

IMPLEMENTASI KLASIFIKASI POTENSI PENYAKIT JANTUNG DENGAN MENGUNAKAN METODE C4.5 BERBASIS WEBSITE (STUDI KASUS KAGGLE.COM)

Alfian Rahmat, M. Syafiih, Mochammad Faid

^{1,2}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Nurul Jadid

Email: hasimasyari1304@gmail.com

ABSTRACT

Heart disease is one of the leading causes of death worldwide. Therefore, early detection and classification of potential heart diseases are very important in prevention and early treatment efforts. In this study, we implemented the C4.5 classification method to analyze the dataset obtained from the Kaggle.com platform. This research aims to build an efficient and accurate classification model to identify potential heart diseases based on relevant medical features. The dataset used includes demographic information, medical history, and other health parameters of a number of patients. The results obtained from using the C4.5 method to make predictions get 85% accuracy so that it can be expected to be a source of information for further research on predictions using the C4.5 method.

Keywords: C4.5, Decision Tree, Heart Disease, Classification, Website

ABSTRAK

Penyakit jantung adalah salah satu penyebab utama kematian seluruh dunia. Oleh karena itu, deteksi dini dan klasifikasi potensi penyakit jantung menjadi sangat penting dalam upaya pencegahan dan pengobatan dini. Dalam penelitian ini, kami mengimplementasikan metode klasifikasi C4.5 untuk menganalisis dataset yang diperoleh dari platform Kaggle.com. Penelitian ini bertujuan untuk membangun sebuah model klasifikasi yang efisien dan akurat untuk mengidentifikasi potensi penyakit jantung berdasarkan fitur-fitur medis yang relevan. Dataset yang digunakan mencakup informasi demografis, riwayat medis, dan parameter kesehatan lainnya dari sejumlah pasien. Hasil yang didapat dari penggunaan metode C4.5 untuk melakukan prediksi mendapatkan akurasi 85% sehingga bias diharapkan menjadi sumber informasi untuk penelitian selanjutnya tentang prediksi dengan menggunakan metode C4.5.

Kata Kunci: C4.5, Pohon Keputusan, Penyakit Jantung, Klasifikasi, Website

Riwayat Artikel :

Tanggal diterima : 28-07-2023

Tanggal revisi : 29-07-2023

Tanggal terbit : 30-07-2023

DOI :

<https://doi.org/10.31949/infotech.v9i2.6295>

INFOTECH journal by Informatika UNMA is licensed under CC BY-SA 4.0

Copyright © 2023 By Author



1. PENDAHULUAN

1.1. Latar Belakang

Penyakit jantung merupakan salah satu penyakit berbahaya. Keterlambatan pengobatan penyakit ini akan membahayakan nyawa penderitanya. Penyakit ini mempengaruhi struktur atau fungsi jantung seperti penyakit arteri, serangan jantung aritma, gagal jantung, penyakit vascular, dan lainnya.

Berdasarkan Global Burden of Disease dan Institute for Health Metrics and Evaluation (IHME) 2014 – 2019 pwnyakit jantung menjadi penyebab kematian tertinggi di Indonesia. Data Riset Kesehatan Dasar (Riskesdas) 2013 dan 2018 menunjukkan tren peningkatan penyakit jantung yakni 0,5% pada 2013 menjadi 1,5% pada 2018. Dr. Radityo mengatakan, data terbaru Organisasi Kesehatan Dunia (WHO) menunjukkan bahwa penyakit jantung dan stroke masih menduduki pertama dan kedua penyebab kematian utama di dunia. Adapun jumlah kematian akibat penyakit jantung ini secara global mencapai 18,6 juta orang setiap tahunnya. (Kompas.com, 4 Agustus 2022). Masalah ini disebabkan sulitnya deteksi dini pada penderita penyakit jantung karena selalu mengabaikan gejala awal.

Algoritma C4.5 merupakan salah satu algoritma dalam data mining yang berfungsi untuk dari algoritma ID3. Cara kerja algoritma C4.5 dengan membentuk pohon keputusan untuk menghasilkan sebuah keputusan. Kelebihan ini misalnya dapat mengolah data numerik dan diskret, dapat menangani nilai atribut yang hilang, menghasilkan aturan- aturan yang mudah diinterpretasikan dan tercepat di antara algoritma algoritma yang lain.

Kaggle adalah salah satu situs yang terkenal di dunia Data Science dan Machine Learning yang terdiri dari lebih dari 6000 dataset yang dapat diunduh dalam format CSV. Kaggle sangat bermanfaat bagi kamu yang sedang mempelajari Data Science. Dataset ini banyak membantu ilmuwan di seluruh dunia untuk membuat model

1.2. Tinjauan Pustaka

Pertama, penelitian oleh Andi (2021) yang berjudul Implementasi Algoritma Decision Tree C4.5 Dalam perancangan Sistem Informasi Data Rekam Medis Penyakit Jantung. Dalam penelitian Pada praktiknya, di dunia pendidikan kedokteran seringkali data rekam medis penyakit jantung disimpan dengan tujuan untuk proses pembelajaran ataupun diolah menjadi sebuah pengetahuan. Pada website <https://www.kaggle.com/>, banyak disediakan dataset yang disimpan dan diolah dengan tujuan pembelajaran, meliputi dataset penyakit, dataset populasi ataupun jenis dataset lainnya. Namun kenyataannya, untuk mengolah data rekam medis tersebut menjadi sebuah pengetahuan tidaklah mudah, dikarenakan banyaknya jumlah data rekam medis yang tercatat tidak memungkinkan bagi manusia dalam mengolahnya. Selain itu, pengelolaan data rekam

medis secara konvensional memiliki tingkat keakuratan yang kurang baik sehingga hasil kesimpulan yang diambil pun tentunya akan berbeda dan tidak begitu akurat. Metode yang digunakan dalam penelitian ini adalah penerapan Data Mining menggunakan metode Algoritma C4.5. Hasil penelitian ini menyatakan Hasil penelitian menunjukkan bahwa penerapan algoritma Decision Tree C4.5 tergolong cukup akurat dalam melakukan prediksi terhadap data rekam medis penyakit jantung dimana hasil pengujian dengan mendapatkan akurasi sebesar 96%.

1.3. Metodologi Penelitian

Metodologi Penelitian Implementasi Klasifikasi Potensi Terkena Penyakit Jantung Menggunakan Metode C4.5 Berbasis Website dapat melibatkan beberapa langkah berikut:

1. Pengumpulan Data

Data yang diperoleh menggunakan teknik sekunder, Data diambil dari situs Kaggle Dataset yang dapat diakses melalui alamat website Data yang digunakan berupa atribut dan label diantaranya: *age, sex, chest paint type, resting bps, cholesterol, fasting blood sugar, resting ecg, max heart rate, exercise angina, oldpeak, ST slope* sebagai atribut dan *target* sebagai label.

2. Pembersihan Dan Praproses Data

- Lakukan pembersihan data untuk mengatasi nilai yang hilang atau tidak valid.
- Lakukan pemilihan atribut (fitur) yang paling relevan untuk digunakan dalam klasifikasi

3. Membuat dan Evaluasi Model *Decision Tree*

Setelah data diproses, langkah selanjutnya adalah membangun model klasifikasi menggunakan algoritma C4.5. Algoritma ini akan menghasilkan pohon keputusan berdasarkan atribut yang ada dalam data. Pohon keputusan ini akan digunakan untuk melakukan prediksi pada data baru.

Berikut adalah rumus perhitungan yang terdapat pada *Decision Tree* yaitu *Entropy* dan *Gain*.

$$Entropy (S) = \sum_{i=1}^n -p_i * \log_2 p_i \quad (1)$$

$$Gain (S, A) Entropy (S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy (S_i) \quad (2)$$

Implementasi algoritma C4.5 menggunakan bahasa Pemrograman *Python* dengan bantuan *Jupyter Notebook Framework Streamlit* guna membuat model klasifikasi potensi terkena penyakit jantung dengan metode C4.5

4. Simulasi Model

Sebelum model diterapkan kedalam *Website* perlu dilakukan simulasi model. Simulasi model ini berfungsi mengetahui apakah model yang digunakan berfungsi atau akurat data yang

digunakan data dari data yang sudah ada sebelumnya.

5. Implementasi Website

Langkah selanjutnya adalah integrasikan model klasifikasi C4.5 ke dalam Website yang sudah dibuat sehingga pengguna dapat mengunggah data pasien data pasien dan melihat hasil klasifikasi penyakit jantung. Aplikasi Website yang sudah dibuat menggunakan bahasa pemrograman Python dengan menggunakan Framework Streamlit. Dalam proses ini juga menampilkan hasil dari Website yang sudah dibuat.

2. PEMBAHASAN

Implementasi sistem merupakan

Isi didukung dengan gambar, tabel, dan persamaan yang dirujuk dalam naskah.

2.1. Pengumpulan Data

Data yang digunakan berupa atribut dan label diantaranya *age, sex, chest paint type, resting bps, cholesterol, fasting blood sugar, resting ecg, max heart rate, exercise angina, oldpeak, ST slope* sebagai atribut dan *target* sebagai label.

Deskripsi data tersebut dapat dilihat padat Tabel 1.

Tabel 1. Deskripsi data

No	Atribut	Tipe Data	Keterangan
1	Age	Numerik	Usia
2	Sex	Numerik	Jenis kelamin
3	Chest Paint Type	Numerik	Tipe sakit dada
4	Resting Bps	Numerik	Tekanan darah
5	Cholesterol	Numerik	Kolesterol
6	Fasting Blood Sugar	Numerik	-
7	Resting Ecg	Numerik	-
8	Max Heart Rate	Numerik	Detak jantung
9	Exercise Angina	Numerik	Tipe rasa sakit didada
10	Oldpeak	Numerik	Tipe depresi
11	ST slope	Numerik	Kemiringan detak jantung
12	Target	Numerik	Label keterangan penyakit jantung atau tidak

2.2. Menghilangkan Atribut

Pada proses tersebut penulis telah memilih atribut yang berpengaruh pada proses data mining. Atribut pada data yang dipilih dan akan diolah diantaranya *Sex, Chest Pain Type, Resting Bps, Cholesterol, Fasting Blood Sugar, Resting Ecg, Max Heart Rate, Exercise Angina, Oldpeak, ST_Slope, dan Target sebagai label* dipilih Karena berpengaruh terhadap gejala gejala yang paling dialami oleh penderita jantung. Sedangkan untuk atribut yang tidak digunakan akan dihapus. Hasil dapat dilihat pada Gambar 1.

```
df = df.drop(['age'], axis=1)

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1198 entries, 0 to 1197
Data columns (total 11 columns):
#   Column              Non-Null Count  Dtype
---  ---              -
0   sex                  1198 non-null   int64
1   chest pain type     1198 non-null   int64
2   resting bp s        1198 non-null   int64
3   cholesterol          1198 non-null   int64
4   fasting blood sugar  1198 non-null   int64
5   resting ecg         1198 non-null   int64
6   max heart rate      1198 non-null   int64
7   exercise angina     1198 non-null   int64
8   oldpeak             1198 non-null   float64
9   ST slope            1198 non-null   int64
10  target              1198 non-null   int64
dtypes: float64(1), int64(10)
memory usage: 102.4 KB
```

Gambar 1. Hasil Hapus Atribut

Model yang dibuat menggunakan *Decision tree classifier*, sebuah library dari sklearn yang dapat memproses data menggunakan algoritma *decision tree*

2.3. Membuat dan Evaluasi Model Decision tree

Model yang dibuat menggunakan *DecisionTree Classifier*, sebuah library dari sklearn yang dapat memproses data menggunakan algoritma *Decision Tree* menggunakan bahasa pemrograman Python dengan bantuan *Jupyter Notebook*.

Berikut hasil dari pembuatan model dapat dilihat pada Gambar 2.

```
...
akurasi data training = 0.851890756302521
akurasi data testing = 0.8361344537815126

confusion matrix :
[[ 83 26]
 [ 13 116]]

classification report :
              precision    recall  f1-score   support

     0       0.86         0.76         0.81         109
     1       0.82         0.90         0.86         129

   accuracy          0.84
  macro avg          0.84         0.83         0.83         238
 weighted avg          0.84         0.84         0.83         238
```

Gambar 2. Membuat Model Decision Tree

Dari hasil membuat model dan evaluasi model didapatkan akurasi data training diatas adalah 85 % ,data testing 83 % dan akurasi yang dihasilkan dari algoritma *Decision Tree* untuk data diatas menghasilkan 84 %

2.4. Simulasi Model

Setelah sebelumnya telah melakukan proses membuat model dan evaluasi model *Decision tree*, selanjutnya melakukan simulasi model. Simulasi model ini dilakukan atau dibuat menggunakan *Jupyter Notebook* Simulasi model dapat dilihat pada Gambar 3.

```

simulasi model
In [ ]:
input_data = (1,1,41,147,9,9,96,6,8,1)
input_data_as_numpy_array = np.array(input_data)
input_data_reshape = input_data_as_numpy_array.reshape(1,-1)
prediction = model.predict(input_data_reshape)
print(prediction)

if (prediction[0]==0):
    print ("Pasien tidak terkena penyakit jantung")
else:
    print ("Pasien terkena penyakit jantung")

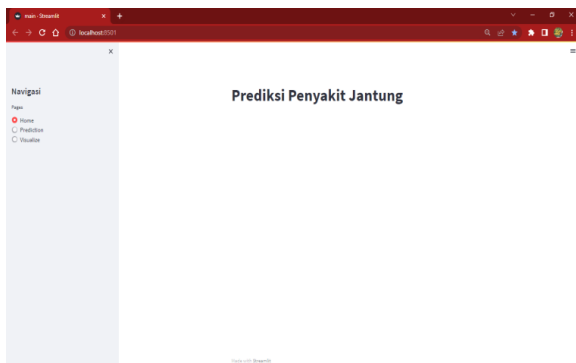
Out [ ]:
[0]
Pacien tidak terkena penyakit jantung
C:\Users\alfira\anaconda3\lib\site-packages\sklearn\base.py:466: UserWarning: X does not have valid feature names, but DecisionTreeClassifier
warns.warn(
    
```

Gambar 3. Simulasi Model

Dalam gambar di atas simulasi model bertujuan untuk mencoba apakah model yang dibuat berdasarkan dataset itu akurat yang nantinya akan di implementasikan ke dalam *Website*. Data yang digunakan berdasarkan data yang ada di dataset, hasil yang didapatkan dari simulasi model diatas terbukti akurat output yang mana hasilnya 0 sama dengan pasien tidak terkena penyakit jantung.

2.5. Implementasi Website

Setelah dilakukan simulasi model langkah selanjutnya mengimplementasikan model *Decision Tree (C4.5)* ke dalam *Website* yang sudah dibuat dan juga melihat halaman yang ada dalam *Website*. Dapat dilihat pada Gambar 4.

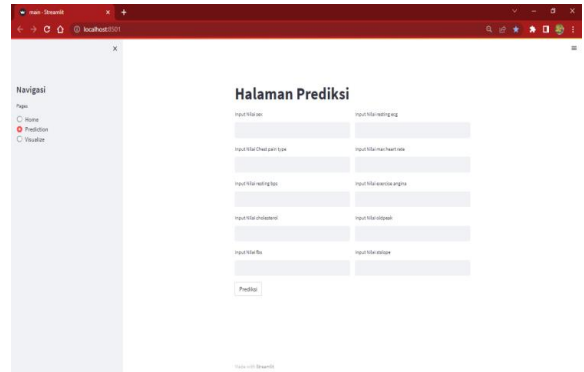


Gambar 4. Halaman Home

Berdasarkan Gambar diatas ada *Sidebar* yang terdiri dari *Home*, *Prediksi*, dan *Visualize* pada halaman pertama yaitu halaman *Home* yang berisi judul *Prediksi Penyakit Jantung*.

2.6. Halaman Prediksi

Pada halaman *Prediksi* menunjukkan beberapa *Feature* yang berupa *Teks Input* dan tombol *Button* untuk memprediksi apakah pasien termasuk dalam kategori terkena penyakit jantung atau tidak.



Gambar 5. Halaman Prediksi

2.7. Hasil Uji Data

Sebelumnya sudah dijelaskan pada halaman . Untuk melihat halaman *Prediksi* berjalan atau tidak dapat dilihat pada Gambar 6.

Halaman Prediksi

Input Nilai sex: 0	Input Nilai resting ecg: 0
Input Nilai Chest pain type: 4	Input Nilai max heart rate: 108
Input Nilai resting bps: 138	Input Nilai exercise angina: 1
Input Nilai cholesterol: 214	Input Nilai oldpeak: 1.5
Input Nilai fbs: 0	Input Nilai ststlope: 2

Prediksi

Prediksi Sukses...

Orang tersebut rentan terkena penyakit jantung

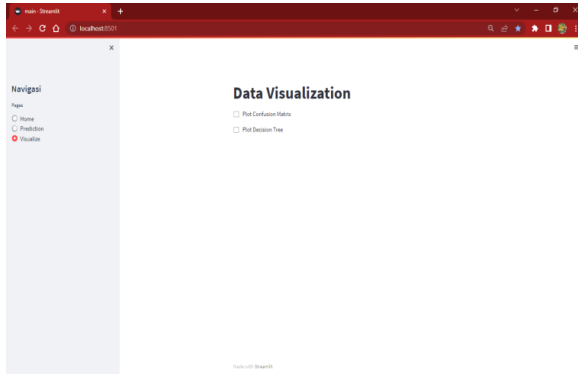
Model yang digunakan memiliki tingkat akurasi 85.4347386946778 %

Gambar 6. Hasil Uji Data

Dari hasil pengujian diatas hasil uji data berjalan sesuai dengan yang diharapkan pengujian menggunakan data yang ada dan hasil dari pengujian memperoleh hasil 1 sama dengan orang tersebut rentan terkena penyakit jantung jika tidak maka 0 sama dengan orang tersebut relatif aman dari penyakit jantung dan dalam gambar diatas juga menampilkan tingkat akurasi yang dihasilkan oleh model adalah 85%

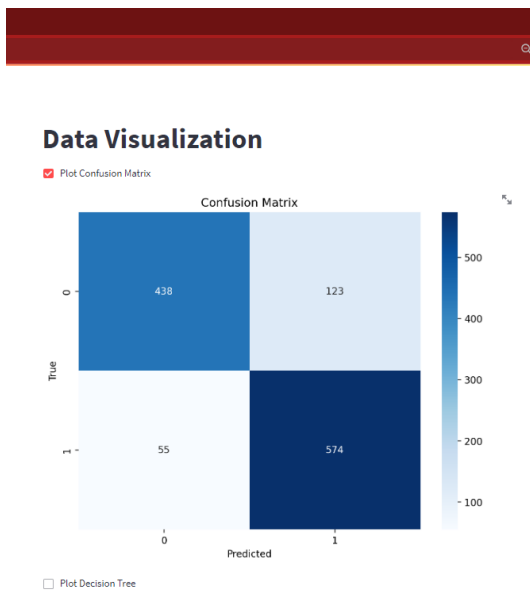
2.8. Halaman *Visualize*

Dari tampilan halaman *Visualize* terdiri dari dua *Checkbox* yang berisi *Plot Confusion Matrix* dan *Plot Decision Tree* ditunjukkan dalam Gambar 7.



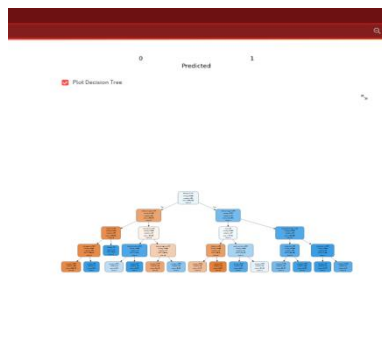
Gambar 7. Halaman *Visualize*

ketika *Checkbox Plot Confusion Matrix* dan *Plot Decision Tree* di pilih maka akan menampilkan hasil berikut. Adapun hasilnya dapat dilihat pada Gambar 8



Gambar 8. Halaman *Plot Confusion Matrix*

Dan untuk melihat halaman *Plot Decision Tree* di tunjukkan pada Gambar 9.



Gambar 9. Halaman *Plot Decision Tree*

3. ALGORITMA ATAU PROGRAM

3.1. *Main.py*

Program dapat dilihat dalam Gambar 10.

```
# import library yang dibutuhkan

import streamlit as st

from web_functions import load_data

from Tabs import home, predict, visualise

Tabs = {
    "Home": home,
    "Prediction": predict,
    "Visualize": visualise
}

# membuat sidebar
st.sidebar.title("Navigasi")

#membuat radio option
page = st.sidebar.radio("Pages",
list(Tabs.keys()))

#load dataset
df, x, y = load_data()

#kondisi call app function
if page in ["Prediction",
"Visualize"]:
    Tabs[page].app(df, x, y)
else:
    Tabs[page].app()
```

Gambar 10. Code *Main.py*

3.2. *Home.py*

Program dapat dilihat dalam Gambar 11.

```
import streamlit as st

def app():
    st.title("Prediksi Penyakit Jantung")
```

Gambar 11. Code *Home.py*

3.3. Predict.py

Program dapat dilihat dalam Gambar 12.

```
import streamlit as st

from web_functions import predict

def app(df, x, y):

    st.title("Halaman Prediksi")

    col1, col2 = st.columns(2)

    with col1:
        sex = st.text_input
        ('Input Nilai sex')
        with col1:
            chest_pain_type =
            st.text_input ('Input Nilai Chest
            pain type')
            with col1:
                resting_bp_s =
                st.text_input ('Input Nilai
                resting bps')
                with col1:
                    cholesterol =
                    st.text_input ('Input Nilai
                    cholesterol')
                    with col1:
                        fasting_blood_sugar =
                        st.text_input ('Input Nilai fbs')
                        with col2:
                            resting_ecg =
                            st.text_input ('Input Nilai
                            resting ecg')
                            with col2:
                                max_heart_rate =
                                st.text_input ('Input Nilai max
                                heart rate')
                                with col2:
                                    exercise_angina =
                                    st.text_input ('Input Nilai
                                    exercise angina')
                                    with col2:
                                        oldpeak = st.text_input
                                        ('Input Nilai oldpeak')
                                        with col2:
                                            ST_slope = st.text_input
                                            ('Input Nilai stslope')
```

```
features = [sex,
chest_pain_type, resting_bp_s,
cholesterol, fasting_blood_sugar,
resting_ecg, max_heart_rate,
exercise_angina, oldpeak,
ST_slope]

#tombol prediksi
if st.button("Prediksi"):
    prediction, score =
    predict(x,y,features)
    score = score
    st.info("Prediksi
    Sukses...")

    if (prediction == 1):
        st.warning("Orang
        tersebut rentan terkena penyakit
        jantung")
    else:
        st.success("Orang
        tersebut relatif aman dari
        penyakit jantung")

    st.write("Model yang
    digunakan memiliki tingkat
    akurasi", (score*100), "%")
```

Gambar 12. Code Predict.py

3.4. Visualize.py

Program dapat dilihat dalam Gambar 13.

```
import warnings
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import
confusion_matrix
from sklearn import tree
import streamlit as st

from web_functions import
train_model

def plot_confusion_matrix(y_true,
y_pred):
    cm = confusion_matrix(y_true,
y_pred)
    sns.heatmap(cm, annot=True,
fmt='d', cmap='Blues')
    plt.xlabel('Predicted')
    plt.ylabel('True')
```

```
plt.title('Confusion Matrix')

def app(df, x, y):

warnings.filterwarnings('ignore')

st.set_option('deprecation.showPyplotGlobalUse', False)

st.title('Data Visualization')

if st.checkbox("Plot Confusion Matrix"):

    model, score = train_model(x,y)
    y_pred = model.predict(x)
    plt.figure(figsize=(8, 6))
    plot_confusion_matrix(y, y_pred)
    st.pyplot()

if st.checkbox("Plot Decision Tree"):

    model, score = train_model(x,y)
    dot_data = tree.export_graphviz(
        decision_tree=model,
        max_depth=5, out_file=None,
        filled=True, rounded=True,

        feature_name=x.columns,
        class_names=['0', '1']
    )

st.graphviz_chart(dot_data)
```

Gambar 13. Code Visualize.py

3.5. Web_Functions.py

Program dapat dilihat pada Gambar 14.

```
# Import modul yang akan digunakan
from sklearn.model_selection
import train_test_split
import numpy as np
import pandas as pd
from sklearn.tree import
DecisionTreeClassifier
import streamlit as st
```

```
@st.cache_data()
def load_data():
    # load dataset
    df = pd.read_csv('jantung.csv')

    x = df[["sex", "chest
pain type", "resting bp s",
"cholesterol", "fasting
blood sugar", "resting ecg",
"max heart rate",
"exercise angina",
"oldpeak", "ST slope"]]
    y = df[["target"]]

    return df, x, y

@st.cache_data()
def train_model(x,y):
    x_train, x_test, y_train,
y_test = train_test_split(
        x, y, test_size=0.3,
        random_state=42, stratify=y,
        shuffle=True
    )
    model = DecisionTreeClassifier(
        ccp_alpha=0.0,
        class_weight=None,
        criterion='entropy',
        max_depth=4,
        max_features=None,
        max_leaf_nodes=None,

        min_impurity_decrease=0.0,
        min_samples_leaf=1,

        min_samples_split=2,
        min_weight_fraction_leaf=0.0,
        random_state=42,
        splitter='best'
    )
    model.fit(x_train,y_train)
    score = model.score(x_test,y_test)
    return model, score

@st.cache_data()
def predict(x,y, features):
    model, score = train_model(x,y)
```



```

prediction =
model.predict(np.array(features).r
eshape(1, -1))

return prediction, score

```

Gambar 14. Code Functions.py

4. KESIMPULAN

Penelitian ini menunjukkan bahwa metode C4.5 berbasis *Website* dapat menjadi alat yang berguna dalam mendukung proses deteksi dini dan klasifikasi penyakit jantung. Namun, penggunaan *Website* ini harus diimbangi dengan evaluasi oleh professional medis yang terlatih untuk mengonfirmasi diagnosis dan meresepkan perawatan yang sesuai. Penelitian ini berpotensi memberikan dampak positif dalam upaya pencegahan dan pengelolaan penyakit jantung secara keseluruhan dan dapat menjadi landasan bagi pengembangan lebih lanjut dalam bidang ini.

PUSTAKA

- Abu Tholib, M.Kom. 2023. Implementasi Algoritma Machine Learning Berbasis Web dengan Framework Streamlit. Probolinggo: Pustaka Nurja.
- Andi. 2021. 'Implementasi Algoritma Decision Tree C4.5 Dalam Perancangan Sistem Informasi Data Rekam Medis Penyakit Jantung'. *Jurnal Ilmiah Smart*, Vol. V, No. 1, hal. 275-282.
- Arni, S., Iwansyah, E.H., & Chaerur, R. 2022. 'Klasifikasi Penyakit Jantung Dengan Menggunakan Algoritma C4.5'. *STRING(Satuan Tulisan Riset dan Inovasi Teknologi)*, Vol. 7, No. 2.
- H. Kartika, N.S. 2022. Angka Kematian Penyakit Jantung Masih Tinggi Di Indonesia. *Kompas.com*, (Online), (<https://www.kompas.com/sains/read/2022/08/04/163200923/angka-kematian-penyakit-jantung-masih-tinggi-di-indonesia-jadi-tantangan?page=all>).
- Muhammad, Agus, P.W., & Suhada. 2019. 'Penerapan Algoritma C4.5 Pada Klasifikasi Potensi Siswa Drop Out'. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, Vol. 3, No. 1, Page: 753-760.
- Muhammad, Y., Hanandriya, R., Dimas, R.A., & Agus, Y. 2021. 'Penerapan Metode Data Mining C4.5 Untuk Pemilihan Penerimaan Kartu Indonesia Pintar (KIP)'. *Paradigma*, Vol. 23, No. 2.
- M., Fikri. H., Sutikno. 2021. 'Implementasi Algoritma C4.5 Untuk Klasifikasi Penyakit Infeksi Saluran Kemih Berbasis Web'. *Jurnal Ilmiah Informatika Komputer*, Vol. 26, No. 3.
- Popy Meilina. 2015. 'Penerapan Data Mining Dengan Metode Klasifikasi Menggunakan Decision Tree Dan Regresi'. *Jurnal Teknologi*, Vol. 7, No. 1.
- Rokom. 2022. Penyakit Jantung Penyebab Utama Kematian. (Online). Biro Komunikasi dan Pelayanan Publik, Kementerian Kesehatan RI, (<https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20220929/0541166/penyakit-jantung-penyebab-utama-kematian-kemenkes-perkuat-layanan-primer/>).
- Rizky, H.P., Budi, D.S., & Indriati. 2018. 'Penerapan Algoritma C4.5 Untuk Memprediksi Nilai Kelulusan Siswa Sekolah Menengah Berdasarkan Faktor Eksternal'. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 2, No7, hlm. 2637-2643.
- Siti Riska, J.I.A., Efy Yosrita & Rizqia Cahyaningtyas. 2021. 'Sistem Diagnosis Penyakit Koroner Dengan Menggunakan Algoritma C4.5 Berbasis Website (Studi Kasus: RSUD Dr. Soedarso Pontianak)'. *PETIR (Jurnal Pengkajian dan Penerapan Teknik Informatika)*, Vol. 14, No. 2.