Memprediksi Kemungkinan Seseorang Mengalami Serangan Jantung Menggunakan Algoritma Decision Tree

Muhamad Calvin Syah Putra[1], Nurul Aini Lativah[2], Pradnja Paramita Cendana Wangi[3], Vernonia Novianna Putri[4], Yuda Ramadhoni[5]

[1] Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia
muhammad.putra@student.umn.ac.id
[2] Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia
nurul.lativah@student.umn.ac.id
[3] Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia
pradnja.wangi@student.umn.ac.id
[4] Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia
vernonia.novianna@student.umn.ac.id
[5] Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia
yuda.ramadhoni@student.umn.ac.id

Abstrak—Penelitian ini dilakukan untuk mengklasifikasi dan mengidentifikasi gejala-gejala yang menyebabkan penyakit serangan jantung pada seseorang dengan menggunakan teknik *Data Mining*. Algoritma *Decision Tree* merupakan salah satu teknik *data mining* yang dapat melakukan klasifikasi dengan mengolah variabel-variabel yang ada pada dataset. Variabel tersebut diklasifikasikan sesuai dengan atributnya. Algoritma *Decision Tree* memecah ke dalam himpunan bagian yang lebih kecil lalu pada saat itu juga sebuah pohon keputusan secara bertahap dikembangkan. Dengan Algoritma *Decision Tree* dapat memberikan informasi prediksi untuk menggambarkan prosess yang terkait dengan prediksi gejala-gejala yang menyebabkan serangan jantung. Karakteristik data yang diklasifikasi dapat diperoleh dengan jelas, baik dalam bentuk struktur pohon keputusan maupun aturan sehingga dalam tahap pengujian dengan *software Anaconda* menggunakan bahasa pemrograman *Python* dapat membantu dalam memprediksi gejala serangan jantung. Pembuatan model klasifikasi dengan *machine learning* akan meningkatkan ketepatan pembuatan keputusan karena dapat menguji menggunakan ukuran data yang lebih besar daripada ukuran data pada metode tradisional. Model dengan ketepatan dan kinerja yang baik akan mempermudah dan mempercepat proses identifikasi penyakit jantung pada seseorang.

Kata kunci—*Machine Learning*; *Heart Attack*; *Decision Tree*; *Disease*

Abstract—This study was conducted to classify and identify the symptoms that cause heart attack in a person using Data Mining techniques. Decision Tree Algorithm is one of the data mining techniques that can perform classification by processing the variables that exist in the dataset. These variables are classified according to their attributes. The Decision Tree algorithm breaks down into smaller subsets and then a decision tree is gradually developed. With the Decision Tree Algorithm, it can provide predictive information to describe the processes associated with predicting the symptoms that cause a heart attack. The characteristics of the classified data can be obtained clearly, both in the form of a decision tree structure and rules so that in the testing phase with Anaconda software using the Python programming language can help predict the symptoms of a heart attack. Making a classification model with machine learning will increase the accuracy of decision making because it can test using a larger data size than the data size in traditional methods. Models with accuracy and good performance will simplify and speed up the process of identifying heart disease in a person.

Keyword—Machine Learning*; *Heart Attack*; *Decision Tree*; *Disease

I. Latar Belakang

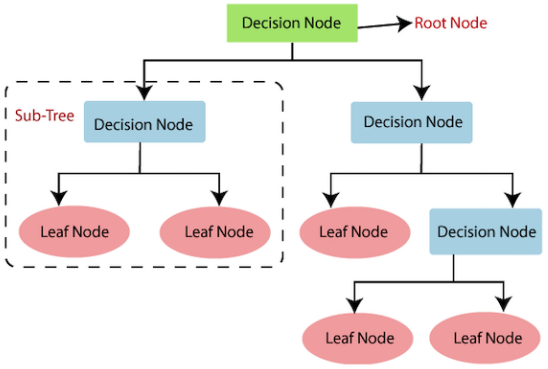
Kesehatan merupakan faktor yang sangat penting yang harus diperhatikan oleh setiap individu. Menurut UU N0. 23/1992 tentang kesehatan, kesehatan adalah suatu keadaan sejahtera dari badan (jasmani), jiwa (rohani) dan sosial yang memungkinkan setiap orang hidup produktif secara sosial dan ekonomis. Kesehatan bisa menurun dikarenakan beberapa faktor. Banyak berbagai penyakit yang bisa dialami oleh berbagai individu, salah satunya penyakit jantung. Penyakit jantung merupakan salah satu dari berbagai bentuk kondisi kesehatan yang kurang baik dan masuk dalam kategori kondisi gangguan kesehatan yang serius. Jantung merupakan organ tubuh utama manusia serta penyakit pada jantung merupakan penyakit mematikan yang sangat berbahaya bagi manusia. Menurut WHO *(World Health Organization),* penyakit jantung dan pembuluh darah (*kardiovaskuler*) merupakan penyebab meninggalnya 50% dari 12 juta penduduk di dunia. Di Indonesia, Penyakit Tidak Menular (PTM) seperti penyakit jantung koroner, gagal jantung, hipertensi dan stroke menyebabkan lebih sekitar 17 juta orang meninggal dunia pada tahun 2018 (Rachmawati, Martini & Artanti, 2021).

Serangan jantung adalah kondisi yang melemahkan, yang disebabkan karena berbagai alasan seperti gaya hidup tidak sehat, pola makan dan olahraga yang tidak tepat, stres, dll., yang dapat merusak otot jantung. Hal ini dapat menghambat peredaran darah, dan akhirnya menjadi mengancam jiwa. Gejala yang biasa dirasakan termasuk sesak atau nyeri di dada, leher, punggung atau lengan, serta kelelahan, pusing, detak jantung abnormal dan kecemasan. Lebih lanjut, menurut para ahli, kadar kolesterol yang tinggi dan tekanan gula darah yang tinggi merupakan salah satu faktor yang dapat memicu adanya resiko penyakit jantung (Alimansur dan Irawan, 2017). Langkah yang dapat ditempuh untuk meminimalisir terjadinya penyakit serangan jantung adalah dengan memperbaiki pola hidup. Dengan melakukan analisa yang akurat akan dihasilkan sebuah diagnosa atau prediksi yang di dalamnya berisi mengenai faktor pemicu penyakit serangan jantung sehingga membantu meminimalisir penyakit serangan jantung.

Oleh karena itu, tidak mudah untuk mendiagnosis penyakit serangan jantung pada seseorang. Selain itu, terdapat kemungkinan seseorang mengalami gejala tersebut dan benar mengidap penyakit serangan jantung, tetapi orang tersebut hanya menganggap gejala tersebut sebagai gejala penyakit biasa. Penelitian kali ini, kami akan menggunakan *Decision Tree* untuk mengklasifikasikan apakah subjek kemungkinan mengalami serangan jantung atau tidak. Agar dapat menyimpulkan kemungkinan seseorang terkena serangan jantung atau tidak berdasarkan gejala-gejala yang ada.

II. KAJIAN PUSTAKA

Salah satu teknik yang banyak digunakan dalam *data mining* adalah sistem yang membuat klasifikasi. Dalam *data mining*, algoritma klasifikasi mampu menangani sejumlah besar informasi. *Decision Tree* dapat digunakan untuk membuat asumsi mengenai nama kelas kategorikal, untuk mengklasifikasikan pengetahuan berdasarkan *train set* dan label kelas, dan untuk mengklasifikasikan data yang baru diperoleh. Algoritma klasifikasi dalam *machine learning* berisi beberapa algoritma, dan dalam penelitian ini difokuskan pada algoritma *Decision Tree* secara umum.



Gambar 1. Struktur *Decision Tree*

Decision Tree merupakan salah satu metode ampuh yang biasa digunakan di berbagai bidang, seperti *machine learning*, *image processing*, dan identifikasi pola. DT adalah model berurutan yang menyatukan serangkaian tes dasar secara efisien dan kohesif dimana fitur numerik dibandingkan dengan nilai ambang di setiap pengujian. Aturan konseptual jauh lebih mudah untuk dibangun daripada bobot numerik dalam jaringan saraf koneksi antara *node*. DT digunakan terutama untuk tujuan pengelompokan. Selain itu, DT adalah model klasifikasi yang biasanya digunakan dalam *Data Mining*. *Node* dan cabang terdiri dari setiap pohon. Setiap *node* mewakili fitur dalam kategori yang akan diklasifikasikan dan setiap subset mendefinisikan nilai yang dapat diambil oleh *node* (Jijo & Abdulazeez, 2021).

Terdapat beberapa penelitian yang telah dilakukan sebelumnya. Pal & Parija mengimplementasikan algoritma *Random Forest data mining* untuk memprediksi penyakit jantung. Dari hasil percobaan diperoleh nilai *sensitivity* 90,6%. nilai *specificity* 82,7%, dan nilai *accuracy* 86,9% untuk prediksi. Mereka memperoleh *classification accuracy* sebesar 86,9% untuk prediksi penyakit jantung dengan tingkat diagnosis sebesar 93,3% menggunakan algoritma *Random Forest*. Sistem ini juga dapat digunakan untuk prediksi penyakit lain dengan menerapkan algoritma *machine learning* lainnya seperti *Naïve Bayes*, *Decision Tree*, *K-NN*, *Linear Regression*, *Fuzzy Logic* untuk akurasi yang lebih baik. Teknologi *cloud computing* juga dapat digunakan oleh sistem ini untuk mengelola volume data pasien yang besar (Pal & Parija, 2021).

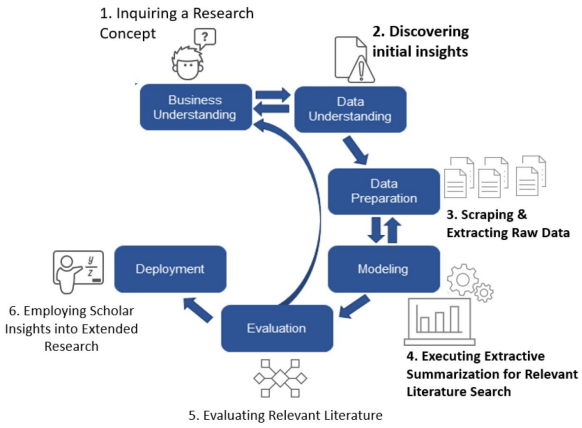
Wang dan teman-teman mengusulkan algoritma pengolahan data UCO dengan menggabungkan tiga metode, yaitu metode *undersampling*, metode *clustering*, dan metode *oversampling*. Algoritma tersebut dapat menangani data pasien stroke yang tidak seimbang. Dari data asli, dipilih delapan indikator medis yang mempengaruhi serangan jantung. Mereka membandingkan kinerja beberapa model *machine learning* dalam memprediksi serangan jantung. Hasil penelitian mereka menunjukkan bahwa *Random Forest* merupakan model terbaik untuk memprediksi kemungkinan serangan jantung pada

database MIMIC-III dari dataset pasien stroke. Akurasinya 70,29%, dan presisinya 70,05% (Wang et al., 2021).

Sedangkan Jindal dan teman-temannya, membuat sebuah model yang mendeteksi penyakit kardiovaskular, model ini dikembangkan menggunakan tiga teknik pemodelan klasifikasi ML. Mereka memprediksi orang dengan penyakit *kardiovaskular* dengan mengekstrak riwayat medis pasien yang mengarah ke penyakit jantung fatal dari kumpulan data yang mencakup riwayat medis pasien seperti nyeri dada, kadar gula, tekanan darah, dll. Sistem deteksi penyakit jantung ini membantu pasien berdasarkan informasi klinisnya pernah didiagnosis menderita penyakit jantung sebelumnya. Algoritma yang digunakan dalam membangun model yang diberikan adalah *Logistic Regression*, *Random Forest Classifier* dan KNN. Keakuratan model mereka adalah sebesar 87,5% (Jindal et al., 2021).

III. METODOLOGI

Peneliti melakukan data mining menggunakan metodologi CRISP-DM untuk memberikan gambaran umum tentang *life cycle* dari *data mining*. *Life cycle* dari *data mining* dibagi dalam enam fase yang ditunjukkan pada Gambar 1.



Gambar 2. CRISP-DM

A. *Business Understanding*

Tujuan bisnis (penelitian) ini untuk mengklasifikasi apakah subyek kemungkinan mengalami serangan jantung atau tidak. Kebutuhan tujuan bisnis (penelitian) ini untuk mengetahui berdasarkan pola dari penyakit pasien berdasarkan pada gejala yang dirasakan.

B. *Data Understanding dan Preparation*

Pada fase ini data yang akan digunakan dilakukan pemeriksaan *null*, korelasi dan EDA untuk melihat *outlier* pada setiap variabel, *outlier* yang besar akan dihapus. Peneliti menggunakan dataset dari kaggle.com. Berikut adalah variabel yang terdapat di dalam dataset *heart.csv*:

1. *age*: umur seseorang dalam tahun
2. *sex*: Jenis kelamin orang tersebut (1 = laki-laki, 0 = perempuan)
3. *cp*: Nyeri dada yang dialami (Nilai 1: angina tipikal, Nilai 2: angina atipikal, Nilai 3: nyeri non-angina, Nilai 4: asimtomatik)
4. *trestbps*: Tekanan darah istirahat seseorang (mmHg saat masuk ke rumah sakit)
5. *chol*: Pengukuran kolesterol seseorang dalam mg/dl
6. *fbs*: Gula darah puasa orang tersebut (> 120 mg/dl, 1 = benar; 0 = salah)
7. *restecg*: Pengukuran elektrokardiografi saat istirahat (0 = normal, 1 = memiliki kelainan gelombang ST-T, 2 = menunjukkan kemungkinan atau pasti hipertrofi ventrikel kiri menurut kriteria Estes)
8. *thalach*: Detak jantung maksimum seseorang tercapai
9. *exang*: Angina yang diinduksi oleh olahraga (1 = ya; 0 = tidak)
10. *oldpeak*: ST *depression* yang disebabkan oleh olahraga relatif terhadap istirahat ('ST' berhubungan dengan posisi pada plot EKG)
11. *slope*: kemiringan puncak latihan segmen ST (Nilai 1: miring ke atas, Nilai 2: datar, Nilai 3: miring ke bawah)
12. *ca*: Jumlah kapal besar (0-3)
13. *thal*: Kelainan darah yang disebut *thalasemia* (3 = normal; 6 = cacat tetap; 7 = cacat reversibel)
14. target/*output*: Penyakit jantung (0 = tidak, 1 = ya)

C. *Modeling*

Pada fase ini peneliti melakukan pemilihan teknik *data mining* yaitu metode *Data Mining Decision Tree* dengan Algoritma *Classification*.

D. *Evaluation*

Pada fase evaluasi ini akan dilakukan pengevaluasian pada hasil *modeling* yang sudah diperoleh untuk mengevaluasi apakah hasil yang didapat sudah memenuhi tujuan/harapan. Di tahap ini kami mengevaluasi pemodelan menggunakan *accuracy*, *precision*, dan *recall.*

E. *Deployment*

Pada fase ini, hasil akhir yang didapatkan masih berupa saran untuk diimplementasikan, yaitu bagi para teknisi medis dengan membuat program/*software* untuk mendeteksi serangan jantung.

IV. HASIL DAN PEMBAHASAN

A. *Import Data*

```
#Importing libraries 📚 (Hidden Input/Output)

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
from sklearn.model_selection import GridSearchCV
sns.set_context("poster")

warnings.filterwarnings("ignore")
```

```
# Reading the dataset and having a look at the first 5 rows of the dataframe... (Hidden Input)

df = pd.read_csv("heart.csv")
df.head().style.set_properties(**{"background-color": "#FF87CA","color": "black",
                                  "border-color": "black","font-size":"10pt",'width': 200})
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.300000	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.500000	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.400000	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.800000	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.600000	2	0	2	1

Gambar 3. *Import* data

Gambar di atas merupakan proses *import library* yang akan digunakan, Seperti meng*import numpy*, *pandas*, *matplotlib*, *seaborn*, dan juga *warnings*. Dapat terlihat juga data yang kita pakai yaitu data *heart.csv*.

B. Read Data

```
# Checking the number of rows and columns (Hidden Input)
print("************************************")
print("Number of Rows:", df.shape[0])
print("Number of Columns:", df.shape[1])

# Checking the percentage of null values (Hidden Input)

print("*****Percentage of Null Values*****")
print(round(df.isnull().sum() * 100/df.shape[0]))
print("************************************")
```

```
************************************
Number of Rows: 303
Number of Columns: 14
*****Percentage of Null Values*****
age         0.0
sex         0.0
cp          0.0
trtbps      0.0
chol        0.0
fbs         0.0
restecg     0.0
thalachh    0.0
exng        0.0
oldpeak     0.0
slp         0.0
caa         0.0
thall       0.0
output      0.0
dtype: float64
************************************
```

Gambar 4. Cek data

Pada gambar di atas ini merupakan proses pengecekan data dari *heart.csv* yang akan kita gunakan. pengecekan yang dilakukan adalah mengecek jumlah kolom dan baris. Diketahui data berjumlah 303 baris, 14 kolom, dan tidak memiliki data *null*.

```
# Descriptive Statistics of the numerical colums and percentile to find any potential outliers... (Hidden Input)

df.describe(percentiles=[0.25,0.5,0.75,0.90,0.95,0.98,0.99]).style.set_properties(**{"background-color": "#FF87CA","color": "black",
                                  "border-color": "black","font-size":"10pt",'width': 200})
```

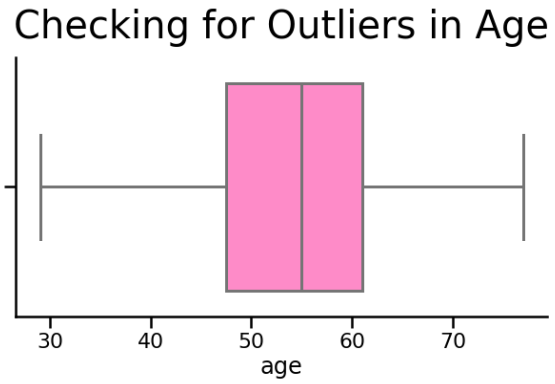
	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
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%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
90%	66.000000	1.000000	2.000000	152.000000	308.800000	1.000000	1.000000	176.600000	1.000000	2.800000	2.000000	2.000000	3.000000	1.000000
95%	68.000000	1.000000	3.000000	160.000000	326.900000	1.000000	1.000000	181.900000	1.000000	3.400000	2.000000	3.000000	3.000000	1.000000
98%	70.000000	1.000000	3.000000	177.840000	353.960000	1.000000	1.000000	186.960000	1.000000	4.000000	2.000000	3.000000	3.000000	1.000000
99%	71.000000	1.000000	3.000000	180.000000	406.740000	1.000000	1.980000	191.960000	1.000000	4.200000	2.000000	4.000000	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 5. Cek outlier

Gambar diatas menunjukan hasil pengecekan *outlier* yang kami lakukan. disini kami mencari variabel yang sekiranya mempunyai *outlier*, dan terlihat hasil yang di dapat pada gambar diatas adalah nilai *cholesterol* yang agak melenceng jauh.

C. EDA

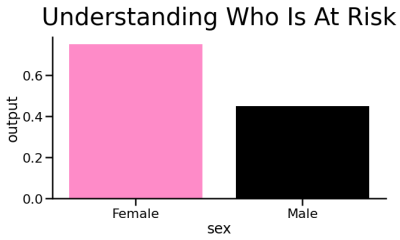
```
# Visualizing the feature: "age" (Hidden Input)

plt.figure(figsize=[10,5])
sns.boxplot(df["age"], color="#FF87CA")
plt.title("Checking for Outliers in Age", size=40, pad=20)
sns.despine()
plt.show()
```



Gambar 6. Cek *outlier* variabel *age*

Pada gambar diatas merupakan hasil dari pengecekan *outlier* pada variabel umur. Terlihat pada gambar di atas, hasil yang kita dapat yaitu tidak ada *outlier* pada variabel umur.

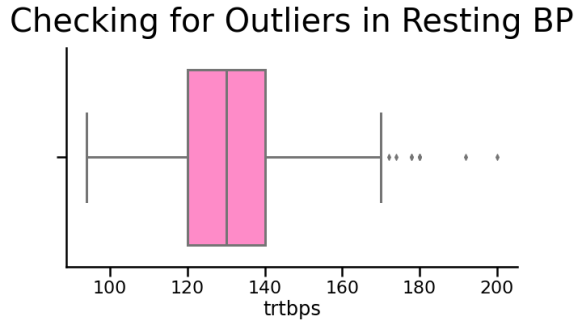
```
# Visualizing the feature: "sex" Assuming 0=Female and 1=Male (No Metadata was provided for gender) (Hidden Input)

plt.figure(figsize=[10,5])
sns.barplot(x = df["sex"], y = df["output"], palette=['#FF87CA','black'], ci=0)
plt.xticks(ticks=[0,1], labels=['Female','Male'])
plt.title("Understanding Who Is At Risk", size=40, pad=20)
sns.despine()
plt.show()
```



Gambar 7. Cek *who is at risk*

Gambar diatas merupakan hasil dari pengecekan siapa yang lebih beresiko terkena serangan jantung berdasarkan jenis kelamin. Terlihat hasil pada gambar diatas menampilkan mayoritas orang yang terkena serangan jantung adalah "*Female*".

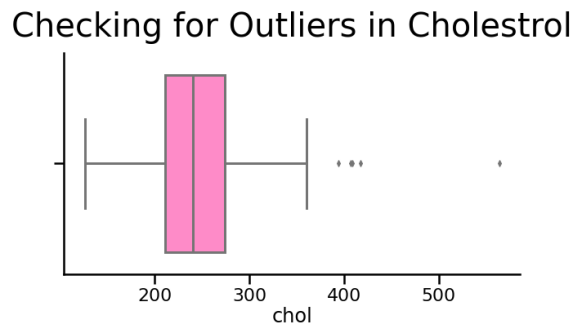
```
# Visualizing the feature: "trtbps" (Hidden Input)

plt.figure(figsize=[10,5])
sns.boxplot(df["trtbps"], color="#FF87CA")
plt.title("Checking for Outliers in Resting BP", size=40, pad=20)
sns.despine()
plt.show()
```



Gambar 8. Cek *outlier* variabel *resting BP*

Pada gambar diatas merupakan hasil dari pengecekan *outlier* pada variabel *resting BP*. hasil yang didapat ternyata terdapat *outlier* pada variabel *resting BP*.

```
# Visualizing the feature: "chol"

plt.figure(figsize=[10,5])
sns.boxplot(df["chol"], color="#FF87CA")
plt.title("Checking for Outliers in Cholestrol", size=40, pad=20)
sns.despine()
plt.show()
```



Gambar 9. Cek *outlier* variabel *cholesterol*

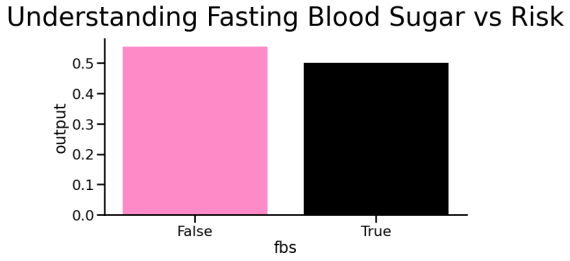
Pada gambar diatas merupakan hasil pengecekan *outlier* pada variabel *Cholesterol*, hasil yang didapat ternyata terdapat *outlier* yang tinggi. Karena nilai yang terlalu tinggi peneliti akan menghapus *outlier* variabel *cholesterol*.

```
# Removing the top 1 percentile...

Q3 = df["chol"].quantile(0.99)
df = df[df["chol"] <= Q3]
```

Gambar 10. *Remove outlier* variabel *cholesterol*

Pada gambar diatas menunjukan *code* untuk menghapus *outlier* variabel *Cholesterol* yang dikarenakan memiliki nilai yang melenceng jauh.

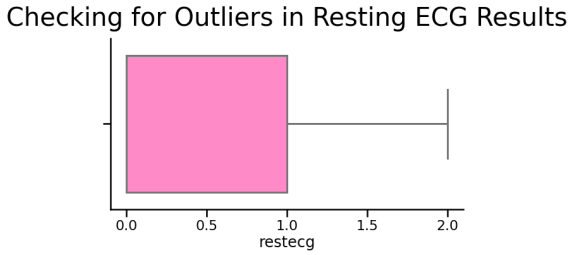
```
# Visualizing the feature: "fbs" (Hidden Input)

plt.figure(figsize=[10,5])
sns.barplot(x = df["fbs"], y = df["output"], palette=['#FF87CA','black'], ci=0)
plt.xticks(ticks=[0,1], labels=['False','True'])
plt.title("Understanding Fasting Blood Sugar vs Risk", size=40, pad=20)
sns.despine()
plt.show()
```



Gambar 11. Cek *fasting blood sugar vs risk*

Pada gambar *understanding fasting blood sugar vs risk* di atas, tidak terlihat adanya perbedaan yang signifikan.

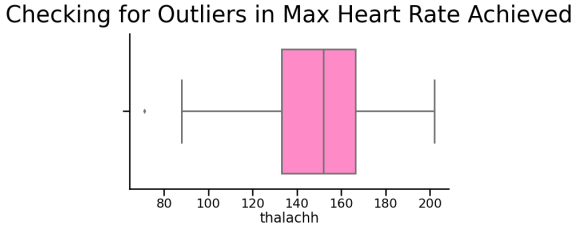
```
# Visualizing the feature: "restecg" (Hidden Input)

plt.figure(figsize=[10,5])
sns.boxplot(df["restecg"], color="#FF87CA")
plt.title("Checking for Outliers in Resting ECG Results", size=40, pad=20)
sns.despine()
plt.show()
```



Gambar 12. Cek *outlier* variabel *resting ECG results*

Gambar diatas merupakan hasil pengecekan *outlier* pada variabel hasil *ECG results*. Pada pengecekan *outlier* pada variabel hasil *ECG results* di ini bisa terlihat jika variabel ini tidak memiliki *outlier*.

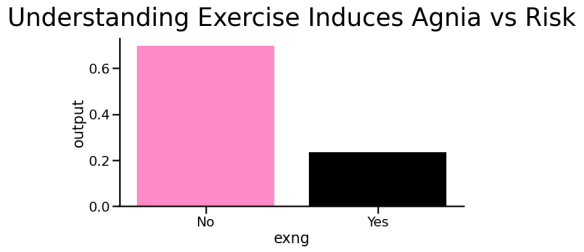
```
# Visualizing the feature: "thalachh" (Hidden Input)

plt.figure(figsize=[10,5])
sns.boxplot(df["thalachh"], color="#FF87CA")
plt.title("Checking for Outliers in Max Heart Rate Achieved", size=40, pad=20)
sns.despine()
plt.show()
```



Gambar 13. Cek *outlier* variabel *max heart race achieved*

Pada gambar diatas merupakan hasil pengecekan *outlier* pada variabel *Max Heart Rate Achieved*, hasil yang didapatkan ternyata terdapat sedikit *outlier*, dan masih dapat diterima.

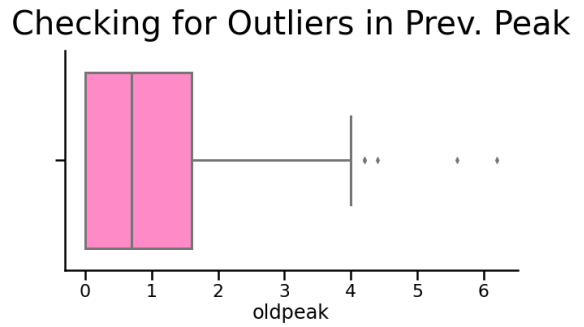
```
# Visualizing the feature: "exng" (Hidden Input)

plt.figure(figsize=[10,5])
sns.barplot(x = df["exng"], y = df["output"], palette=['#FF87CA','black'], ci=0)
plt.xticks(ticks=[0,1], labels=['No','Yes'])
plt.title("Understanding Exercise Induces Agnia vs Risk", size=40, pad=20)
sns.despine()
plt.show()
```



Gambar 14. Cek *exercise induced angina vs risk*

Pada gambar diatas merupakan pengecekan serangan jantung yang disebabkan oleh *Exercise Induced Angina* dan *Risk*. Hasil yang didapat bahwa serangan jantung yang dikarenakan *Exercise Induced Angina* lebih sedikit dibandingkan dengan *risk*.

```
# Visualizing the feature: "oldpeak" (Hidden Input)

plt.figure(figsize=[10,5])
sns.boxplot(df["oldpeak"], color="#FF87CA")
plt.title("Checking for Outliers in Prev. Peak", size=40, pad=20)
sns.despine()
plt.show()
```



Gambar 15. Cek *outlier* variabel *prev. peak*

Pada gambar diatas merupakan hasil pengecekan outlier pada variabel *Prev. Peak*, hasil yang didapat ternyata juga terdapat *outlier* yang tinggi. Karena nilai yang terlalu tinggi peneliti juga akan menghapus outlier variabel *Prev. Peak*.

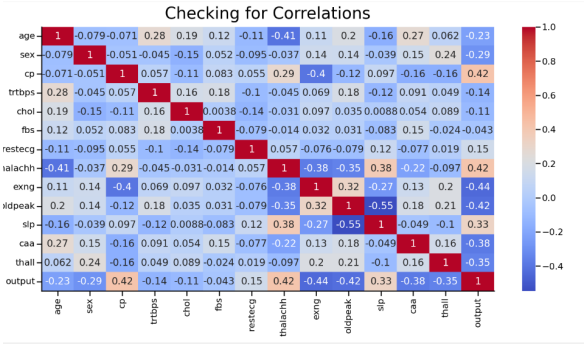
```
# Removing the top 1 percentile...

Q3 = df["oldpeak"].quantile(0.99)
df = df[df["oldpeak"] <= Q3]
```

Gambar 16. *Remove outlier* variabel *prev. peak*

Pada gambar diatas menunjukan *code* untuk menghapus *outlier* variabel *Prev. Peak* yang dikarenakan memiliki nilai yang melenceng jauh.

```
[36] # Create a heatmap

     plt.figure(figsize=[25,12])
     heat = df.corr()
     sns.heatmap(heat, cmap='coolwarm', annot=True)
     plt.title("Checking for Correlations", size=40, pad=20)
     sns.despine()
     plt.show()
```

Gambar 17. Cek *correlation* (1)



Gambar 18. Cek *correlation* (2)

Korelasi positif menggambarkan hubungan antara dua variabel yang saling mempengaruhi dan perubahan yang dialami berjalan searah, sedangkan korelasi terbalik menggambarkan hubungan antara dua variabel yang berubah dengan arah yang berlawanan. Korelasi terbalik kadang-kadang digambarkan sebagai korelasi negatif. Pada gambar korelasi di atas, korelasi negatif kuat memiliki warna biru gelap sedangkan korelasi positif ditunjukan dengan warna orange muda. Hubungan korelasi negatif paling kuat adalah hubungan '*oldpeak*' dan '*slp*' sedangkan hubungan korelasi positif paling kuat antar variabel adalah '*output*' dengan '*thalach*h' dan '*cp*'.

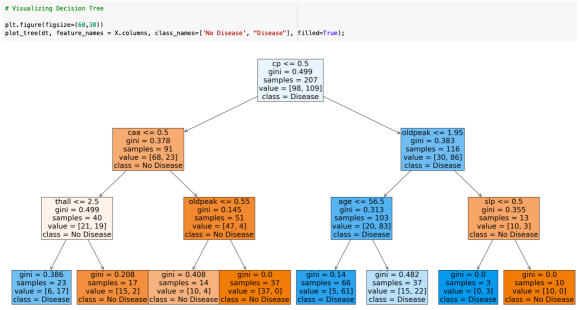
D. Build Model

```
[107] # Seperating the target(y) and the independent(X)
      y = df['output']
      X = df.drop(columns=['output'])

[108] # Performing train_test_split
      X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state = 42)

[109] # Verifying the split
      X_train.shape, y_train.shape
      ((207, 13), (207,))

[110] # Build a random Decision Tree as our first model. Define a max_depth of 3
      dt = DecisionTreeClassifier(max_depth = 3)
      dt.fit(X_train, y_train)
      DecisionTreeClassifier(max_depth=3)
```

Gambar 19. Membuat model

Pembuatan model dilakukan dengan memisahkan variabel target dari set data variabel prediktor (X) ke dalam variabel lain (y). Kemudian, dilakukan pemisahan set data *training* dan set data *testing* dengan ukuran perbandingan 70:30.

Selanjutnya, fungsi pemodelan *decision tree* dipanggil dan diatur kedalaman menjadi 3 sekaligus melakukan *fitting* pada set data *training*. Hasil dari pemodelan akan ditampilkan pada bentuk plot visualisasi.



Gambar 20. Visualisasi model *Decision Tree*

Gambar di atas adalah visualisasi dari hasil pemodelan yang telah dilakukan. Cara membaca keputusan adalah melalui akar dari *decision tree* yang terletak pada kotak paling atas. Di setiap kotak terdapat *feature* yang telah ditentukan oleh mesin sebagai pendukung pembuatan keputusan. Jika nilai keputusan benar maka lanjut ke arah kiri sedangkan jika salah maka akan lanjut ke arah kanan. Kotak yang memiliki *impurity (gini)* paling rendah atau nilai value sudah condong kepada suatu kategori kelas (kelas dengan nilai mayoritas pada kotak) akan menunjukan kelas akhir (*disease, no disease*).

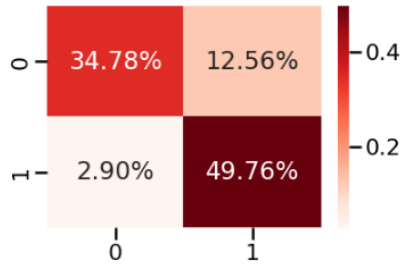
```
# Finding the y_train_pred and the y_test_pred

y_train_pred = dt.predict(X_train)
y_test_pred = dt.predict(X_test)
```

Gambar 21. Membuat variabel *predicted*

Variabel prediksi perlu dibuat untuk melakukan pengujian prediksi dan evaluasi pada hasil prediksi model.

```
# Evaluating the model: (Confusion Matrix)

print("Confusion Matrix for training set:\n")
print(confusion_matrix(y_train, y_train_pred))
print('*'*20)
cm2 = confusion_matrix(y_train, y_train_pred)
sns.heatmap(cm2/np.sum(cm2), annot = True, fmt=  '0.2%', cmap = 'Reds')
```

```
Confusion Matrix for training set:

[[ 72  26]
 [  6 103]]
********************
<matplotlib.axes._subplots.AxesSubplot at 0x7f1ecf025390>
```



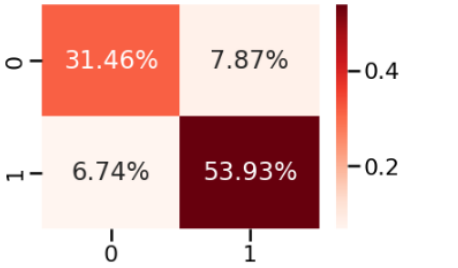
Gambar 22. Evaluasi set train

Pada gambar di atas, peneliti membuat variabel predicted untuk data *training* dan *testing*. Lalu ditampilkan dalam bentuk *Confusion Matrix* untuk dievaluasi.

Di sini, kesalahan Tipe-1 adalah 12.56% yang juga dikenal sebagai Positif Palsu atau *False Positive*. Kesalahan Tipe-2 adalah 2.90% yang juga dikenal sebagai Negatif Palsu atau *False Negative*. Sementara nilai % lainnya dalam matriks konfusi menunjukkan bahwa mereka diprediksi dengan benar dalam kategori spesifiknya.

```
# Evaluating the model: (Confusion Matrix)

print("Confusion Matrix for test set:\n")
print(confusion_matrix(y_test, y_test_pred))
print('*'*20)
cm2 = confusion_matrix(y_test, y_test_pred)
sns.heatmap(cm2/np.sum(cm2), annot = True, fmt=  '0.2%', cmap = 'Reds')
```

```
Confusion Matrix for test set:

[[28  7]
 [ 6 48]]
********************
<matplotlib.axes._subplots.AxesSubplot at 0x7f1ecef3ce50>
```



Gambar 22. Evaluasi set test

Di sini, set data testing memiliki kesalahan Tipe-1 sebesar 7,87% yang juga dikenal sebagai Positif Palsu atau *False Positive*. Kesalahan Tipe-2 adalah 6,74% yang juga dikenal sebagai Negatif Palsu atau *False Negative*. Sementara nilai % lainnya dalam matriks konfusi menunjukkan bahwa mereka diprediksi dengan benar dalam kategori spesifiknya.

```
# Evaluating the model: (Accuracy)

print("Accuracy on the training set: " + str(accuracy_score(y_train, y_train_pred)))
print('*'*20)
print("Accuracy on the test set: " + str(accuracy_score(y_test, y_test_pred)))
```

```
Accuracy on the training set: 0.8454106280193237
********************
Accuracy on the test set: 0.8539325842696629
```

```
# Evaluating the model: (Precision)

print("Precision on the training set: " + str(precision_score(y_train, y_train_pred)))
print('*'*20)
print("Precision on the test set: " + str(precision_score(y_test, y_test_pred)))
```

```
Precision on the training set: 0.7984496124031008
********************
Precision on the test set: 0.8727272727272727
```

```
# Evaluating the model: (Recall)

print("Recall on the training set: " + str(recall_score(y_train, y_train_pred)))
print('*'*20)
print("Recall on the test set: " + str(recall_score(y_test, y_test_pred)))
```

```
Recall on the training set: 0.944954128440367
********************
Recall on the test set: 0.8888888888888888
```

Gambar 23. Evaluasi *accuracy*, *precision*, dan *recall*

Accuracy: rasio pasien yang diklasifikasikan dengan benar (TP+TN) dengan jumlah total pasien (TP+TN+FP+FN).

Precision: rasio pasien dengan penyakit yang diklasifikasikan dengan benar (TP) dibagi dengan total pasien yang diprediksi menderita penyakit (TP+FP).

Recall: rasio pasien sakit yang diklasifikasikan dengan benar (TP) dibagi dengan jumlah pasien yang benar-benar menderita penyakit tersebut (TP+FN).

Dari ketiga jenis akurasi tersebut dapat dilihat dan disimpulkan bahwa semua nilainya hampir satu yang artinya bahwa model ini dapat dikatakan sudah bagus.

V. KESIMPULAN

Peneliti membuat pemodelan *Decision Tree* untuk melihat kemungkinan seseorang mengalami serangan jantung. Peneliti dapat menyimpulkan bahwa pemodelan yang telah dilakukan sudah baik. Lalu, peneliti menemukan beberapa kesimpulan, yaitu:

1. Kemungkinan besar seseorang mengalami serangan jantung, tanpa memandang jenis kelamin, jika tingkat *Chest Pain*-nya 1 atau lebih.
2. Subjek *Female* lebih berisiko mengalami serangan jantung daripada *Male*.
3. Jika mengalami *Heart Rate* yang tinggi dan *Chest Pain* minimal level 1, maka subjek pasti akan terkena serangan jantung.
4. Jika tingkat *Chest Pain* di bawah 1, tetapi memiliki *Exercise Induced Angina*, ia berisiko lebih besar terkena serangan jantung.

Oleh karena itu, jika salah satu dari kondisi di atas diamati pada seseorang, mereka harus segera diberikan penanganan medis.

DAFTAR PUSTAKA

[1] Alimansur M., & Irawan H. (2017). PENGARUH PENINGKATAN KADAR KOLESTEROL DAN GLUKOSA DARAH TERHADAP PULSE PRESSURE PENDERITA HIPERTENSI. *Jurnal Keperawatan*.

[2] Jijo, B. T., & Abdulazeez, A. M. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(1), 20-28.

[3] Jindal, H., Agrawal, S., Khera, R., Jain, R., & Nagrath, P. (2021). Heart disease prediction using machine learning algorithms. *OP Conference Series: Materials Science and Engineering*, *1022*(1).

[4] Pal, M., & Parija, S. (2021, March). Prediction of heart diseases using random forest. *Journal of Physics: Conference Series*, *1817*(1). 10.1088/1742-6596/1817/1/012009

[5] Rachmawati C., Martini S., & D, A. K. (2021). ANALISIS FAKTOR RISIKO MODIFIKASI PENYAKIT JANTUNG KORONER DI RSU HAJI SURABAYA TAHUN 2019. *Media Gizi Kesmas*, 47-55.

[6] Wang, M., Yao, X., & Chen, Y. (2021). An imbalanced-data processing algorithm for the prediction of heart attack in stroke patients. *IEEE Access*, *9*, 25394-2540.