**NAÏVE BAYES**

**Source Code :**

**Python**

import numpy as np

import pandas as pd

import seaborn as sns

from sklearn import datasets

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score ,precision\_score,recall\_score,f1\_score

from sklearn.naive\_bayes import GaussianNB

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

import sklearn.metrics

from sklearn.ensemble import RandomForestClassifier

from collections import Counter

import matplotlib.pyplot as plt

import matplotlib.colors as colors

import sklearn.model\_selection as model\_selection

from sklearn.metrics import (confusion\_matrix, accuracy\_score,

                             f1\_score, ConfusionMatrixDisplay,

                             classification\_report)

pd.options.mode.chained\_assignment = None

#membaca data

dataframe = pd.read\_excel(r"D:\Pengajaran\2022\Genap\Pembelajaran Mesin (Prakt)\Naive Bayes\BlaBla.xlsx")

data=dataframe[['A','B','C',

                'D','E','F',

                'G','H',

                'I','J',

                'K','L',

                'M','N']]

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

print(data)

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

#pengecekan missing value

print("pengecekan missing value".center(75,"="))

print(data.isnull().sum())

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

#grouping yang dibagi menjadi dua

print("GROUPING VARIABEL".center(75,"="))

X=data.iloc[:,0:13].values

y=data.iloc[:,13].values

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

print(X)

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

print(y)

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

#pembagian training dan testing

print("SPLITTING DATA 20-80".center(75,"="))

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=0)

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()

#pemodelan naive bayes

print("PEMODELAN DENGAN NAIVE BAYES".center(75,"="))

gaussian = GaussianNB()

gaussian.fit(X\_train, y\_train)

Y\_pred = gaussian.predict(X\_test)

accuracy\_nb=round(accuracy\_score(y\_test,Y\_pred)\* 100, 2)

acc\_gaussian = round(gaussian.score(X\_train, y\_train) \* 100, 2)

print("instance prediksi naive bayes:")

print(Y\_pred)

#perhitungan confusion matrix

cm = confusion\_matrix(y\_test, Y\_pred)

print('CLASSIFICATION REPORT NAIVE BAYES'.center(75,'='))

#Mendapat Akurasi

accuracy = accuracy\_score(y\_test, Y\_pred)

# Mendapat Akurasi

precision = precision\_score(y\_test, Y\_pred)

# Menampilkan recision    recall  f1-score   support

print(classification\_report(y\_test, Y\_pred))

cm = confusion\_matrix(y\_test, Y\_pred)

TN = cm[1][1] \* 1.0

FN = cm[1][0] \* 1.0

TP = cm[0][0] \* 1.0

FP = cm[0][1] \* 1.0

total = TN + FN + TP + FP

sens = TN / (TN + FP) \* 100

spec = TP / (TP + FN) \* 100

print('Akurasi : ', accuracy \* 100, "%")

print('Sensitivity : ' + str(sens))

print('Specificity : ' + str(spec))

print('Precision : ' + str(precision))

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

print()

#Menampilkan Confusion Matrix

cm\_display=ConfusionMatrixDisplay(confusion\_matrix=cm)

print('Confusion matrix for Naive Bayes\n',cm)

f, ax = plt.subplots(figsize=(8,5))

sns.heatmap(confusion\_matrix(y\_test, Y\_pred), annot=True, fmt=".0f", ax=ax)

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

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

print()

#COBA INPUT

A = int(input("Umur Pasien = "))

print("Isi Jenis kelamin dengan 0 jika Perempuan dan dan 1 jika Laki-Laki")

B = input("Jenis Kelamin Pasien = ")

print("Isi Y jika mengalami dan N jika tidak")

C = input("Apakah pasien mengalami C? = ")

D = input("Apakah pasien mengalami D? = ")

E = input("Apakah pasien mengalami E? = ")

F = input("Apakah pasien mengalami F? = ")

G = input("Apakah pasien mengalami G? = ")

H = input("Apakah pasien mengalami H? = ")

I = input("Apakah pasien mengalami I? = ")

J = input("Apakah pasien mengalami J? = ")

K = input("Apakah pasien mengalami K? = ")

L = input("Apakah pasien mengalami L? = ")

M = input("Apakah M? = ")

umur\_k = 0

A\_k = 0

B\_k = 0

if A<21:

   A\_k=1

if A>20 and A<31:

   A\_k=2

if A>30 and A<41:

   A\_k=3

if A>40 and A<51:

   A\_k=4

if A>50:

   A\_k=5

print("kode umur pasien adalah",A\_k)

if B=="P":

   B\_k=1

else:

   B\_k=0

if C=="Y":

   C=1

else:

   C=0

if D=="Y":

    D=1

else:

   D=0

if E=="Y":

   E=1

else:

   E=0

if F=="Y":

   F=1

else:

   F=0

if G=="Y":

   G=1

else:

   G=0

if H=="Y":

   H=1

else:

   H=0

if I=="Y":

   I=1

else:

   I=0

if J=="Y":

   J=1

else:

   J=0

if K=="Y":

   K=1

else:

   K=0

if L=="Y":

   L=1

else:

   L=0

if M=="Y":

   M=1

else:

   M=0

Train = [A\_k,B\_k,C,D,E,F,G,

         H,I,J,K,L,M]

print(Train)

test = pd.DataFrame(Train).T

predtest = gaussian.predict(test)

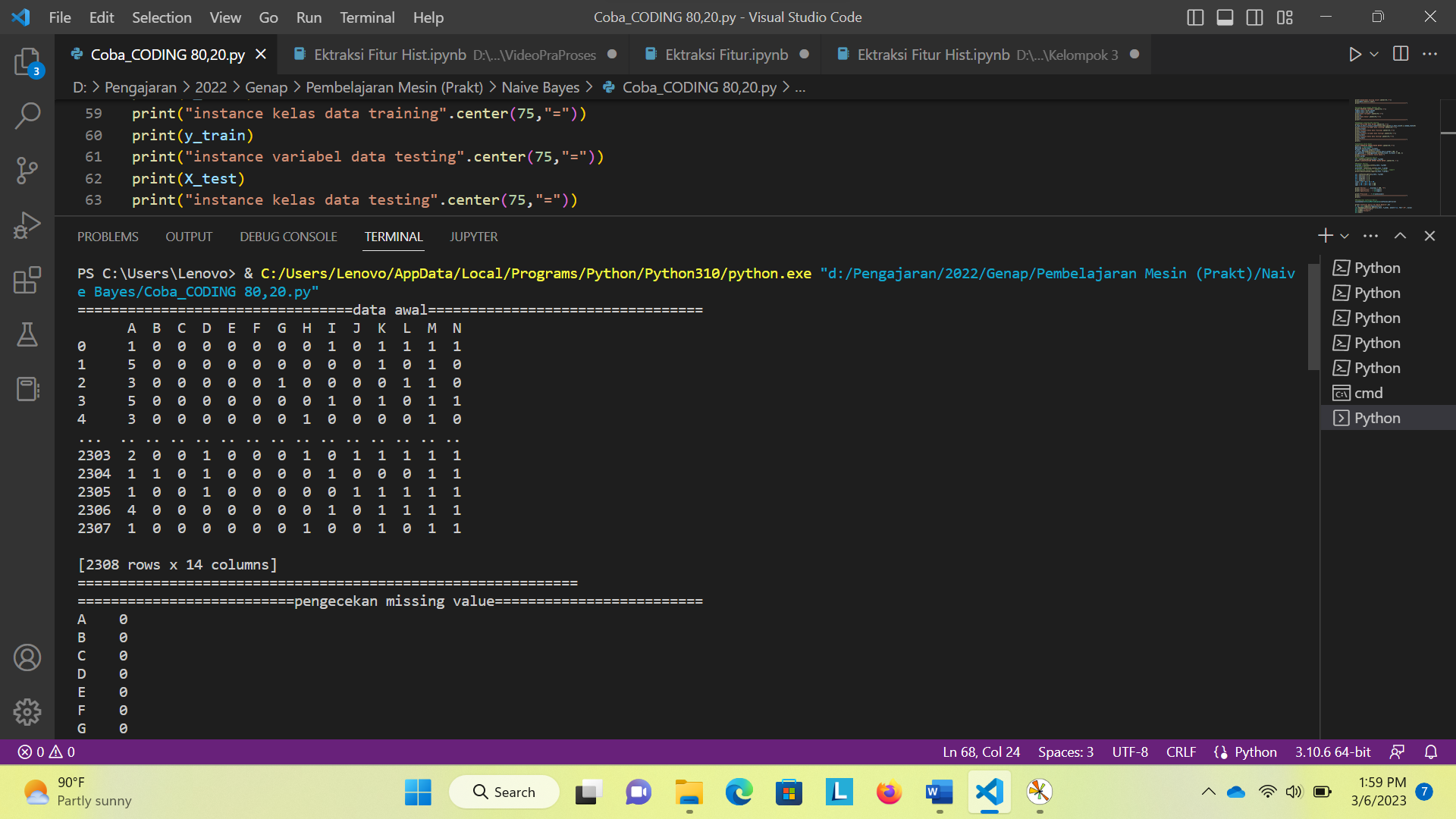
if predtest==1:

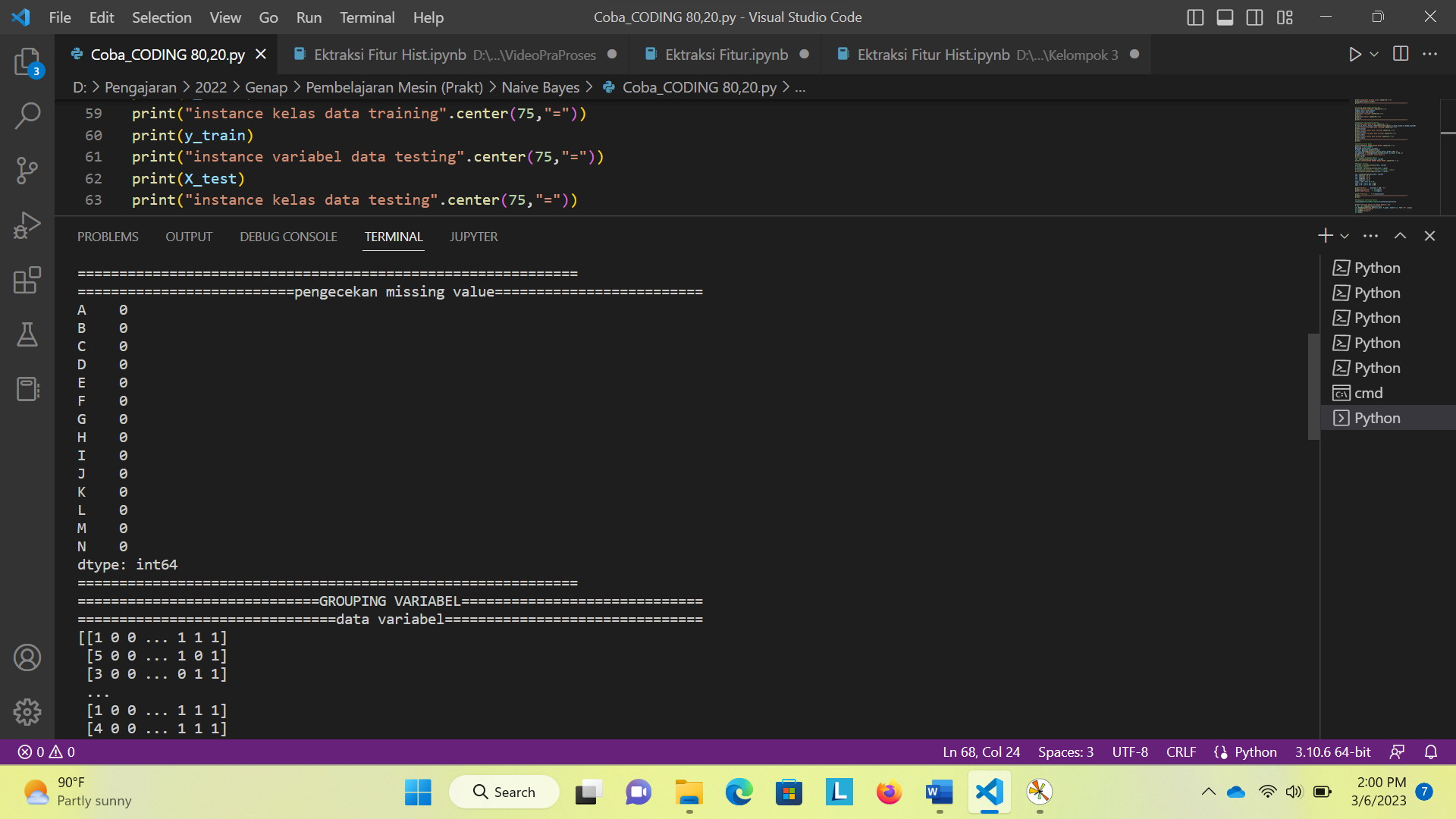
    print("Pasien Positive ")

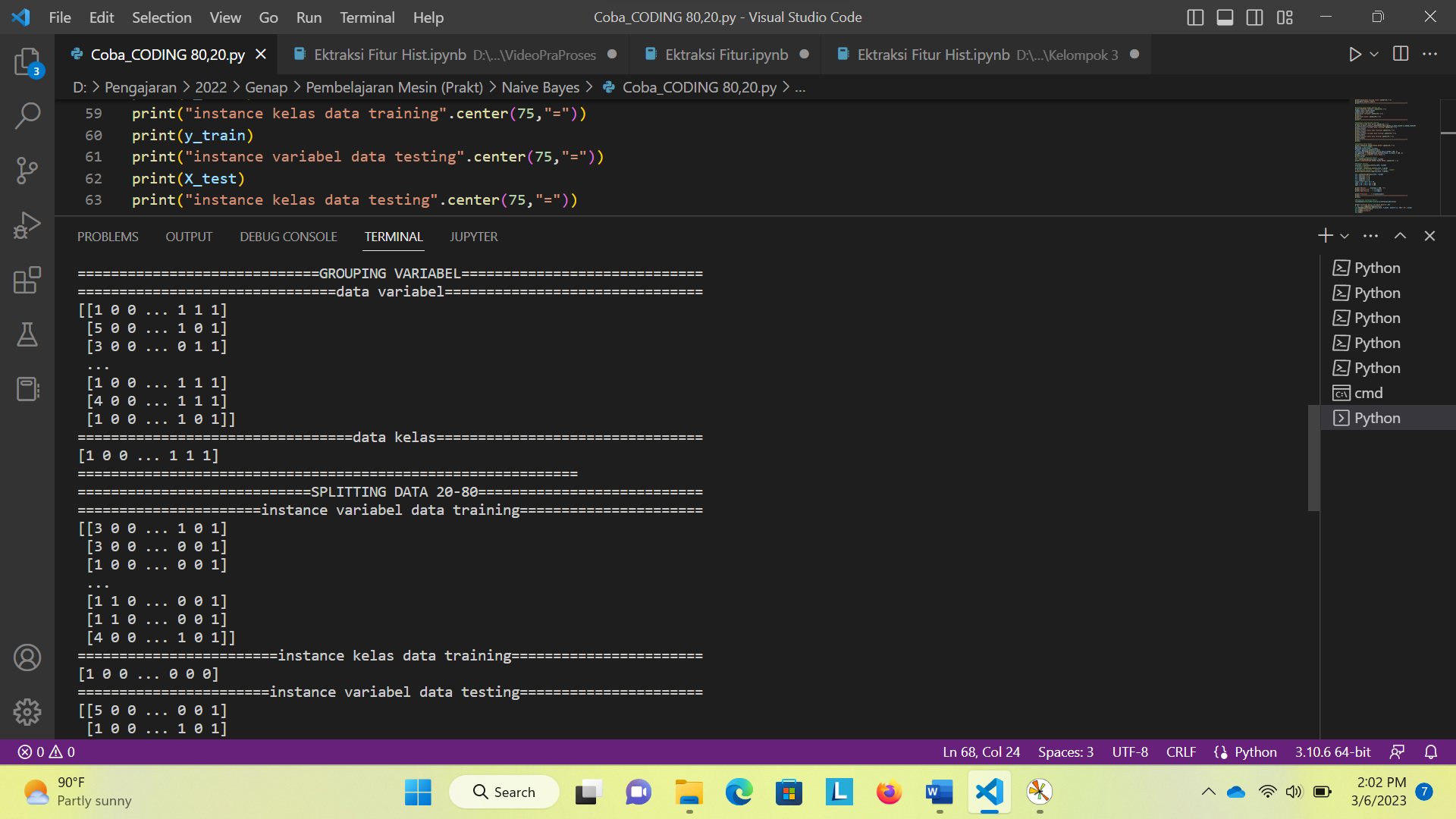
else:

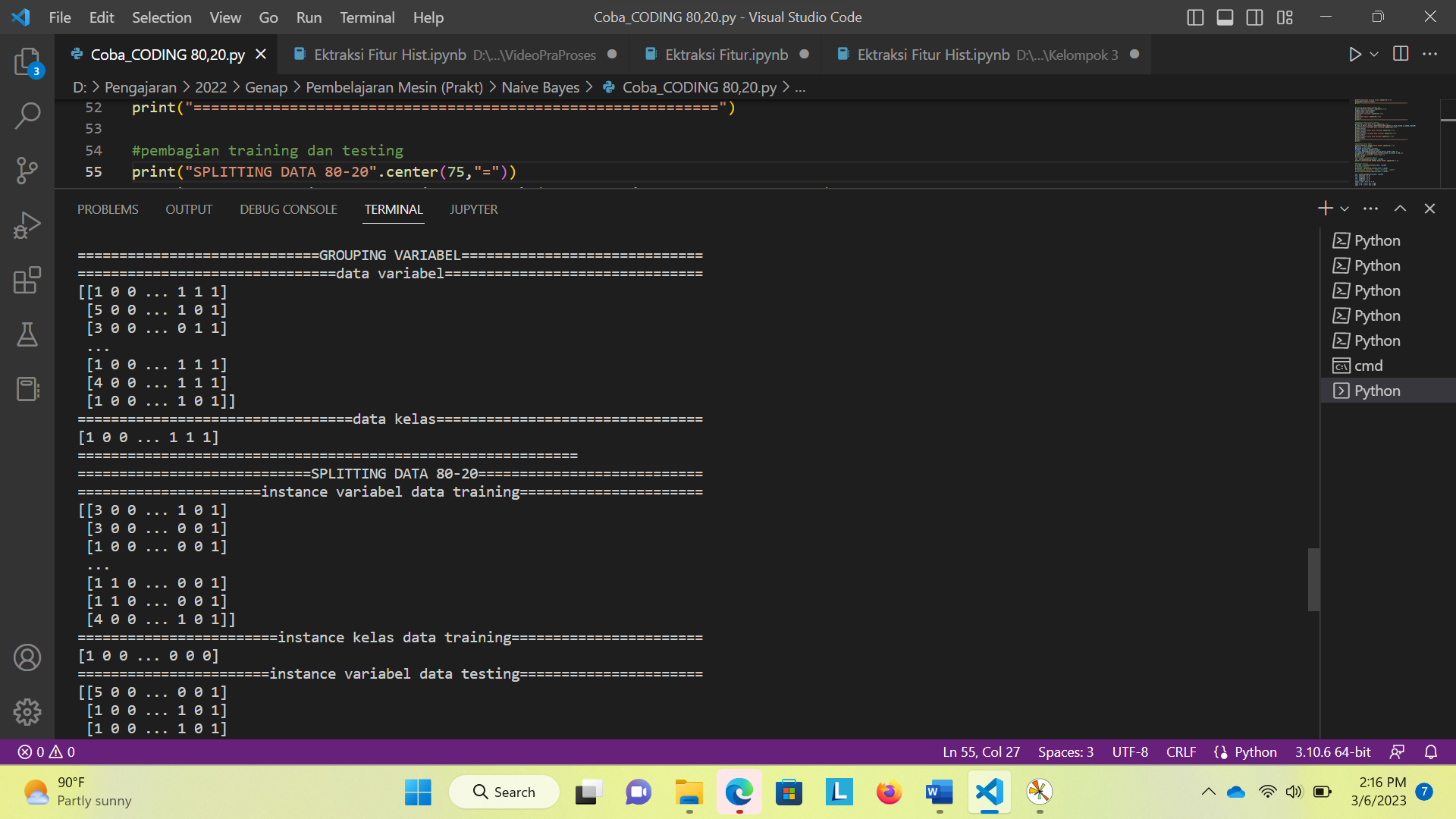
    print("Pasien Negative ")

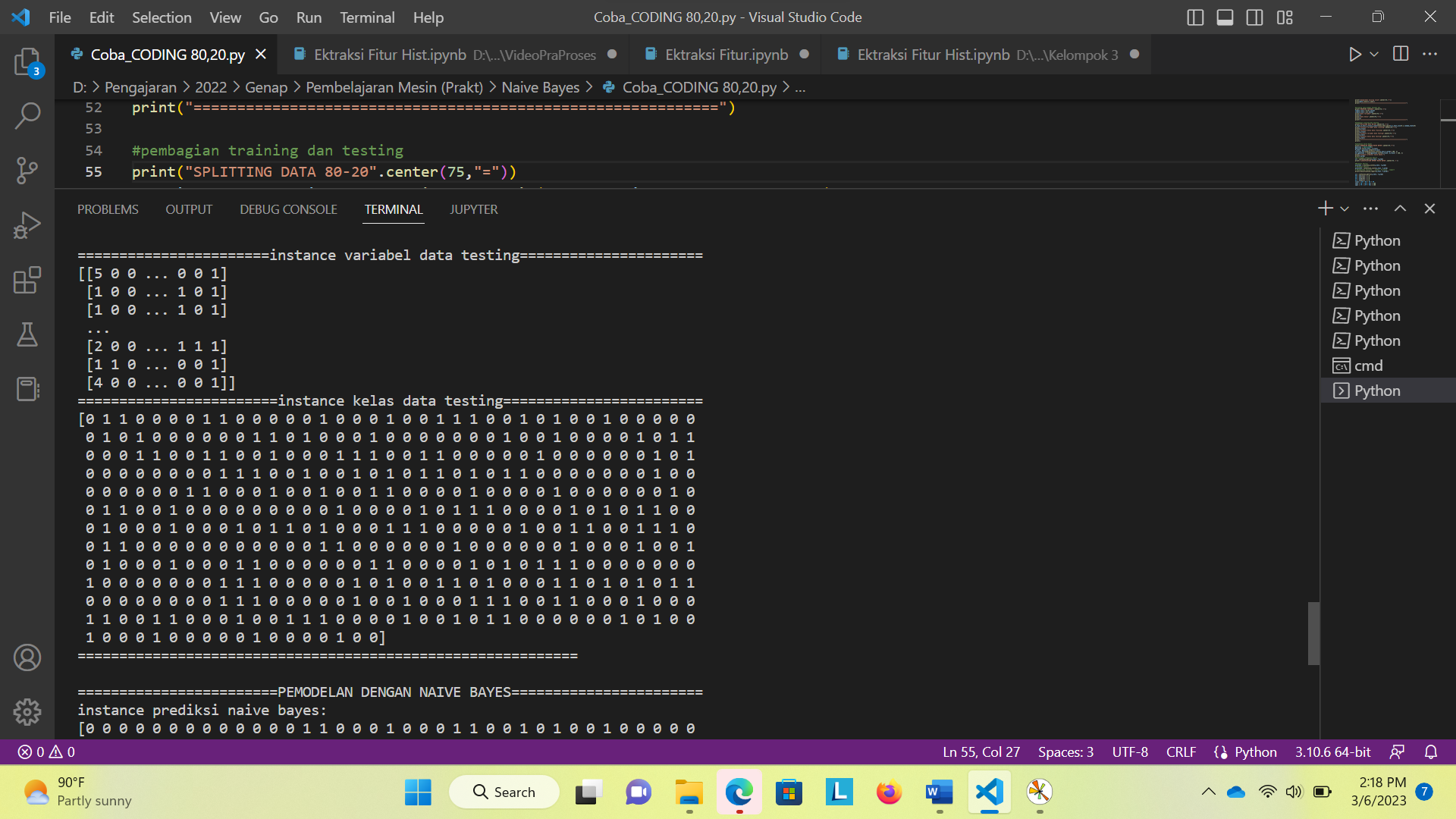
**Hasil**

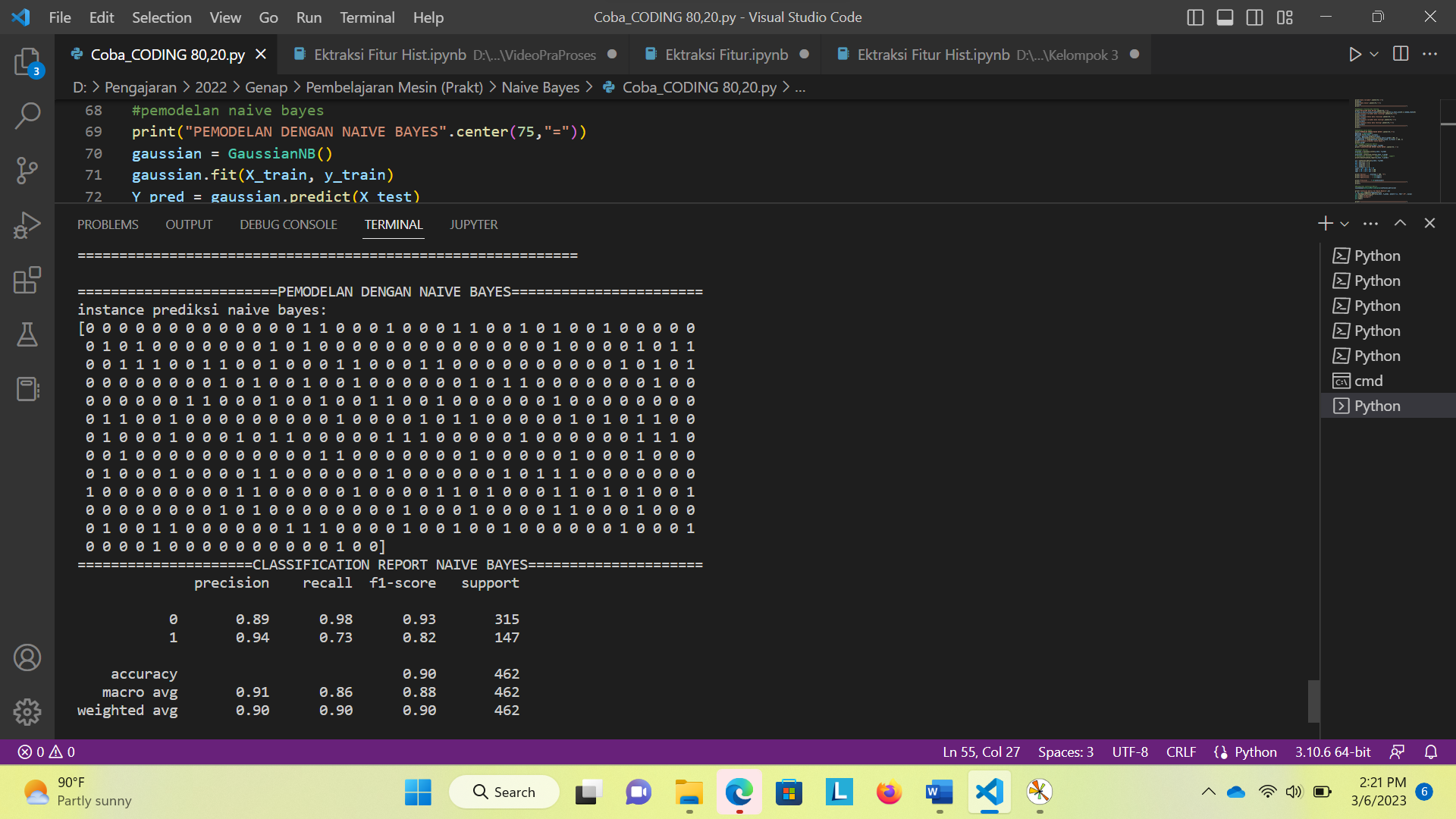
****

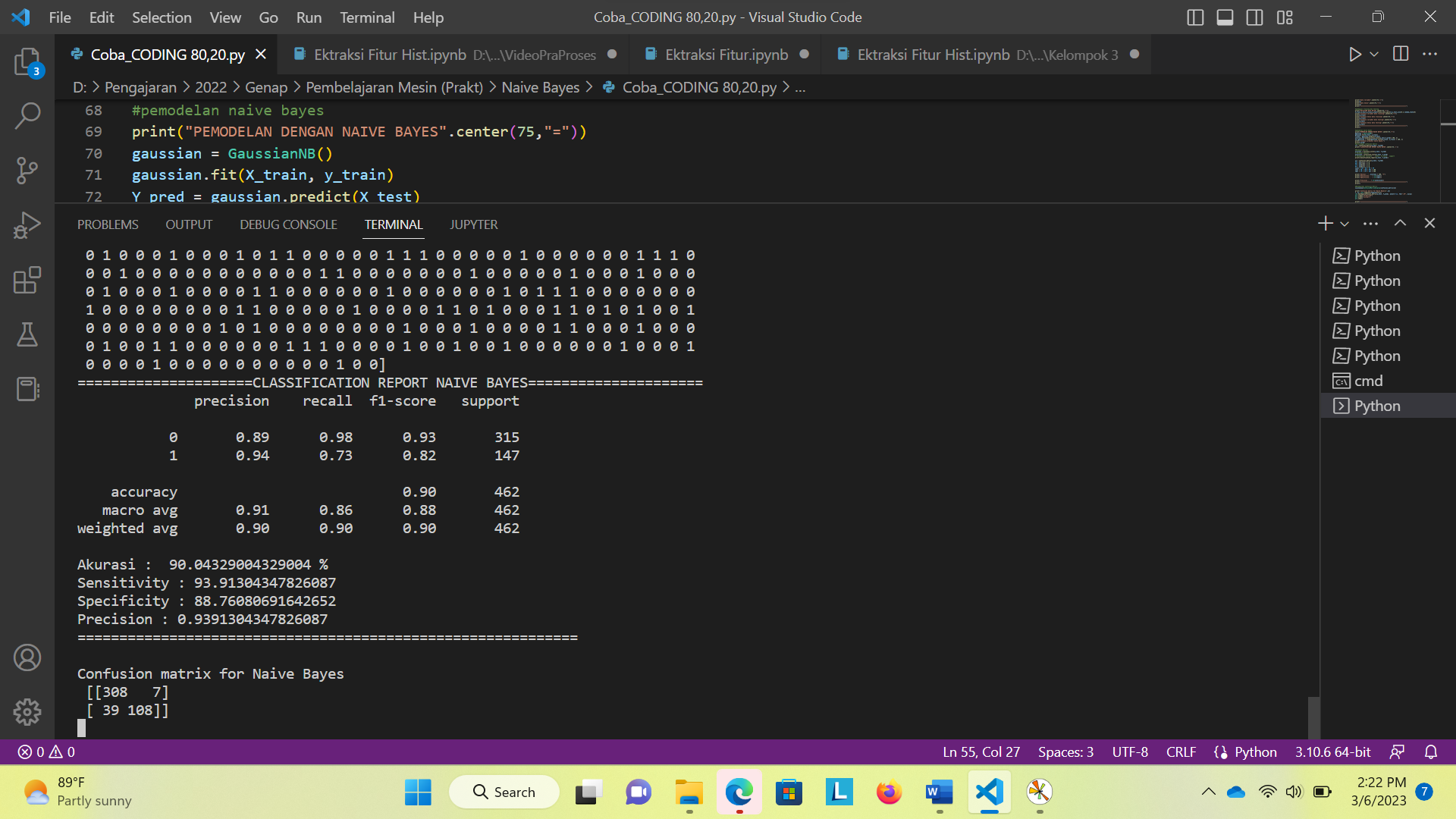
****

****

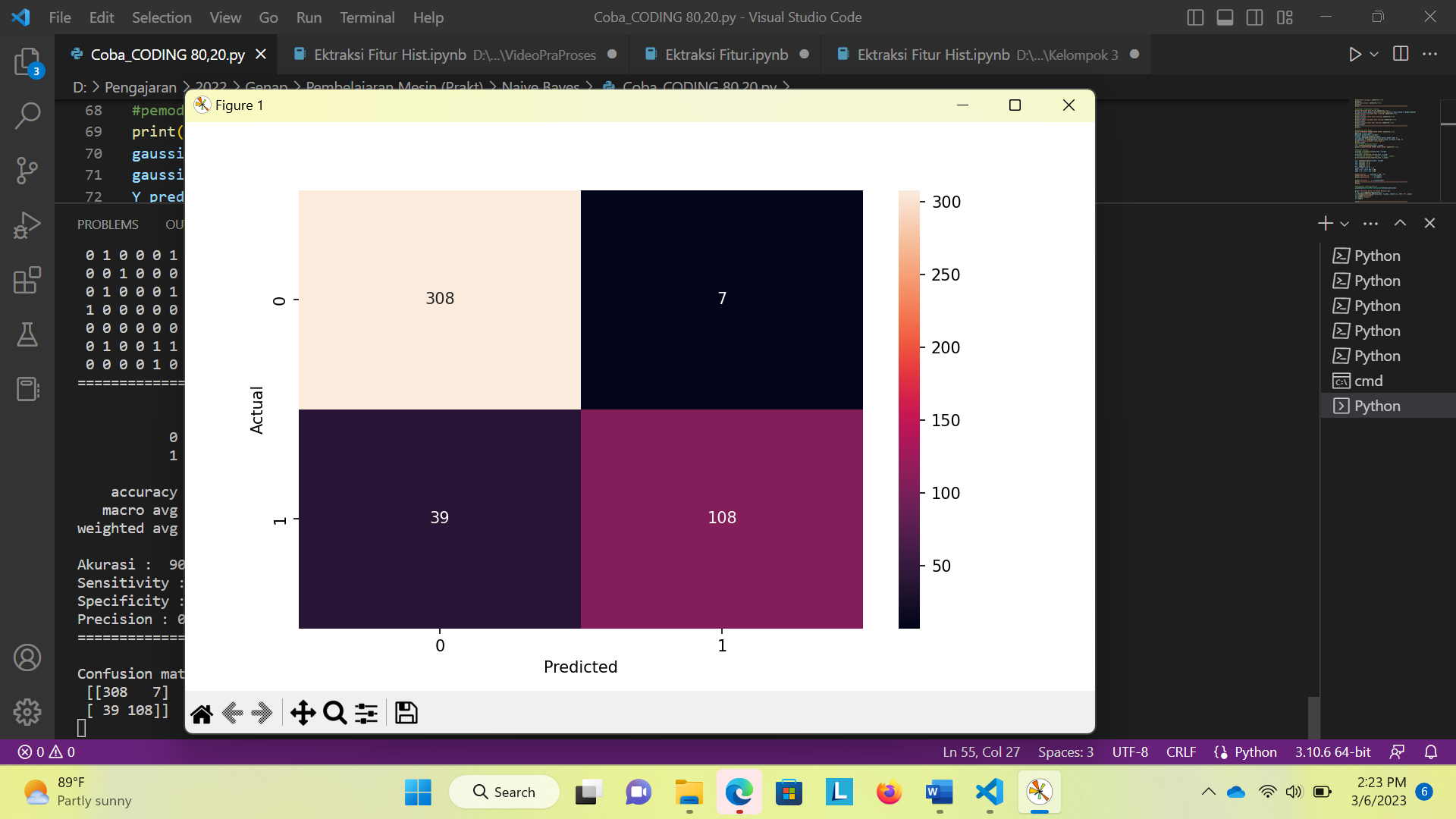
****

****

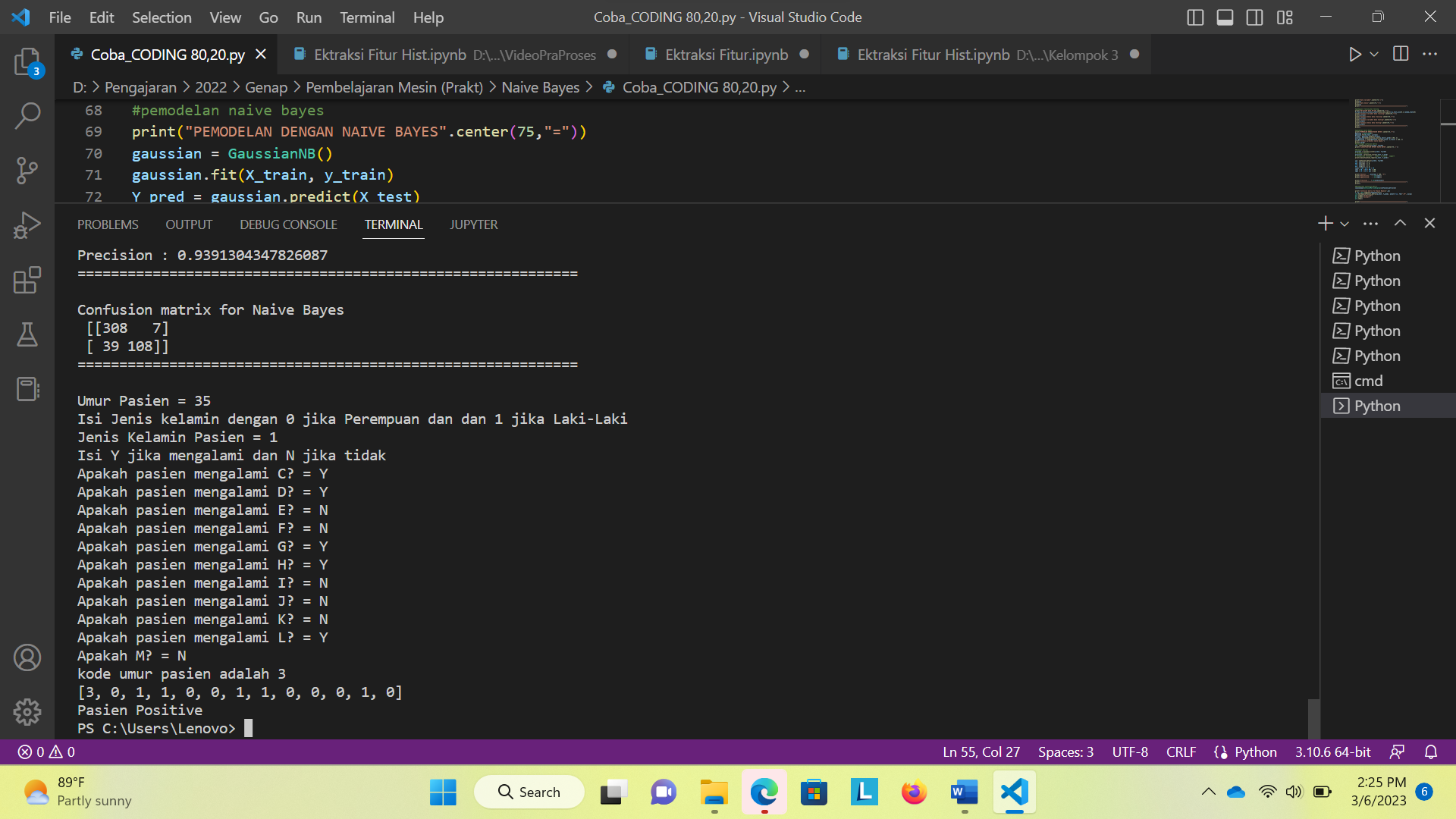
****

****

**Figure :**

****

**Hasil Klasifikasi :**

****

**LATIHAN :**

Cobalah source code di atas

**Tugas :**

1. Carilah data bebas

2. Lakukan klasifikasi dengan naïve bayes

3. Buatlah laporan yang terdiri dari print screen hasil dan penjelasannya

4. Berilah nama “ Naive Bayes\_Kelompok.pdf

6. Tugas terdiri dari Laporan, file phyton, dan data aslinya dan dikumpulkan dengan nama “Tugas Naïve Bayes\_Kelompok.Zip

7. Tugas dikumpulkan paling lambat hari Sabtu / 11 Maret 2023 pukul 23.59 Wib