Logo

Description automatically generated with medium confidence

Pembelajaran Mesin

( Praktikum )

TI -C4

Fakultas Vokasi

Universitas Airlangga

[152111283042] | [NELA ANJANI P P A] | [05 November 2023]

**TUGAS INDIVIDU DATA IRIS SVM**

1. menampilkan data iris

import numpy as np

import pandas as pd

from sklearn import datasets

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.svm import SVC

df = pd.read\_csv('IRIS.csv')

print(" Data Awal ".center(100, "="))

print(df)

print("============================================================")

# Separate features (x) and target (y)

X = df.drop(columns=['sepal\_length','sepal\_width','petal\_length','petal\_width'])

print(X)

y = df['species']

print(y)

Hasil



1. grouping variable

print("GROUPING VARIABEL".center(100, "="))

X=df.iloc[:,0:4].values

y=df.iloc[:,4].values

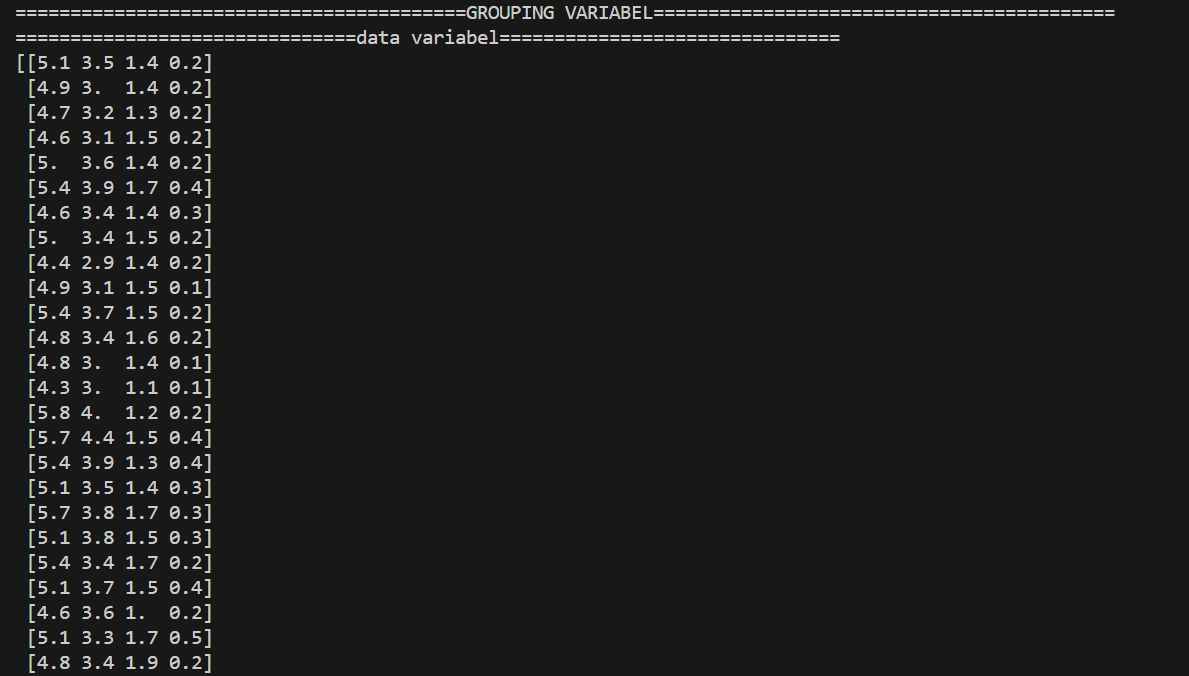
print("data variabel".center(75,"="))

print(X)

print("data kelas".center(75,"="))

print(y)

Hasil



1. membagi data training dan testing

# Split data into training and testing

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=100)

print("Instance variabel data training".center(75, "="))

print(x\_train)

print("Instance kelas data training".center(75, "="))

print(y\_train)

print("Instance variabel data testing".center(75, "="))

print(x\_test)

print("Instance kelas data testing".center(75, "="))

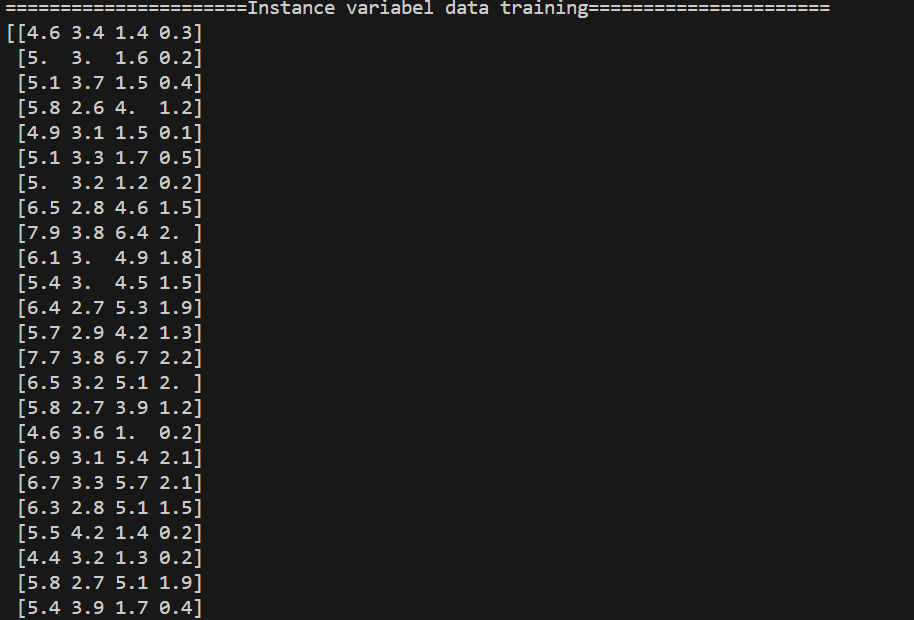
print(y\_test)

print("============================================================")

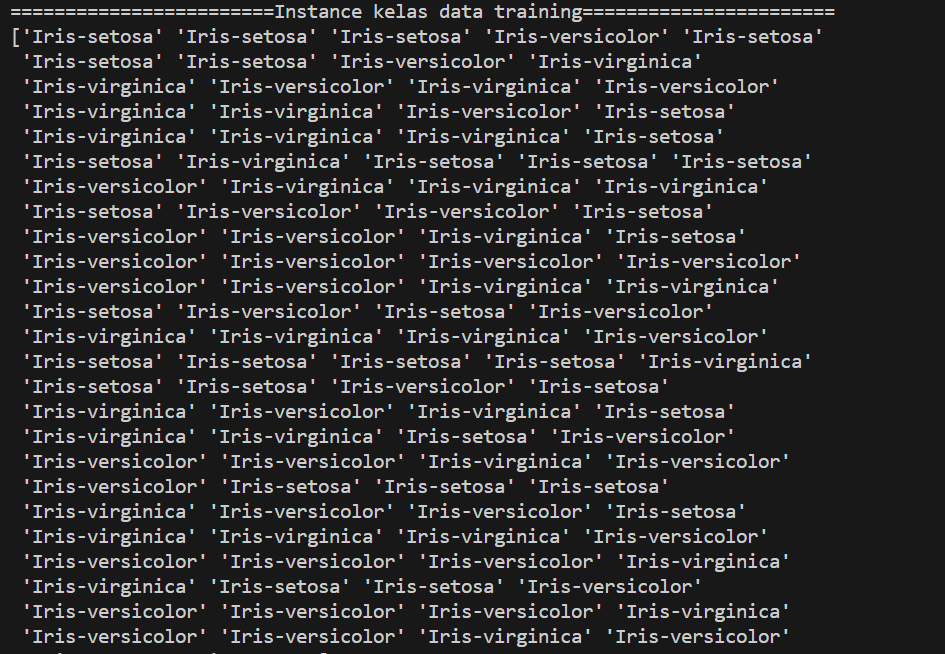
print()

Hasil

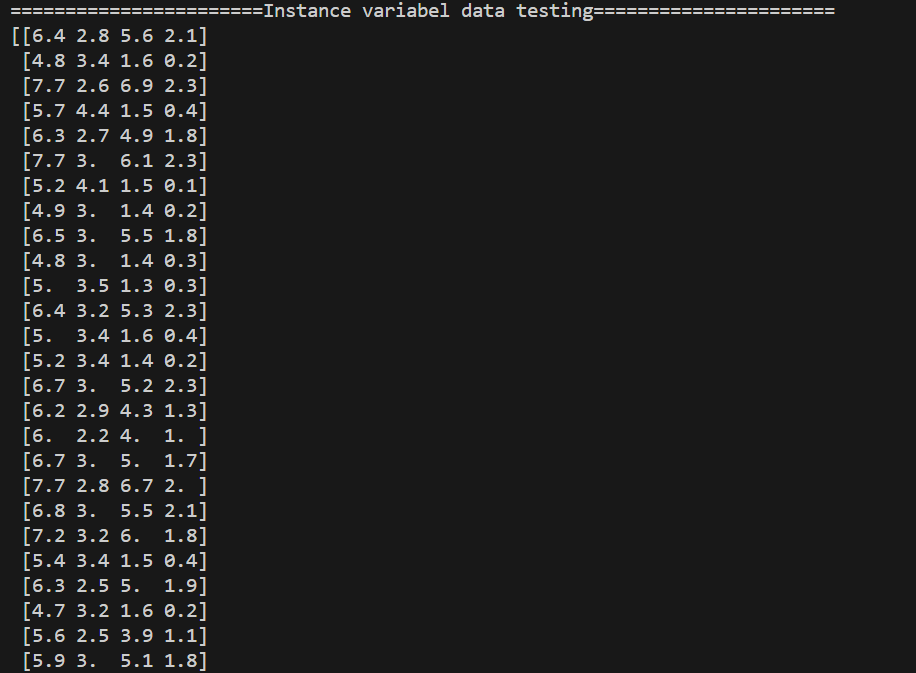
1. Variable data training



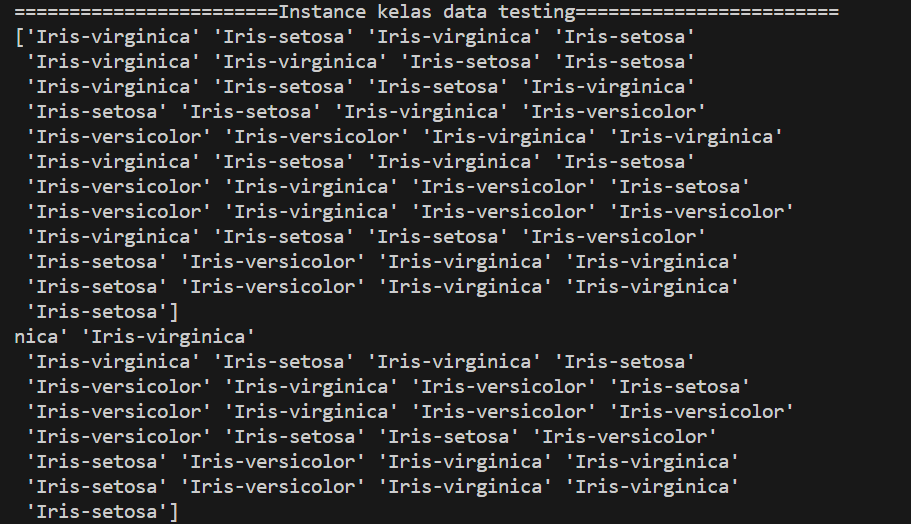
1. Instance kelas data training



1. Variable data testing



1. Instance kelas data testing



1. Hasil confusion Matrix

#Pemodelan SVM

model = SVC()

model.fit(x\_train, y\_train)

y\_pred = model.predict(x\_test)

print("hasil prediksi SVM")

print(y\_pred)

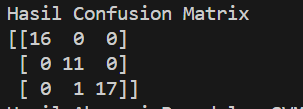
#Evaluasi Confusion Matrix dan Evaluasi Akurasi Python

print("Hasil Confusion Matrix")

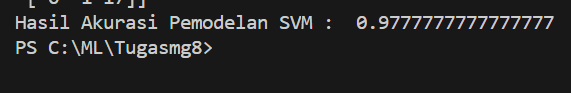
print(confusion\_matrix(y\_test, y\_pred))

print("Hasil Akurasi Pemodelan SVM : ", accuracy\_score(y\_test, y\_pred))

Hasil



1. Hasil Akurasi pemodelan SVM



**TUGAS KELOMPOK LANJUTAN UTS SVM DAN NN**

1. menampilkan data awal

import pandas as pd

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.neural\_network import MLPClassifier

# Import MLPClassifier from scikit-learn

# Load data from Excel file

df = pd.read\_csv('lung\_cancer\_examples.csv')

print(" Data Awal ".center(100, "="))

print(df)

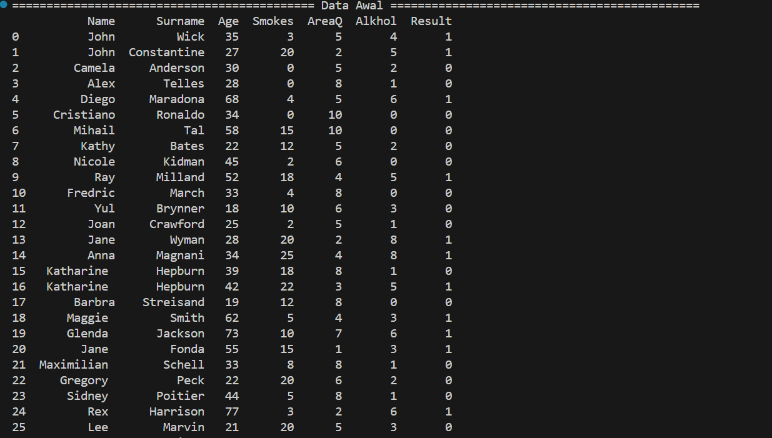
print("============================================================")

# Separate features (x) and target (y)

x = df.drop(columns=['Name','Surname','Age','Smokes','AreaQ','Alkhol',])

y = df['Result']

Hasil



1. Grouping 2 variable

# Grouping yang dibagi menjadi dua

print("GROUPING VARIABEL".center(100, "="))

X=df.iloc[:,0:6].values

y=df.iloc[:,6].values

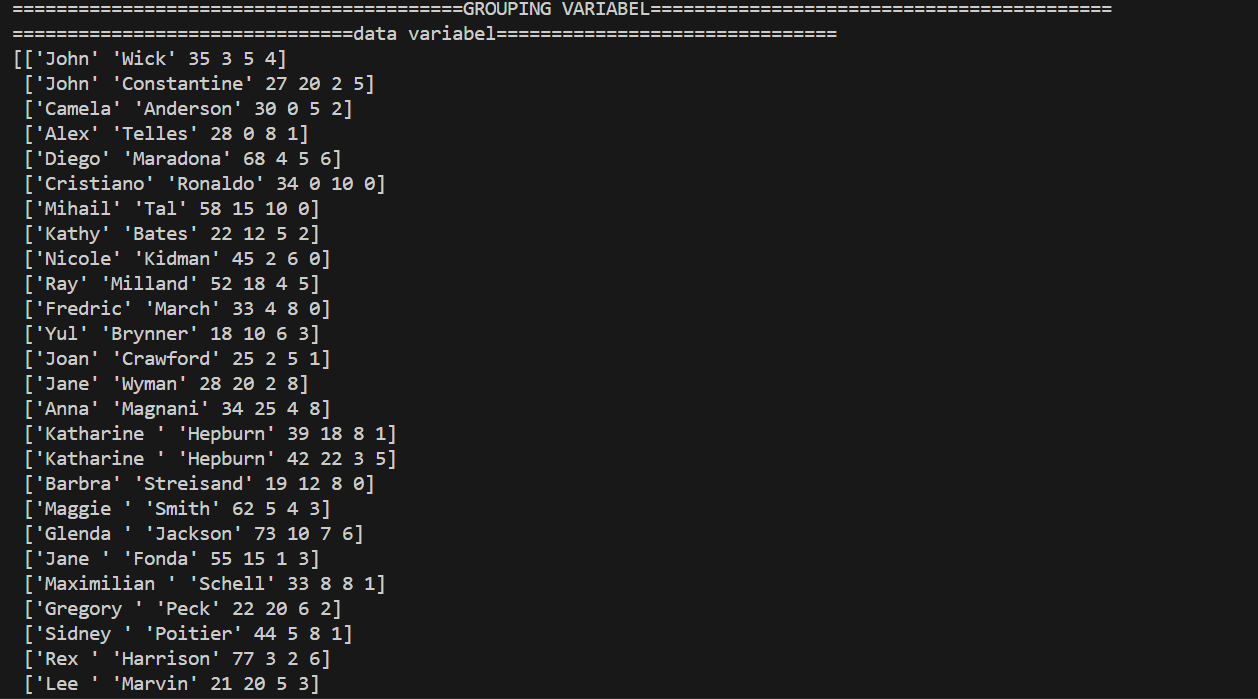
print("data variabel".center(75,"="))

print(X)

print("data kelas".center(75,"="))

print(y)

Hasil



1. Pembagian data training dan testing

# Split data into training and testing

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.30, random\_state=100)

print("Instance variabel data training".center(75, "="))

print(x\_train)

print("Instance kelas data training".center(75, "="))

print(y\_train)

print("Instance variabel data testing".center(75, "="))

print(x\_test)

print("Instance kelas data testing".center(75, "="))

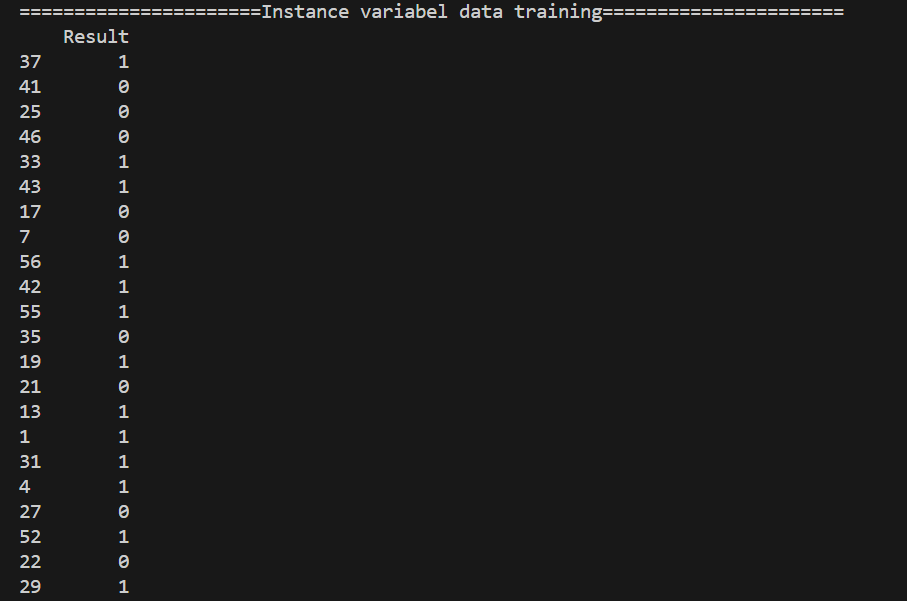
print(y\_test)

print("=====================================================")

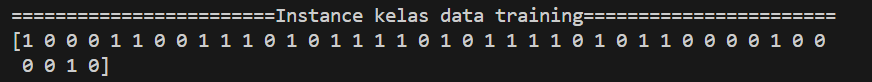
print()

Hasil

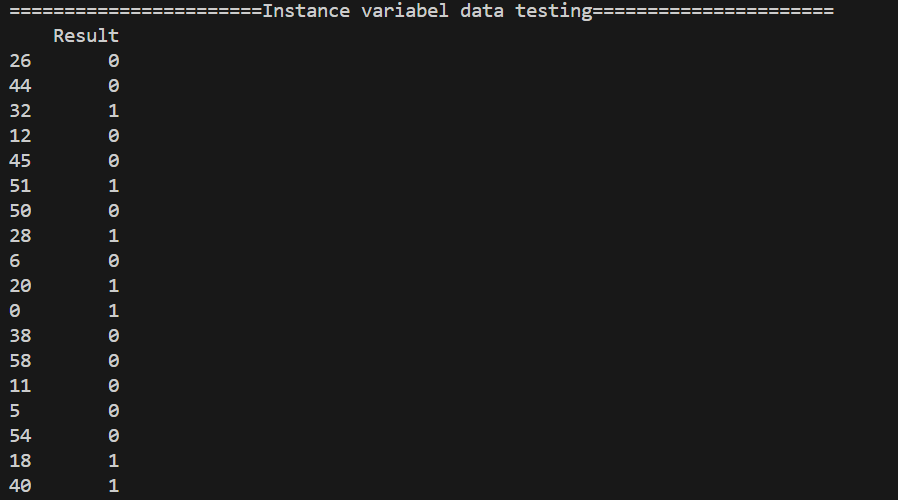
1. Variable data training



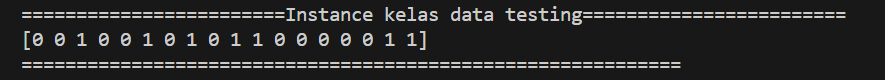
1. Kelas data training



1. Variable data testing



1. Kelas data testing



1. Cofusion Matrix SVM

# Evaluasi confusion matrix dan evaluasi akurasi SVM

print("Hasil confusion matrix SVM".center(75, "="))

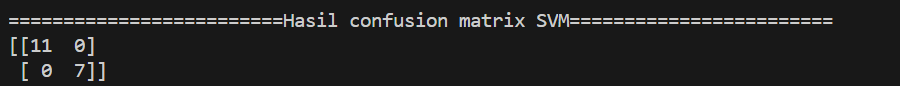
print(confusion\_matrix(y\_test, y\_pred\_svm))

print("Hasil akurasi pemodelan SVM:", accuracy\_score(y\_test, y\_pred\_svm))

print("====================================================")

print()

Hasil



1. Hasil prediksi SVM dan Akurasi

# Pemodelan SVM

svm\_model = SVC()

svm\_model.fit(x\_train, y\_train)

y\_pred\_svm = svm\_model.predict(x\_test)

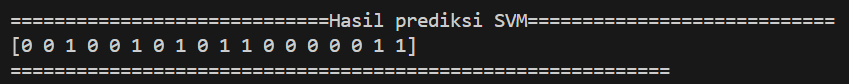
print("Hasil prediksi SVM".center(75, "="))

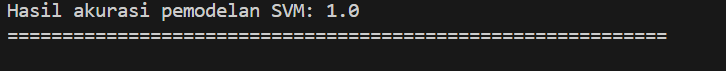
print(y\_pred\_svm)

print("======================================================")

print()

Hasil





1. Cofusion Matrix NN

# Evaluasi confusion matrix dan evaluasi akurasi Neural Network

print("Hasil confusion matrix Neural Network".center(75, "="))

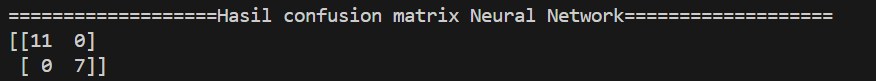
print(confusion\_matrix(y\_test, y\_pred\_nn))

print("Hasil akurasi pemodelan Neural Network:", accuracy\_score(y\_test, y\_pred\_nn))

print("======================================================")

print()

Hasil



1. Hasil prediksi NN dan Akurasi

# Pemodelan Neural Network (Multilayer Perceptron)

nn\_model = MLPClassifier(hidden\_layer\_sizes=(100, 50), max\_iter=1000, random\_state=100)  # Adjust the parameters as needed

nn\_model.fit(x\_train, y\_train)

y\_pred\_nn = nn\_model.predict(x\_test)

print("Hasil prediksi Neural Network".center(75, "="))

print(y\_pred\_nn)

print("=====================================================")

print()

Hasil

