

# Penerapan Algoritma K-Nearest Neighbor Dalam Klasifikasi Penyakit Jantung

<sup>1</sup>Mohammad Fauzi Akbarollah Sularno\*, <sup>2</sup>Wiyanto Wiyanto, <sup>3</sup>Dodit Ardiatma, <sup>4</sup>Ahmad Turmudi Zy

Fakultas Teknik, Program Studi Informatika, Universitas Pelita Bangsa, Kab. Bekasi, Indonesia

Email: <sup>1,\*</sup>mohammadfauziakbarollahs@gmail.com, <sup>2</sup>wiyanto@pelitabangsa.ac.id, <sup>3</sup>doditardiatma@pelitabangsa.ac.id, <sup>4</sup>turmudi@pelitabangsa.ac.id

Correspondence Author Email: mohammadfauziakbarollahs@gmail.com

Submitted: 12/08/2023; Accepted: 24/08/2023; Published: 25/08/2023

**Abstrak**—Penyakit jantung adalah kondisi ketika bagian jantung yang meliputi pembuluh darah jantung, selaput jantung, katup jantung, dan otot jantung mengalami gangguan. Penyakit jantung bisa disebabkan oleh berbagai hal, seperti sumbatan pada pembuluh darah jantung, peradangan, infeksi, atau kelainan bawaan. Berdasarkan catatan Organisasi Kesehatan Sedunia (WHO), diperkirakan sebanyak 17,9 juta kematian setiap tahunnya disebabkan oleh masalah kardiovaskular. Angka ini setara dengan 32 persen kasus kematian secara global. Dari angka tersebut, sebanyak 85 persen diantaranya disebabkan oleh serangan jantung dan stroke. Tujuan Penelitian ini adalah mengimplementasikan data mining untuk meningkatkan efektivitas pengambilan keputusan dalam pengobatan penyakit jantung. Pada penelitian ini dilakukan proses klasifikasi terhadap sebuah dataset penyakit jantung dengan menggunakan algoritma K-Nearest Neighbor. Hasil dari pengujian mendapatkan nilai accuracy sebesar 92%, precision sebesar 90%, dan nilai recall sebesar 92%..

**Kata Kunci:** K-Nearest Neighbor; Penyakit Jantung; Python

**Abstract**—Heart disease is a condition in which parts of the heart, including the heart's blood vessels, heart membranes, heart valves, and heart muscles, experience disturbances. Heart disease can be caused by various factors, such as blockages in the heart's blood vessels, inflammation, infections, or congenital abnormalities. According to the World Health Organization (WHO), an estimated 17.9 million deaths each year are caused by cardiovascular problems. This number is equivalent to 32 percent of global deaths. Among these figures, 85 percent are attributed to heart attacks and strokes. The aim of this study is to implement data mining to enhance decision-making effectiveness in the treatment of heart disease. This research involves a classification process on a heart disease dataset using the K-Nearest Neighbor algorithm. The test results yielded an accuracy of 92%, a precision of 90%, and a recall value of 92%.

**Keywords:** K-Nearest Neighbor; Heart Disease; Python

## 1. PENDAHULUAN

Penyakit jantung adalah kondisi ketika bagian jantung yang meliputi pembuluh darah jantung, selaput jantung, katup jantung, dan otot jantung mengalami gangguan. Penyakit jantung bisa disebabkan oleh berbagai hal, seperti sumbatan pada pembuluh darah jantung, peradangan, infeksi, atau kelainan bawaan. Pada dinding jantung, terdapat pembuluh darah jantung atau arteri koroner yang mengalirkan darah kaya oksigen ke seluruh bagian jantung. Pembuluh ini memiliki dua cabang, yaitu pembuluh darah koroner kanan dan kiri [1].

Berdasarkan catatan Organisasi Kesehatan Sedunia (WHO), diperkirakan sebanyak 17,9 juta kematian setiap tahunnya disebabkan oleh masalah kardiovaskular. Angka ini setara dengan 32 persen kasus kematian secara global. Dari angka tersebut, sebanyak 85 persen diantaranya disebabkan oleh serangan jantung dan stroke. Hal ini masih harus terus menjadi perhatian masyarakat. Betapa tidak, ancaman penyakit jantung begitu nyata [2].

Laporan riset Kesehatan Dasar 2018 menunjukkan, rata-rata prevalensi penyakit ini di Indonesia sebesar 1,5% pada tahun tersebut. Ada 11 provinsi yang memiliki prevalensi penyakit jantung di atas rata-rata nasional tersebut. Kalimantan Utara memiliki prevalensi penyakit jantung tertinggi di Indonesia sebesar 2,2%. DI Yogyakarta dan Gorontalo dengan prevalensi penyakit jantung masing-masing sebesar 2%. Kemudian, Kalimantan Timur, DKI Jakarta, dan Sulawesi Tengah masing-masing memiliki prevalensi penyakit jantung sebesar 1,9%. Kemudian, prevalensi penyakit jantung di Sulawesi Utara sebesar 1,8%. Sementara itu, sebesar masing-masing 1,6% prevalensi penyakit jantung berasal dari Aceh, Sumatera Barat, Jawa Barat, dan Jawa Tengah [3].

Penyakit jantung memiliki berbagai jenis, seperti penyakit jantung koroner, gagal jantung, aritmia, dan lain-lain. Setiap jenis penyakit jantung memiliki ciri dan gejala yang berbeda. Oleh karena itu, penting untuk mengembangkan metode yang efektif untuk mengklasifikasikan jenis penyakit jantung berdasarkan gejala pasien, tes diagnostik, riwayat medis, dan faktor risiko. Kemajuan teknologi sistem informasi telah memecahkan banyak masalah di berbagai bidang, terutama bidang medis, salah satunya adalah penerapan data mining. Data mining adalah proses penggalian pengetahuan atau pola yang berguna dan bermakna dari kumpulan data yang besar dan kompleks. Tujuan dari data mining adalah untuk mengidentifikasi hubungan, pola, atau tren tersembunyi dalam data yang dapat memberikan wawasan dan wawasan yang berharga untuk pengambilan keputusan.

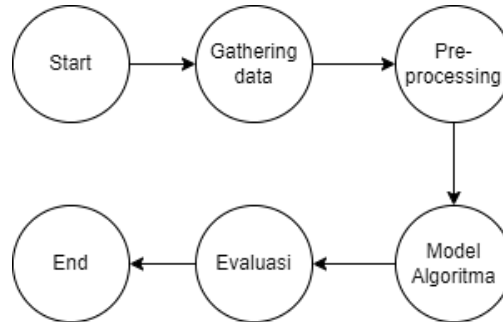
Algoritma KNN merupakan algoritma klasifikasi yang sederhana namun efektif dalam pengolahan data. Algoritma ini dapat digunakan untuk mengklasifikasikan data berdasarkan pola dan kemiripan dengan data

training yang ada. Sebagai bagian dari penerapan KNN pada klasifikasi penyakit jantung, algoritma ini dapat digunakan untuk mengklasifikasikan pasien berdasarkan gejala, riwayat medis, dan faktor risiko yang terkait dengan penyakit jantung.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Dalam penelitian ini akan dilakukan klasifikasi menggunakan algoritma *K-Nearest Neighbor* terhadap dataset penyakit jantung. Penelitian ini akan melalui beberapa proses, tahapan penelitian bisa dilihat pada gambar 1. dibawah ini.



Gambar 1. Tahapan Penelitian Kalsifikasi dengan Algoritma KNN.

Berikut penjelasan untuk uraian diagram diatas:

a. *Gathering data*

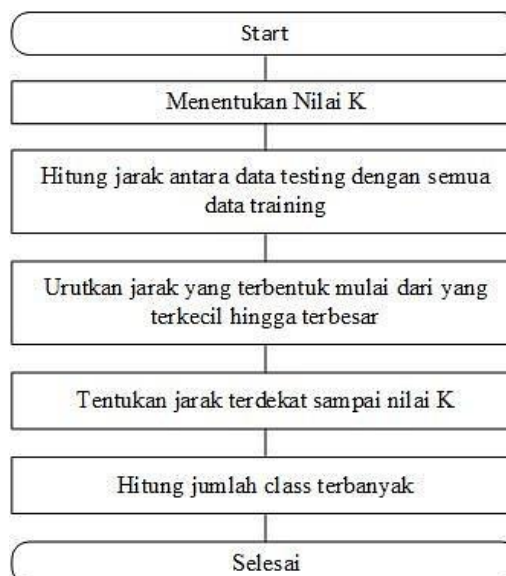
Tahapan awal dalam penelitian ini ialah dimulai dengan memilih dataset yang akan digunakan. Pada penelitian ini dataset yang digunakan adalah merupakan dataset pasien penyakit jantung dengan jumlah record data sebanyak 1024 dengan jumlah atribut sebanyak 14 kolom termasuk class. Dengan data ini diharapkan dapat memperoleh hasil yang akurat dan terpercaya.

b. *Pre-processing*

Sebelum dilakukan proses pengklasifikasian data, dilakukan metode pemrosesan data awal terlebih dahulu sesuai dengan proses metode *Knowledge Discovery in Database* (KDD) yaitu preprocessing data. Dataset diabetes yang didapatkan tidak semua data atau record dan atribut digunakan. Data atau record dan atribut harus melalui beberapa tahap pengolahan awal data seperti pembersihan data (cleaning), filtering, transformasi data, dan lain-lain sehingga mendapatkan data yang berkualitas.

c. Model Algoritma

Pada penelitian ini algoritma yang digunakan adalah *K-Nearest Neighbors*. Pada tahapan ini penggunaan K-NN digunakan untuk mendapatkan sejumlah nilai titik terdekat dengan nilai titik baru dengan nilai  $K = 3, 5, 7, 9$  dan seterusnya. Berikut untuk alur perhitungan algoritma *K-Nearest Neighbor*.



Gambar 2. Alur perhitungan algoritma K-NN

d. Evaluasi

Evaluasi dilakukan dengan cara menganalisa hasil dari algoritma yang digunakan untuk memastikan bahwa hasil perhitungan dan pengujian sesuai dengan tujuan penelitian. Validasi dilakukan untuk mengukur hasil klasifikasi guna mengetahui tingkat accuracy, precision, dan recall.

1. Accuracy

Nilai akurasi didapatkan dari jumlah data bernilai positif yang diprediksi positif dan data bernilai negatif yang diprediksi negatif dibagi dengan jumlah seluruh data di dalam dataset.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{1}$$

2. Precision

Precision adalah peluang kasus yang diprediksi positif yang pada kenyataannya termasuk kasus kategori positif.

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

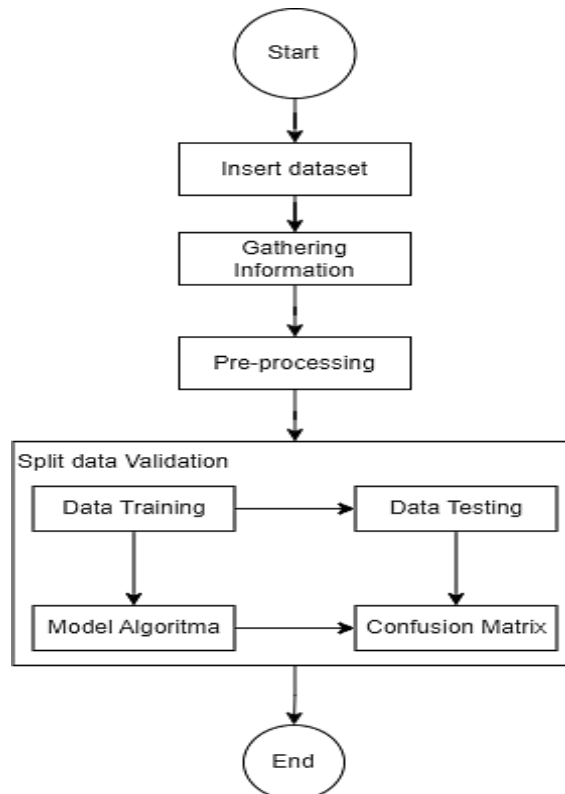
3. Recall

Recall adalah peluang kasus dengan kategori positif yang dengan tepat diprediksi positif.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

2.2 Pengujian

Software yang akan digunakan untuk pengujian pada penelitian ini yaitu *Google Colaboratory* dengan menggunakan bahasa pemrograman python. Untuk model pengujian bisa dilihat pada gambar berikut.



Gambar 3. Logo Forum Kerjasama Pendidikan Tinggi (FKPT)

Adapun penjelasan mengenai langkah-langkah dalam pengujian sebagai berikut:

a. Insert Dataset

Langkah pertama ialah memasukkan dataset yang akan diujikan.

b. Gathering Information

Pada tahapan ini mengumpulkan sejumlah informasi pada dataset mulai dari jumlah baris, jumlah atribut, korelasi antar atribut, jumlah variabel NULL/NaN, dan lain-lain.

c. Data Preprocessing

Langkah selanjutnya adalah melakukan preprocessing seperti *data cleansing*, *selection feature*, *transformation data*, dan *feature engineering*. Ini berguna untuk membersihkan nilai value yang kosong dan bersifat redundansi, dan juga pemilihan atribut karena tidak semua atribut akan dipakai.

d. Split data Validation

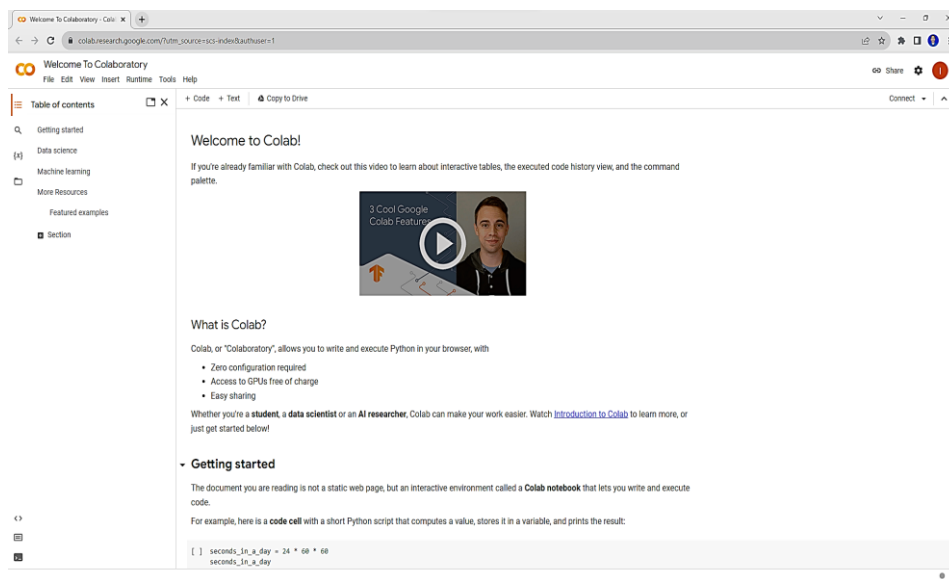
Pada tahap ini data akan membagi jumlah keseluruhan dataset menjadi 2 yaitu data training dan data testing dengan nilai yaitu 80:20, dengan data training 80% dan data testing 20%. Setelah membagi dataset langkah selanjutnya adalah memasukkan data training kedalam model algoritma. Setelah melakukan penerapan algoritma, kemudian hasil tersebut akan diujikan dengan data testing sehingga akan mendapatkan bentuk confusion matrix sebagai nilai akhir.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil Pengujian

##### 3.1.1 Pengujian Algoritma K-NN menggunakan Python

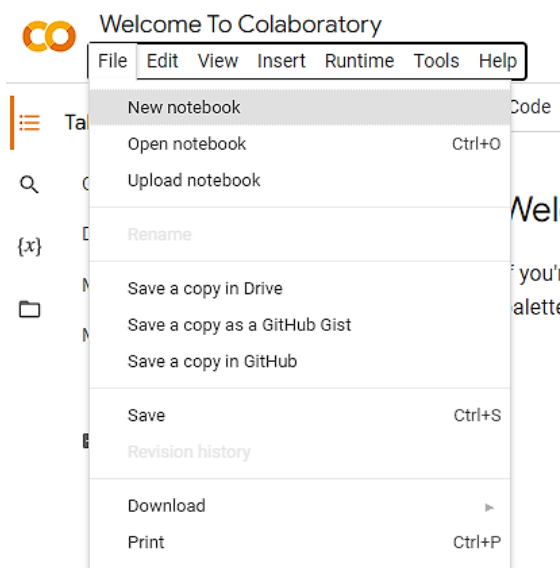
Pada tahap pengujian ini, penulis menggunakan platform aplikasi dari google yaitu Google Collaboratory untuk membuat sebuah program dengan menggunakan bahasa pemrograman python. Berikut tampilan interface dari Google Collaboratory yang bisa dilihat pada gambar 4.



**Gambar 4.** Halaman Utama *Google Collaboratory*

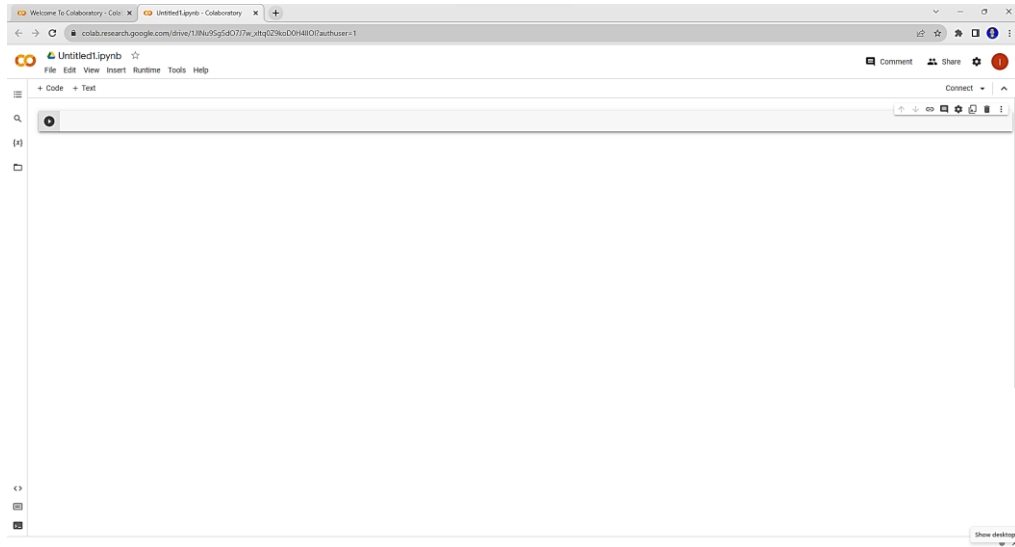
##### 3.1.2 Membuat file Baru

Langkah pertama yang akan dilakukan adalah membuat file baru untuk membuat sebuah program. Untuk membuat file baru bisa dilihat pada gambar 5.



**Gambar 5.** *New File Notebook*

Berikut tampilan dari new notebook yang telah dibuat untuk membuat program python bisa dilihat pada gambar 6.



**Gambar 6.** *Interface New Notebook*

### 3.1.3 Upload Dataset

Sebelum melakukan pengetikan program, terlebih dahulu harus upload dataset yang akan digunakan ke *Google Drive*. Pastikan akun yang sama dipakai untuk upload dataset ke *Google Drive* dan yang digunakan untuk *Google Collaboratory*. Pada penelitian ini penulis menggunakan dataset dengan format CSV (*Comma Separated Value*). Setelah file dataset sudah di upload hubungkan google drive dengan google collab menggunakan sintaks berikut:

```
#menghubungkan google colab dengan google  
from google.colab import drive  
drive.mount('/content/drive')
```

Mounted at /content/drive

**Gambar 7.** *Connect to Google Drive*

### 3.1.4 Memanggil Dataset

Sebelum memanggil dataset yang akan digunakan jalankan library pendukung untuk memanipulasi data dan untuk visualisasi data dengan menggunakan sintaks berikut:

```
#import library pendukung  
  
import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt
```

**Gambar 8.** *Import Library Pendukung*

Setelah import library sudah dilakukan langkah selanjutnya adalah memanggil dataset yang akan digunakan. Dengan menggunakan sintaks berikut:

```
dataset = '/content/drive/My Drive/Datasets/heart_disease_dataset.csv'
df = pd.read_csv(dataset, sep=',', header=0, quotechar='/')
df
```

|      | age | sex | cp  | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca  | thal | target |
|------|-----|-----|-----|----------|------|-----|---------|---------|-------|---------|-------|-----|------|--------|
| 0    | 52  | 1   | 0   | 125      | 212  | 0   | 1       | 168     | 0     | 1.0     | 2     | 2   | 3    | 0      |
| 1    | 53  | 1   | 0   | 140      | 203  | 1   | 0       | 155     | 1     | 3.1     | 0     | 0   | 3    | 0      |
| 2    | 70  | 1   | 0   | 145      | 174  | 0   | 1       | 125     | 1     | 2.6     | 0     | 0   | 3    | 0      |
| 3    | 61  | 1   | 0   | 148      | 203  | 0   | 1       | 161     | 0     | 0.0     | 2     | 1   | 3    | 0      |
| 4    | 62  | 0   | 0   | 138      | 294  | 1   | 1       | 106     | 0     | 1.9     | 1     | 3   | 2    | 0      |
| ...  | ... | ... | ... | ...      | ...  | ... | ...     | ...     | ...   | ...     | ...   | ... | ...  | ...    |
| 1020 | 59  | 1   | 1   | 140      | 221  | 0   | 1       | 164     | 1     | 0.0     | 2     | 0   | 2    | 1      |
| 1021 | 60  | 1   | 0   | 125      | 258  | 0   | 0       | 141     | 1     | 2.8     | 1     | 1   | 3    | 0      |
| 1022 | 47  | 1   | 0   | 110      | 275  | 0   | 0       | 118     | 1     | 1.0     | 1     | 1   | 2    | 0      |
| 1023 | 50  | 0   | 0   | 110      | 254  | 0   | 0       | 159     | 0     | 0.0     | 2     | 0   | 2    | 1      |
| 1024 | 54  | 1   | 0   | 120      | 188  | 0   | 1       | 113     | 0     | 1.4     | 1     | 1   | 3    | 0      |

1025 rows x 14 columns

Gambar 9. Memanggil Dataset

### 3.1.5 Gathering Information

Langkah selanjutnya adalah mendapatkan informasi yang ada pada dataset yang akan digunakan dan akan dijelaskan di masing-masing gambar.

```
df.shape
```

(1025, 14)

Gambar 10. Menampilkan Jumlah Baris Dan Kolom Dataset

```
df.describe()
```

|       | age         | sex         | cp          | trestbps    | chol        | fbs         | restecg     | thalach     | exang       | oldpeak     | slope       | ca          | thal        | target      |
|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| count | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 | 1025.000000 |
| mean  | 54.434146   | 0.695610    | 0.942439    | 131.611707  | 246.000000  | 0.149268    | 0.529756    | 149.114146  | 0.336585    | 1.071512    | 1.385366    | 0.754146    | 2.323902    | 0.513171    |
| std   | 9.072290    | 0.460373    | 1.029641    | 17.516718   | 51.592511   | 0.356527    | 0.527878    | 23.005724   | 0.472772    | 1.175053    | 0.617755    | 1.030798    | 0.620660    | 0.500070    |
| min   | 29.000000   | 0.000000    | 0.000000    | 94.000000   | 126.000000  | 0.000000    | 0.000000    | 71.000000   | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    | 0.000000    |
| 25%   | 48.000000   | 0.000000    | 0.000000    | 120.000000  | 211.000000  | 0.000000    | 0.000000    | 132.000000  | 0.000000    | 0.000000    | 1.000000    | 0.000000    | 2.000000    | 0.000000    |
| 50%   | 56.000000   | 1.000000    | 1.000000    | 130.000000  | 240.000000  | 0.000000    | 1.000000    | 152.000000  | 0.000000    | 0.800000    | 1.000000    | 0.000000    | 2.000000    | 1.000000    |
| 75%   | 61.000000   | 1.000000    | 2.000000    | 140.000000  | 275.000000  | 0.000000    | 1.000000    | 166.000000  | 1.000000    | 1.800000    | 2.000000    | 1.000000    | 3.000000    | 1.000000    |
| max   | 77.000000   | 1.000000    | 3.000000    | 200.000000  | 564.000000  | 1.000000    | 2.000000    | 202.000000  | 1.000000    | 6.200000    | 2.000000    | 4.000000    | 3.000000    | 1.000000    |

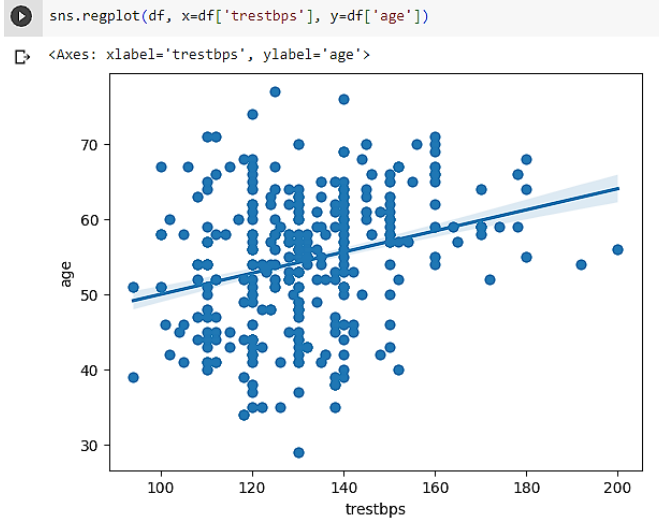
Gambar 11. Menampilkan Statistik Deskripsi Dataset

```
df.corr()
```

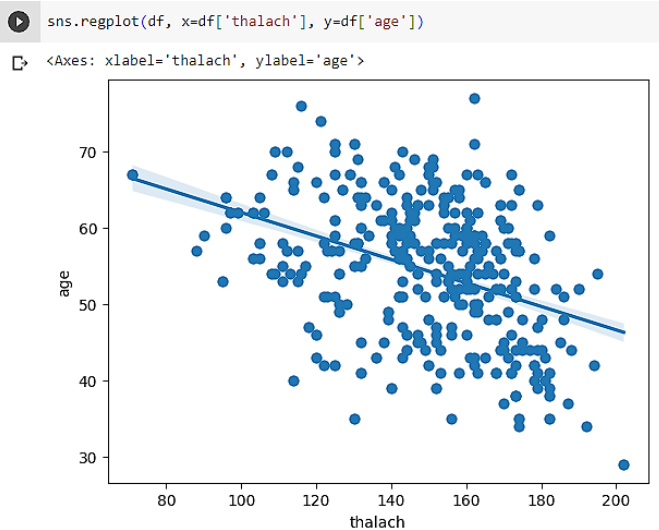
|          | age       | sex       | cp        | trestbps  | chol      | fbs       | restecg   | thalach   | exang     | oldpeak   | slope     | ca        | thal      | target    |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| age      | 1.000000  | -0.103240 | -0.071966 | 0.271121  | 0.219823  | 0.121243  | -0.132696 | -0.390227 | 0.088163  | 0.208137  | -0.169105 | 0.271551  | 0.072297  | -0.229324 |
| sex      | -0.103240 | 1.000000  | -0.041119 | -0.078974 | -0.198258 | 0.027200  | -0.055117 | -0.049365 | 0.139157  | 0.084687  | -0.026666 | 0.111729  | 0.198424  | -0.279501 |
| cp       | -0.071966 | -0.041119 | 1.000000  | 0.038177  | -0.081641 | 0.079294  | 0.043581  | 0.306839  | -0.401513 | -0.174733 | 0.131633  | -0.176206 | -0.163341 | 0.434854  |
| trestbps | 0.271121  | -0.078974 | 0.038177  | 1.000000  | 0.127977  | 0.181767  | -0.123794 | -0.039264 | 0.061197  | 0.187434  | -0.120445 | 0.104554  | 0.059276  | -0.138772 |
| chol     | 0.219823  | -0.198258 | -0.081641 | 0.127977  | 1.000000  | 0.026917  | -0.147410 | -0.021772 | 0.067382  | 0.064880  | -0.014248 | 0.074259  | 0.100244  | -0.099966 |
| fbs      | 0.121243  | 0.027200  | 0.079294  | 0.181767  | 0.026917  | 1.000000  | -0.104051 | -0.008866 | 0.049261  | 0.010859  | -0.061902 | 0.137156  | -0.042177 | -0.041164 |
| restecg  | -0.132696 | -0.055117 | 0.043581  | -0.123794 | -0.147410 | -0.104051 | 1.000000  | 0.048411  | -0.065606 | -0.050114 | 0.086086  | -0.078072 | -0.020504 | 0.134468  |
| thalach  | -0.390227 | -0.049365 | 0.306839  | -0.039264 | -0.021772 | -0.008866 | 0.048411  | 1.000000  | -0.380281 | -0.349796 | 0.395308  | -0.207888 | -0.098068 | 0.422895  |
| exang    | 0.088163  | 0.139157  | -0.401513 | 0.061197  | 0.067382  | 0.049261  | -0.065606 | -0.380281 | 1.000000  | 0.310844  | -0.267335 | 0.107849  | 0.197201  | -0.438029 |
| oldpeak  | 0.208137  | 0.084687  | -0.174733 | 0.187434  | 0.064880  | 0.010859  | -0.050114 | -0.349796 | 0.310844  | 1.000000  | -0.575189 | 0.221816  | 0.202672  | -0.438441 |
| slope    | -0.169105 | -0.026666 | 0.131633  | -0.120445 | -0.014248 | -0.061902 | 0.086086  | 0.395308  | -0.267335 | -0.575189 | 1.000000  | -0.073440 | -0.094090 | 0.345512  |
| ca       | 0.271551  | 0.111729  | -0.176206 | 0.104554  | 0.074259  | 0.137156  | -0.078072 | -0.207888 | 0.107849  | 0.221816  | -0.073440 | 1.000000  | 0.149014  | -0.382085 |
| thal     | 0.072297  | 0.198424  | -0.163341 | 0.059276  | 0.100244  | -0.042177 | -0.020504 | -0.098068 | 0.197201  | 0.202672  | -0.094090 | 0.149014  | 1.000000  | -0.337838 |
| target   | -0.229324 | -0.279501 | 0.434854  | -0.138772 | -0.099966 | -0.041164 | 0.134468  | 0.422895  | -0.438029 | -0.438441 | 0.345512  | -0.382085 | -0.337838 | 1.000000  |

Gambar 12. Menampilkan Korelasi Antar Atribut

Informasi yang didapatkan dari korelasi antar atribut adalah melihat persebaran data yang terjadi di antara 2 atribut yang bersangkutan, contohnya seperti:



Gambar 13. Persebaran Data Atribut Age Dan Tresbps



Gambar 14. Persebaran Data Atribut Age Dan Thalach

```
df.dtypes
```

|          |         |
|----------|---------|
| age      | int64   |
| sex      | int64   |
| cp       | int64   |
| trestbps | int64   |
| chol     | int64   |
| fbs      | int64   |
| restecg  | int64   |
| thalach  | int64   |
| exang    | int64   |
| oldpeak  | float64 |
| slope    | int64   |
| ca       | int64   |
| thal     | int64   |
| target   | int64   |
| dtype:   | object  |

Gambar 15. Menampilkan Type Masing-Masing Atribut

```
df.isna().sum()
age      0
sex      0
cp       0
trestbps 0
chol     0
fbs      0
restecg  0
thalach  0
exang    0
oldpeak  0
slope    0
ca       0
thal     0
target   0
dtype: int64
```

Gambar 16. Menampilkan Atribut Yang Memiliki Nilai NULL/Nan

### 3.1.6 Menentukan nilai Atribut dan Class

Langkah selanjutnya dalam penelitian ini yaitu menentukan nilai atribut dan class pada dataset. Pada tahapan ini semua nilai atribut dan class akan ditampung kedalam bentuk variable x dan y, dimana variable x akan berisi nilai atribut dan variable y akan berisi class. Dengan menggunakan sintaks berikut:

```
x = df.iloc[:, :-1].values #atribut
y = df.iloc[:, -1].values #class
```

Gambar 17. Menentukan Nilai Atribut Dan Class

### 3.1.7 Split data Validation

Langkah selanjutnya adalah membagi dataset menjadi 2 yaitu data training dan data testing. Data dibagi 2 dengan bobot nilai sebesar 80:20 dimana 80% data akan digunakan sebagai data training dan 20% data akan digunakan sebagai data testing. Dengan menggunakan sintaks berikut:

```
from sklearn.model_selection import train_test_split
X_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
                                                    random_state=3)

print('Jumlah data training sebanyak:', len(X_train))
print('Jumlah data test sebanyak:', len(x_test))
```

```
Jumlah data training sebanyak: 820
Jumlah data test sebanyak: 205
```

Gambar 18. Split data

Bisa dilihat pada gambar 4.13 dimana jumlah data training sebanyak 820 data dan data testing sebanyak 205 dengan jumlah keseluruhan total data pada dataset berjumlah 1025 data.

### 3.2 Penerapan Algoritma *K-Nearest Neighbor*

Setelah dataset sudah di bagi menjadi 2, langkah selanjutnya adalah menerapkan data training pada algoritma *K-Nearest Neighbor*. Dengan menggunakan sintaks berikut:

```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors = 3, metric = 'euclidean')
neigh.fit(X_train, y_train)
```

KNeighborsClassifier  
KNeighborsClassifier(metric='euclidean', n\_neighbors=3)

Gambar 19. Penerapan Algoritma

Pada gambar 4.14 perlu terlebih dahulu untuk meng-import library KNeighborClassifier untuk melakukan penerapan algoritma. Setelah meng-import library KNN, langkah selanjutnya adalah memanggil fungsi *KNeighborsClassifier* dengan mendefinisikan parameter nilai K=3 dan metric yang digunakan adalah *Euclidean distance*. Setelah itu langsung terapkan algoritma KNN dengan data training.

### 3.3 Pengujian

Langkah selanjutnya adalah melakukan pengujian terhadap data testing dengan data training yang sudah diterapkan pada algoritma. Dengan menggunakan sintaks berikut:

```
y_pred = neigh.predict(x_test)

#melihat hasil prediksi
for pred, value in zip(y_pred[:5], x_test[:5]):
    print(f'value : {value} - PREDICT : {pred} \n ')

value : [ 58.   1.   2. 105. 240.   0.   0. 154.   1.   0.6   1.   0.
 3. ] - PREDICT : 1

value : [ 53.   1.   0. 123. 282.   0.   1.  95.   1.   2.   1.   2.   3. ] - PREDICT : 0

value : [ 53.   0.   0. 130. 264.   0.   0. 143.   0.   0.4   1.   0.
 2. ] - PREDICT : 1

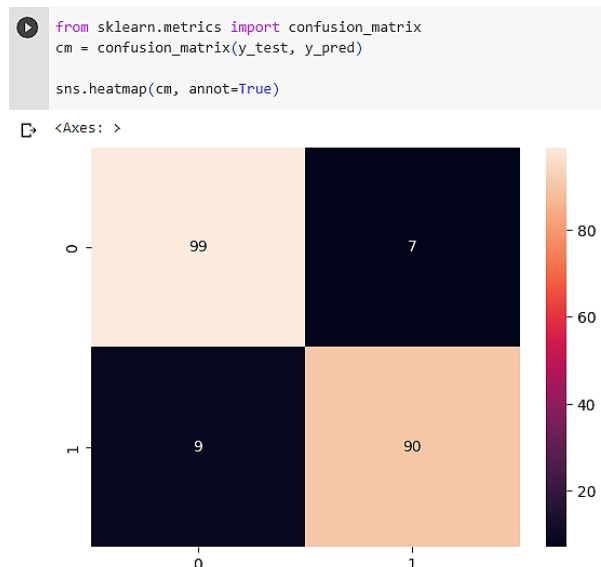
value : [5.80e+01 1.00e+00 0.00e+00 1.00e+02 2.34e+02 0.00e+00 1.00e+00 1.56e+02
 0.00e+00 1.00e-01 2.00e+00 1.00e+00 3.00e+00] - PREDICT : 0

value : [ 56.   0.   0. 200. 288.   1.   0. 133.   1.   4.   0.   2.   3. ] - PREDICT : 0
```

Gambar 20. Hasil Prediksi 5 Baris Pertama

#### 3.3.1 Confusion Matrix

Setelah melakukan pengujian. Langkah selanjutnya adalah melakukan evaluasi terhadap hasil kinerja algoritma *K-Nearest Neighbor* dengan menggunakan confusion matrix. Pada tahapan ini akan dilakukan pembuatan confusion matrix dengan menggunakan library pendukung dari ScikitLearn yaitu *confusion\_matrix* dan memvisualisasikannya menggunakan *seaborn*. Dengan menggunakan sintaks berikut:



Gambar 21. Confusion Matrix

### 3.3.2 Validasi dan Evaluasi

Langkah terakhir dalam penelitian ini adalah melakukan validasi dan evaluasi terhadap hasil pengujian. Pada tahapan ini akan didapatkan hasil akhir dari kemampuan algoritma dalam klasifikasi dataset pasien penyakit jantung. Hasil yang akan didapatkan adalah berupa *accuracy score*, *recall*, dan *precision*. Proses untuk mendapatkan hasil akhir menggunakan sintaks berikut:

```

from sklearn.metrics import accuracy_score, precision_score, recall_score
print('Accuracy score: ', accuracy_score(y_pred, y_test))
print('Precision score: ', precision_score(y_pred, y_test))
print('Recall score: ', recall_score(y_pred, y_test))

```

Accuracy score: 0.9219512195121952  
 Precision score: 0.9090909090909091  
 Recall score: 0.9278350515463918

Gambar 22. Hasil Akurasi, Presisi, Dan Recall

```

from sklearn.metrics import classification_report
print(classification_report(y_pred, y_test))

```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.93      | 0.92   | 0.93     | 108     |
| 1            | 0.91      | 0.93   | 0.92     | 97      |
| accuracy     |           |        | 0.92     | 205     |
| macro avg    | 0.92      | 0.92   | 0.92     | 205     |
| weighted avg | 0.92      | 0.92   | 0.92     | 205     |

Gambar 22. Classification Report

### 3.4 Pembahasan

Berdasarkan hasil perhitungan *confusion matrix* pada data testing yang berjumlah 205 data. Sebanyak 99 data pasien dinyatakan kecil kemungkinan terkena penyakit jantung, dan sebanyak 7 data pasien yang dinyatakan kecil kemungkinan terkena penyakit jantung diklasifikasi besar kemungkinan terkena penyakit jantung. Sebanyak 9 data pasien yang dinyatakan besar kemungkinan terkena penyakit jantung diklasifikasikan kecil kemungkinan terkena penyakit jantung dan sebanyak 90 data pasien dinyatakan besar kemungkinan terkena penyakit jantung. Berdasarkan hal tersebut hasil nilai *accuracy* sebesar 92%, *precision* sebesar 90%, dan nilai *recall* sebesar 92%.

Tabel 1. Tabel Score

| Accuracy Score | Precision Score | Recall Score |
|----------------|-----------------|--------------|
| 0.92           | 0.90            | 0.92         |

## 4. KESIMPULAN

Berdasarkan hasil penelitian maka dapat diambil kesimpulan bahwa penerapan algoritma *K-Nearest Neighbor* dalam klasifikasi pasien penyakit jantung bekerja dengan sangat baik ini dibuktikan dengan menggunakan nilai  $K = 3$  dengan bobot data training sebanyak 80% dan 20% data testing mendapatkan *score accuracy* sebesar 92%, *precision* sebesar 90%, dan nilai *recall* sebesar 92%.

## REFERENCES

- [1] Dr Pittara, "Penyakit Jantung - Gejala, Penyebab, dan Pengobatan," *Alo Dokter*, Mar. 16, 2023.
- [2] CNN, "Ingat, Penyakit Jantung Masih Jadi Pembunuh Nomor satu di Dunia," *CNN Indonesia*, Sep. 29, 2022.
- [3] Cindy Mutia Annur, "Prevalensi Penyakit Jantung di Provinsi Ini Paling Tinggi di Indonesia," *Databooks*, Jan. 26, 2022.
- [4] R. Annisa, "ANALISIS KOMPARASI ALGORITMA KLASIFIKASI DATA MINING UNTUK PREDIKSI PENDERITA PENYAKIT JANTUNG," *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 3, no. 1, 2019.

- [5] D. A. Muhammad, R. Amril, and M. Siregar, "Penerapan Algoritma *K-Nearest Neighbord* Untuk Prediksi Kematian Akibat Penyakit Gagal Jantung," vol. III, no. 1, 2022, [Online]. Available: <https://www.kaggle.com/andrewmvd/heart-failure-clinical-data>.
- [6] D. Sitanggang, N. Nicholas, V. Wilson, A. R. A. Sinaga, and A. D. Simanjuntak, "IMPLEMENTASI DATA MINING UNTUK MEMPREDIKSI PENYAKIT JANTUNG MENGGUNAKAN METODE *K-NEAREST NEIGHBOR* DAN LOGISTIC REGRESSION," *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, vol. 5, no. 2, p. 493, Dec. 2022, doi: 10.37600/tekinkom.v5i2.698.
- [7] A. Samosir, M. Hasibuan, W. E. Justino, and T. Hariyono, "Komparasi Algoritma Random Forest, Naïve Bayes dan *K-Nearest Neighbor* Dalam klasifikasi Data Penyakit Jantung".
- [8] H. W. Dhany, *Seminar Nasional Informatika (SENATIKA) Prosiding SENATIKA 2021 Performa Algoritma K-Nearest Neighbour dalam Memprediksi Penyakit Jantung*. [Online]. Available: <https://www.kaggle.com/>
- [9] S. T. M. K. Yahya, *Data Mining*. CV Jejak (Jejak Publisher), 2022. [Online]. Available: <https://books.google.co.id/books?id=0J2mEAAAQBAJ>
- [10] S. S. M. K. Muhammad Arhami and S. T. M. T. Muhammad Nasir, *Data Mining - Algoritma dan Implementasi*. Andi Offset, 2020. [Online]. Available: <https://books.google.co.id/books?id=AtcCEAAAQBAJ>
- [11] E. Buulolo, *Data Mining Untuk Perguruan Tinggi*. Deepublish, 2020. [Online]. Available: [https://books.google.co.id/books?id=-K\\_SDwAAQBAJ](https://books.google.co.id/books?id=-K_SDwAAQBAJ)
- [12] Y. Ardilla *et al.*, *DATA MINING DAN APLIKASINYA*. Penerbit Widina, 2021. [Online]. Available: <https://books.google.co.id/books?id=53FXEAAAQBAJ>
- [13] A. Wulandari and M. K. Agus Sifaunajah, *Perbandingan Efektivitas Klasifikasi Algoritma C.4.5 dan Algoritma Naive Bayes dengan Menggunakan Pihak ke 3 (WEKA)*. Lembaga Penelitian dan Pengabdian kepada Masyarakat Universitas KH. A. Wahab Hasbullah, 2019. [Online]. Available: <https://books.google.co.id/books?id=s6t-EAAAQBAJ>
- [14] D. P. Utomo and M. Mesran, "Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 4, no. 2, p. 437, Apr. 2020, doi: 10.30865/mib.v4i2.2080.
- [15] I. Id, *Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Python*. 2021. doi: 10.5281/zenodo.5113507.
- [16] D. Cahyanti, A. Rahmayani, and S. Ainy Husniar, "Indonesian Journal of Data and Science Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara," vol. 1, no. 2, pp. 39–43, 2020.
- [17] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," 2021.
- [18] L. Afifah, "Apa itu Confusion Matrix di Machine Learning?," <https://ilmudatapy.com/apa-itu-confusion-matrix/>.
- [19] Dicoding Intern, "Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya," <https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/>, May 31, 2023.
- [20] D. Lapp, "Heart Disease Dataset," <https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>, 2023.