
weighted avg	0.98	0.98	0.98	60

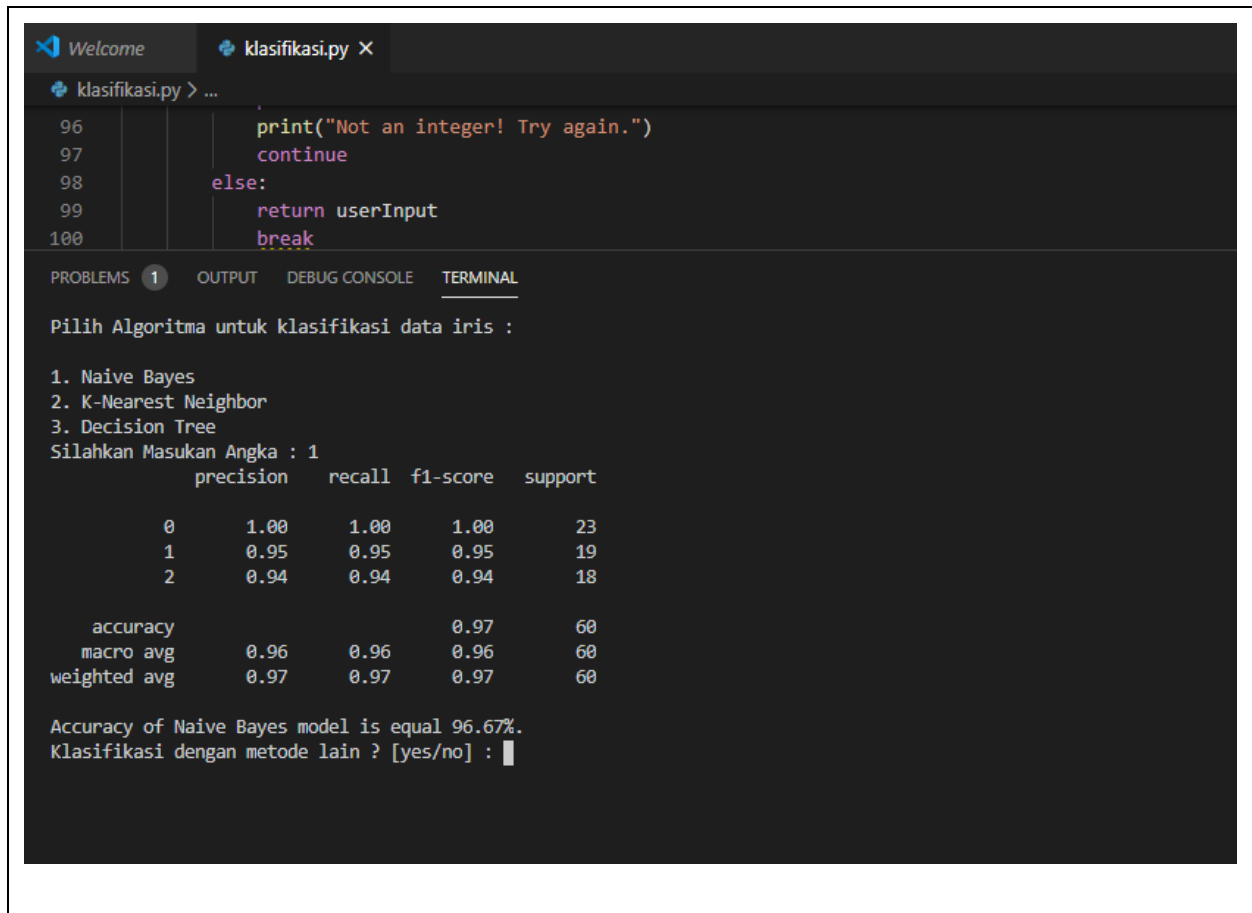
Accuracy of K-Nearest Neighbor model is equal 98.33%.
Klasifikasi dengan metode lain ? [yes/no] :

Gambar 1 Hasil Model algoritma KNN

Gambar 1 merupakan hasil dari algoritma KNN yang berisi tingkat hasil klasifikasi report yang berisi nilai untuk masing-masing spesiesnya dengan 4 parameter yaitu precision, recall, f1-score, dan support dan untuk algoritma KNN berhasil mendapat skor sempurna dalam dalam

prediksi karena semua prediksi yang dihasilkan oleh algoritma ini sesuai dengan data iris awal sehingga model algoritma KNN mendapat akurasi 100%.

2. HASIL ALGORITMA NAÏVE BAYES



```
96         print("Not an integer! Try again.")
97         continue
98     else:
99         return userInput
100    break
```

Pilih Algoritma untuk klasifikasi data iris :

1. Naive Bayes
2. K-Nearest Neighbor
3. Decision Tree

Silahkan Masukan Angka : 1

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	0.95	0.95	0.95	19
2	0.94	0.94	0.94	18
accuracy			0.97	60
macro avg	0.96	0.96	0.96	60
weighted avg	0.97	0.97	0.97	60

Accuracy of Naive Bayes model is equal 96.67%.
Klasifikasi dengan metode lain ? [yes/no] : █

Gambar 2 Hasil algoritma naïve bayes

Gambar 2 merupakan hasil dari algoritma Naïve Bayes yang berisi tingkat hasil klasifikasi report yang berisi nilai untuk masing-masing spesiesnya dengan 4 parameter yaitu precision, recall, fi-score, dan support dan untuk algoritma Naïve Bayes berhasil mendapat skor hampir sempurna dalam dalam prediksi karena hampir semua prediksi yang dihasilkan oleh algoritma ini sesuai dengan data iris awal sehingga model algoritma Naïve Bayes mendapat akurasi 97.78%.

3. ALGORITMA DECISION TREE

```
96     print("Not an integer! Try again.")
97     continue
98 else:
99     return userInput
100    break
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE **TERMINAL**

Pilih Algoritma untuk klasifikasi data iris :

1. Naive Bayes
2. K-Nearest Neighbor
3. Decision Tree

Silahkan Masukan Angka : 3

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	0.95	0.95	0.95	19
2	0.94	0.94	0.94	18
accuracy			0.97	60
macro avg	0.96	0.96	0.96	60
weighted avg	0.97	0.97	0.97	60

Accuracy of Decision Tree model is equal 96.67%.
Klasifikasi dengan metode lain ? [yes/no] : █

Gambar 3 Hasil algoritma decision tree

Gambar 3 merupakan hasil dari algoritma Decision Tree yang berisi tingkat hasil klasifikasi report yang berisi nilai untuk masing-masing spesiesnya dengan 4 parameter yaitu precision, recall, fi-score, dan support dan untuk algoritma Decision Tree berhasil mendapat skor hamper sempurna dalam dalam prediksi karena beberapa prediksi yang dihasilkan oleh algoritma ini tidak sesuai dengan data iris awal sehingga model algoritma Decision tree mendapat akurasi 96.67%.