

NIM : 2111523024

Tangga : Sabtu, 25 Maret 2023

Nama : Khairin Nisa

Asisten :

Mata : Praktikum Data Mining

Kuliah

1. Bobby Darmawan
2. Dwisuci Insani Karimah
3. Iqbal Fitrahul Ramadhan
4. Iqbal Manazil Yuni
5. Muhammad Afif
6. M. Rayhan Rizaldi

Modul : 05

Kelas : 02/B

K-Nearest Neighbors (KNN)

1. Import library pandas dan numpy

```
In [1]: import numpy as np
import pandas as pd
```

2. Masukkan dataset

```
In [7]: prediket = pd.read_csv("lulus (1).csv")
prediket.head()

Out[7]:
```

	IPK	Pelatihan Penguasaan (K)	Pemahaman	Formasi Komunikasi (K)	Regulasi Organisasi	Lulus Capaian
0	2.28	0	1	2	4	0
1	3.12	0	0	1	1	0
2	2.28	0	4	2	3	0
3	3.18	2	0	0	2	1
4	3.10	1	0	4	2	0

3. Menentukan variable-varibel independen dan dependen

```
In [8]: x = prediket.drop(["lulus (1)"], axis=1)
In [9]: y = prediket["lulus (1)"]
```

4. Import package model selection dari SKlearn

```
In [11]: from sklearn.model_selection import train_test_split
```

5. Membagi data training dan data testing

```
In [12]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30)
```

6. Mengaktifkan package StandardScaler

```
In [14]: from sklearn.preprocessing import StandardScaler
standard_scaler = StandardScaler()
standard_scaler.fit(x_train)
x_train = standard_scaler.transform(x_train)
x_test = standard_scaler.transform(x_test)
```

7. Mengaktifkan package dan mengaktifkan fungsi klasifikasi untuk klasifikasi KNN

```
In [17]: from sklearn.neighbors import KNeighborsClassifier
In [18]: knn = KNeighborsClassifier(n_neighbors=4)
```

8. Memasukkan data training pada fungsi klasifikasi untuk KNN

```
In [17]: knn.fit(x_train, y_train)
Out[17]:
NeighborsClassifier
NeighborsClassifier(n_neighbors=4)
```

9. Menentukan prediksi KNN dan probabilitas

```
In [18]: y_predict = knn.predict(x_test)
Out[18]: array([1, 1, 1, 1, 0, 1, 1, 0, 0], dtype=int64)
```

10. Mengimport package untuk melihat keakuratan data hasil prediksi dengan data actual dan Menampilkan matriks hasil prediksi

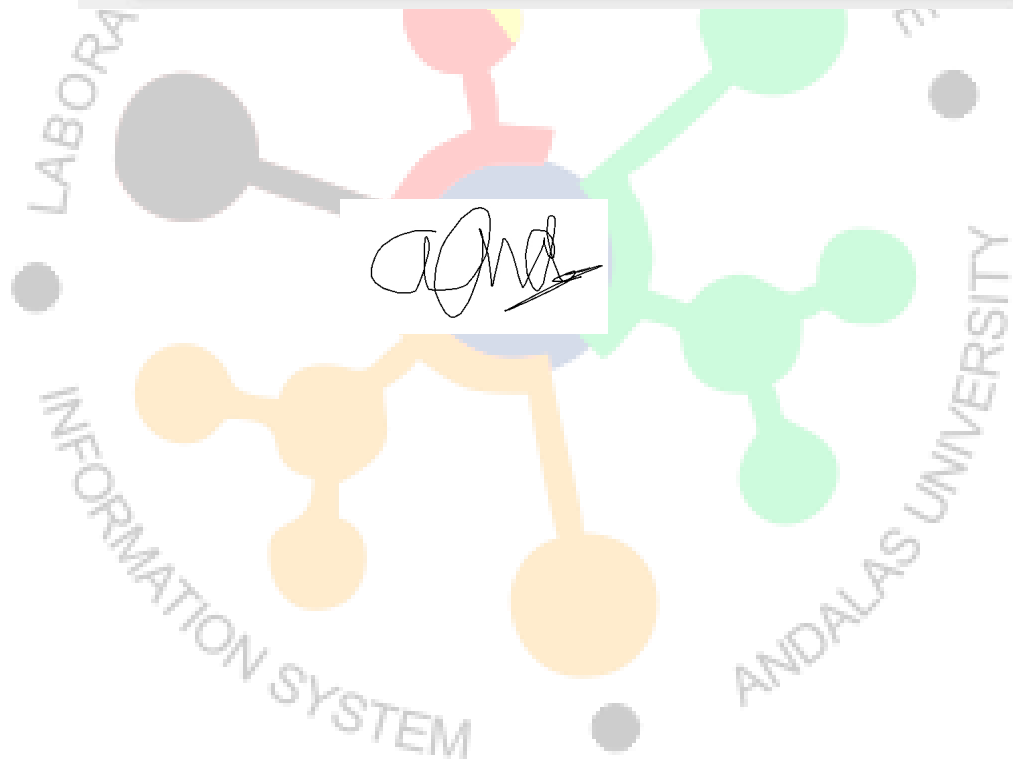
```
In [22]: from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_predict))
```

```
[[0 0]
 [4 1]]
```

11. Mengukur ketepatan atau keakuratan hasil prediksi

```
In [23]: print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1
1	0.83	0.50	0.67	9
accuracy			0.56	10
macro avg	0.42	0.25	0.33	10
weighted avg	0.75	0.50	0.60	10



Handwritten signature