

LAPORAN TUGAS BESAR

PRAKTIKUM KECERDASAN BUATAN

**“PREDIKSI PENYAKIT JANTUNG MENGGUNAKAN METODE DECISION
TREE”**

Dibuat untuk memenuhi tugas yang di ampu oleh :

Leni Fitriani ST.M.kom



Disusun oleh :

Danilh Jian H (2106171)

M Bambang I A (2106106)

JURUSAN ILMU KOMPUTER

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI GARUT

2023

DAFTAR ISI

DAFTAR ISI.....	2
BAB I PENDAHULUAN.....	3
2.1. Latar belakang	3
2.2. Tujuan.....	3
2.3. Manfaat.....	5
BAB II HASIL DAN PEMBAHASAN.....	7
2.1. langkah-langkah program prediksi penyakit jantung	7
2.2. Pengumpulan Dataset	8
2.3. Visualisasi	9
2.4. Mengambil Mean dari Target.....	9
2.5. Scaling.....	10
2.6. Heatmap	12
2.7. Modeling	13
2.8. Predicting	13
BAB III KESIMPULAN & SARAN	15
DAFTAR PUSTAKA	16

BAB I

PENDAHULUAN

2.1. Latar belakang

Penyakit jantung merupakan salah satu penyakit yang menjadi masalah kesehatan masyarakat di seluruh dunia. Penyakit ini dapat menyebabkan komplikasi serius dan bahkan kematian jika tidak terdeteksi dan ditangani dengan tepat. Oleh karena itu, penting untuk mengembangkan metode prediksi yang efektif dalam mengidentifikasi risiko penyakit jantung pada individu. Metode prediksi menggunakan kecerdasan buatan, seperti algoritma Decision Tree, telah menunjukkan hasil yang menjanjikan dalam berbagai bidang, termasuk di bidang kesehatan. Algoritma Decision Tree memanfaatkan teorema untuk menghitung probabilitas suatu kejadian berdasarkan kemunculan gejala atau faktor risiko terkait. Dengan mengatasi kendala-kendala tersebut, diharapkan metode prediksi penyakit jantung menggunakan algoritma Decision Tree dapat memberikan kontribusi yang signifikan dalam mendukung diagnosis dini, pengobatan yang tepat, dan pencegahan lebih lanjut terhadap penyakit jantung.

Data mining mengacu pada ekstraksi informasi yang diperlukan dari kumpulan data besar di berbagai bidang seperti bidang medis, bidang bisnis, dan bidang pendidikan. Algoritma ini dapat menganalisis data yang sangat besar dari berbagai bidang, salah satu bidang penting tersebut adalah bidang medis. Data mining mengeksplorasi kumpulan data besar untuk mengekstrak informasi pengambilan keputusan penting yang tersembunyi dari kumpulan repositori masa lalu untuk analisis di masa mendatang. Bidang medis terdiri dari data pasien yang luar biasa. Berbagai teknik data mining seperti regresi, pengelompokan, aturan asosiasi dan teknik klasifikasi yang digunakan sebagai penelitian pada paper ini untuk mengklasifikasikan berbagai atribut penyakit jantung dalam memprediksi penyakit jantung.

2.2. Tujuan

Tujuan prediksi penyakit jantung menggunakan metode Decision Tree adalah untuk memberikan perkiraan atau prediksi tentang kemungkinan seseorang menderita penyakit jantung berdasarkan

faktor-faktor risiko dan gejala yang diamati. Berikut adalah beberapa tujuan utama dari penggunaan metode Decision Tree dalam prediksi penyakit jantung:

1. Deteksi dini: Salah satu tujuan utama adalah mendeteksi penyakit jantung pada tahap dini. Dengan menggunakan metode Decision Tree, faktor risiko seperti riwayat keluarga, kebiasaan hidup, dan gejala yang diamati dapat digunakan untuk memperkirakan risiko seseorang terkena penyakit jantung. Hal ini memungkinkan tindakan pencegahan atau pengobatan yang lebih awal untuk dilakukan, yang dapat membantu mengurangi dampak dan risiko komplikasi yang mungkin terjadi.
2. Pengklasifikasi pasien: Metode Decision Tree dapat digunakan untuk mengklasifikasikan pasien ke dalam kelompok risiko yang berbeda berdasarkan probabilitas mereka menderita penyakit jantung. Hal ini membantu dalam mengidentifikasi pasien dengan risiko tinggi yang memerlukan perhatian dan pengawasan lebih intensif, serta pasien dengan risiko rendah yang mungkin hanya memerlukan tindak lanjut yang terbatas.
3. Pendukung pengambilan keputusan: Prediksi penyakit jantung dengan metode Decision Tree memberikan informasi yang berharga bagi tenaga medis dalam pengambilan keputusan. Dalam kombinasi dengan pengetahuan medis dan evaluasi yang lebih mendalam, prediksi ini dapat digunakan sebagai alat bantu dalam merencanakan strategi pengobatan yang tepat, melakukan tindakan pencegahan, atau mengarahkan pasien ke tes diagnostik yang lebih lanjut.
4. Pemantauan dan evaluasi: Metode Decision Tree juga dapat digunakan untuk memantau dan mengevaluasi respons pasien terhadap pengobatan atau intervensi tertentu. Dengan melacak perubahan probabilitas prediksi seiring waktu, tenaga medis dapat menilai efektivitas perawatan yang diberikan dan membuat penyesuaian yang diperlukan.
5. Riset dan pemahaman penyakit jantung: Penggunaan metode Decision Tree dalam prediksi penyakit jantung juga dapat memberikan wawasan baru dan pemahaman lebih mendalam tentang hubungan antara faktor risiko, gejala, dan penyakit jantung itu sendiri. Analisis yang dilakukan menggunakan metode ini dapat membantu mengidentifikasi faktor risiko yang paling berpengaruh dan mengungkapkan pola hubungan yang berguna dalam konteks riset dan pemahaman penyakit jantung.

Penting untuk dicatat bahwa prediksi menggunakan metode Decision Tree bukanlah diagnosis yang pasti, tetapi merupakan alat bantu yang dapat memberikan indikasi atau estimasi probabilitas penyakit jantung. Keputusan medis akhir harus didasarkan pada evaluasi menyeluruh oleh tenaga medis yang berkualifikasi dan menggunakan metode diagnostik yang lebih lanjut.

2.3. Manfaat

Prediksi penyakit jantung menggunakan metode Decision Tree memiliki beberapa manfaat, antara lain:

1. Identifikasi dini penyakit jantung: Metode Decision Tree dapat membantu dalam mengidentifikasi dini kemungkinan adanya penyakit jantung pada seseorang. Dengan menggunakan data yang relevan seperti riwayat medis, faktor risiko, dan gejala yang terkait, metode ini dapat memberikan prediksi kemungkinan seseorang menderita penyakit jantung.
2. Efisiensi dan kecepatan: Metode Decision Tree merupakan metode yang relatif sederhana dan komputasionalnya efisien. Algoritma ini dapat dengan cepat melakukan prediksi penyakit jantung berdasarkan data yang diberikan. Hal ini memungkinkan deteksi dini penyakit jantung dengan waktu yang singkat.
3. Manfaat dalam pengambilan keputusan: Prediksi penyakit jantung menggunakan metode Decision Tree dapat memberikan informasi yang berharga bagi tenaga medis dalam mengambil keputusan terkait pengobatan dan langkah-langkah pencegahan yang tepat. Dengan memiliki prediksi yang akurat, langkah-langkah yang diperlukan dapat segera diambil untuk mencegah atau mengobati penyakit jantung.
4. Pendekatan probabilistik: Metode Decision Tree menggabungkan pemikiran probabilistik dalam proses prediksi. Hal ini memungkinkan penilaian kemungkinan penyakit jantung berdasarkan probabilitas yang terkait dengan faktor risiko dan gejala yang diamati. Pendekatan ini dapat membantu dalam menghasilkan prediksi yang lebih akurat.
5. Mudah digunakan dan diimplementasikan: Algoritma Decision Tree relatif mudah untuk dipahami dan diimplementasikan. Dalam konteks prediksi penyakit jantung, metode ini

dapat diterapkan dengan cepat tanpa memerlukan persyaratan yang kompleks. Ini menjadikannya sebagai metode yang dapat diadopsi dengan mudah oleh profesional medis.

BAB II

HASIL DAN PEMBAHASAN

2.1. langkah-langkah program prediksi penyakit jantung

1. Pengumpulan Data: Kumpulkan dataset yang berisi informasi pasien yang relevan, seperti usia, jenis kelamin, tekanan darah, kolesterol, gula darah, riwayat merokok, riwayat diabetes, dan lain-lain. Pastikan dataset tersebut mencakup pasien dengan dan tanpa penyakit jantung.
2. Preprocessing Data: Lakukan tahap preprocessing data untuk membersihkan dan mengubah format data agar sesuai dengan persyaratan algoritma Decision Tree. Hal ini mungkin melibatkan penghilangan missing values, normalisasi data, konversi variabel kategorikal menjadi numerik, dan lain-lain.
3. Pembagian Data: Bagi dataset menjadi dua subset, yaitu data latih (training data) dan data uji (testing data). Biasanya, sekitar 70-80% data digunakan untuk training, sementara sisanya digunakan untuk pengujian.
4. Pelatihan Model: Gunakan data latih untuk melatih model Decision Tree. Algoritma Decision Tree akan mempelajari hubungan antara fitur-fitur yang ada dan label kelas (penyakit jantung atau tidak). Model akan mengestimasi probabilitas kondisional untuk setiap fitur, berdasarkan label kelas.
5. Pengujian Model: Gunakan data uji yang telah dipisahkan untuk menguji model yang telah dilatih. Masukkan fitur-fitur dari data uji ke dalam model dan amati prediksi yang dihasilkan. Bandingkan prediksi dengan label kelas yang sebenarnya untuk mengukur akurasi model.
6. Evaluasi Model: Evaluasi performa model dengan menggunakan metrik evaluasi yang sesuai, seperti akurasi, presisi, recall, atau F1-score. Analisis juga dapat dilakukan untuk mengetahui faktor-faktor yang paling berpengaruh dalam prediksi penyakit jantung.
7. Penyempurnaan Model: Jika performa model belum memuaskan, beberapa langkah penyempurnaan dapat dilakukan. Misalnya, menggunakan teknik seleksi fitur untuk memilih fitur-fitur yang paling informatif, melakukan tuning parameter, atau mencoba metode Decision Tree yang lebih canggih seperti Decision Tree Berstruktur atau Decision Tree .

8. Implementasi dan Penggunaan: Setelah model telah memberikan hasil yang memadai, model dapat diimplementasikan dalam aplikasi atau sistem yang lebih besar untuk membantu dalam prediksi penyakit jantung pada pasien baru. Pastikan juga untuk mengikuti standar keamanan dan privasi data yang berlaku.

Penelitian ini menggunakan metode eksperimen. Dimana peneliti melakukan percobaan menggunakan dataset yang disediakan oleh UCI Machine learning. Dataset ini digunakan sebagai data training dengan menggunakan pemrograman bahasa Python dan menggunakan library Sklearn untuk pengklasifikasian target berupa diagnosis ada atau tidaknya penyakit jantung menggunakan metode Decision Tree. Peneliti menguji keakuratan dari sistem prediksi yang dibangun dengan menghitung persentase benar atau salah dari hasil prediksi sistem.

2.2. Pengumpulan Dataset

▼ Membaca Dataset

✓
0 d

▶

```
df = pd.read_csv('heart.csv')
df.head()
```

📄

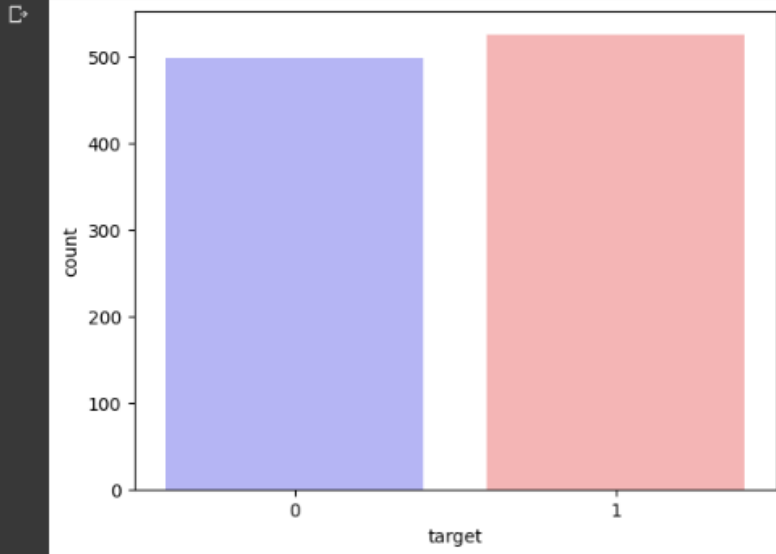
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

✎

2.3. Visualisasi

Visualisasi

```
df.target.value_counts()  
sns.countplot(x="target", data=df, palette="bwr")  
plt.show()
```



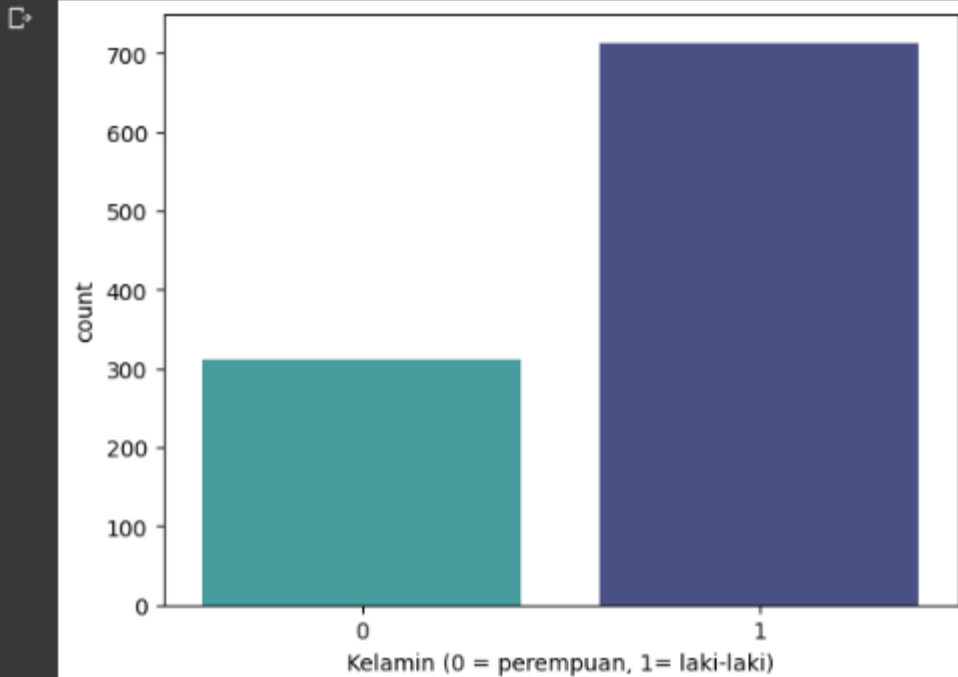
target: memiliki penyakit jantung atau tidak (1 = ya, 0 = tidak)

```
[6] countNoDisease = len(df[df.target == 0])  
countHaveDisease = len(df[df.target == 1])  
print("Persentase pasien tidak memiliki penyakit jantung: {:.2f}%".format((countNoDisease / (len(df.target))*100))  
print("Persentase pasien memiliki penyakit jantung: {:.2f}%".format((countHaveDisease / (len(df.target))*100))
```

```
Persentase pasien tidak memiliki penyakit jantung: 48.68%  
Persentase pasien memiliki penyakit jantung: 51.32%
```

Persentase pasien memiliki penyakit jantung: 51.32%

```
0d sns.countplot(x='sex', data=df, palette="mako_r")
plt.xlabel("Kelamin (0 = perempuan, 1= laki-laki)")
plt.show()
```



```
0d [8] countFemale = len(df[df.sex == 0])
countMale = len(df[df.sex == 1])
print("Persentase pasien perempuan: {:.2f}%".format((countFemale / (len(df.sex))*100))
print("Persentase pasien laki-laki: {:.2f}%".format((countMale / (len(df.sex))*100))
```

Persentase pasien perempuan: 30.44%
Persentase pasien laki-laki: 69.56%

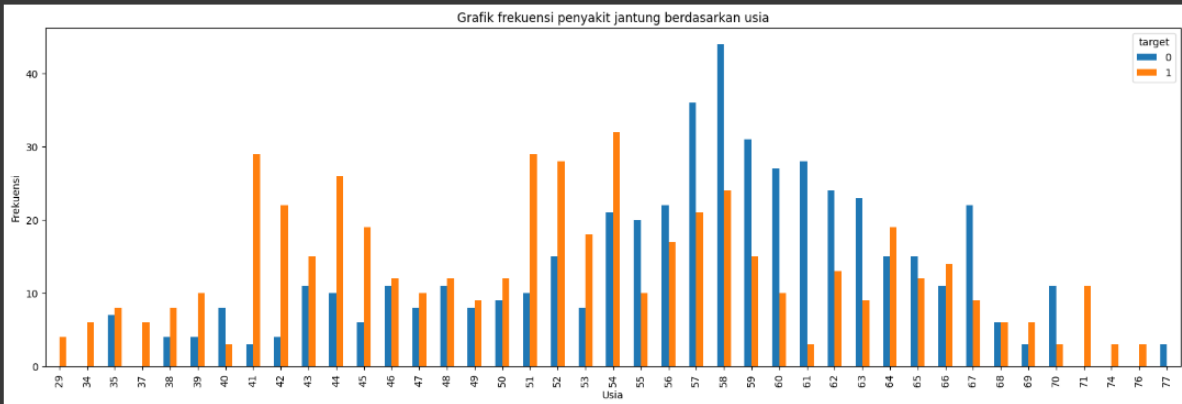
2.4. Mengambil Mean Dari Target

▼ Mengambil mean dari target

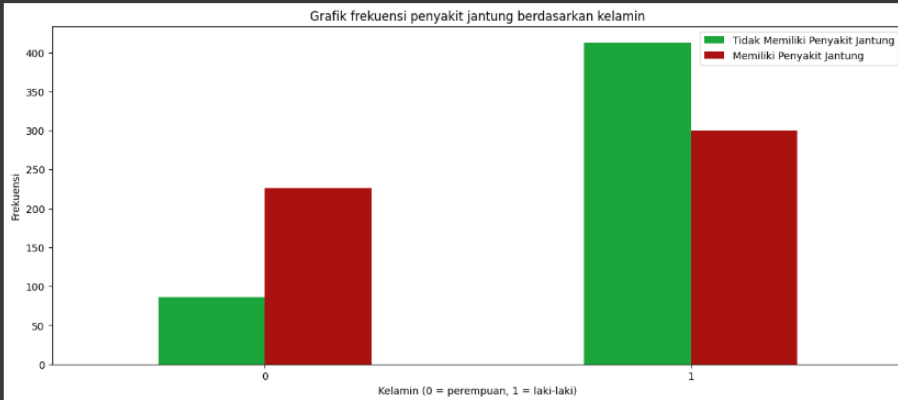
```
0d [9] df.groupby('target').mean()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
target													
0	56.569138	0.827655	0.482966	134.106212	251.292585	0.164329	0.456914	139.130261	0.549098	1.600200	1.166333	1.158317	2.539078
1	52.408745	0.570342	1.378327	129.245247	240.979087	0.134981	0.598859	158.585551	0.134981	0.569962	1.593156	0.370722	2.119772

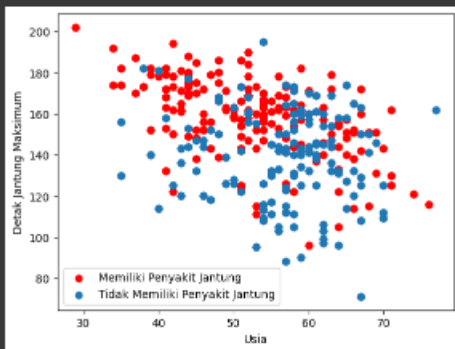
```
0d pd.crosstab(df.age,df.target).plot(kind="bar",figsize=(20,6))
plt.title('Grafik frekuensi penyakit jantung berdasarkan usia')
plt.xlabel('Usia')
plt.ylabel('Frekuensi')
plt.savefig('PenyakitJantungDanUmur.png')
plt.show()
```



```
[11] pd.crosstab(df.sex,df.target).plot(kind="bar",figure=(15,6),color=['#1CA53B','#AA1111' ])
plt.title('Grafik frekuensi penyakit jantung berdasarkan kelamin')
plt.xlabel('Kelamin (0 = perempuan, 1 = laki-laki)')
plt.xticks(rotation=0)
plt.legend(['Tidak Memiliki Penyakit Jantung', 'Memiliki Penyakit Jantung'])
plt.ylabel('Frekuensi')
plt.show()
```

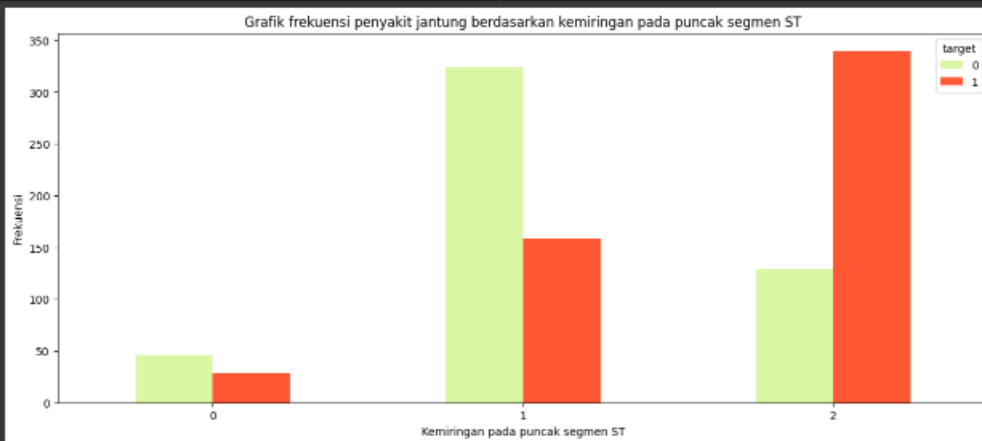


```
[12] plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c="red")
plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)])
plt.legend(["Memiliki Penyakit Jantung", "Tidak Memiliki Penyakit Jantung"])
plt.xlabel("Usia")
plt.ylabel("Detak Jantung Maksimum")
plt.show()
```



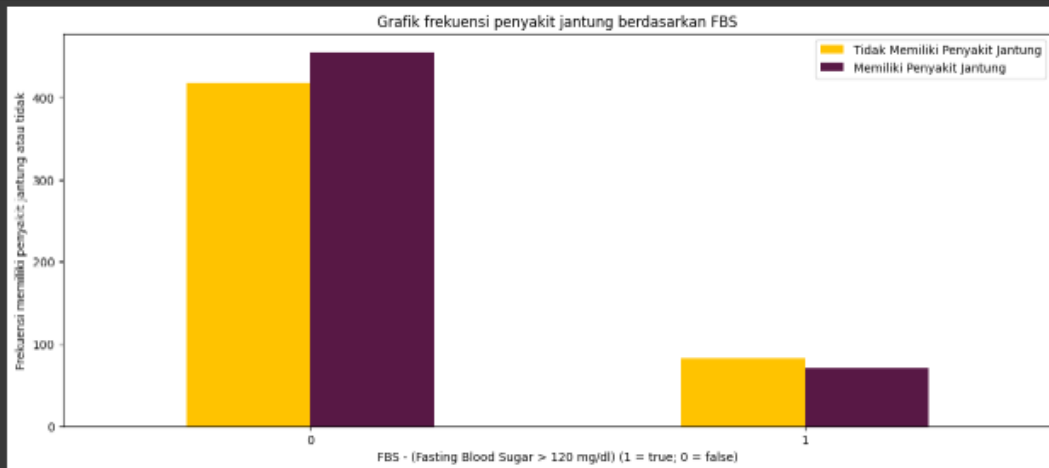
▼ thalach - Detak jantung maksimum yang dicapai

```
[13] pd.crosstab(df.slope, df.target).plot(kind="bar", figsize=(15,6), color=["#DAF7A6", "#FF5733"])
plt.title("Grafik frekuensi penyakit jantung berdasarkan kemiringan pada puncak segmen ST")
plt.xlabel("Kemiringan pada puncak segmen ST")
plt.xticks(rotation = 0)
plt.ylabel("Frekuensi")
plt.show()
```



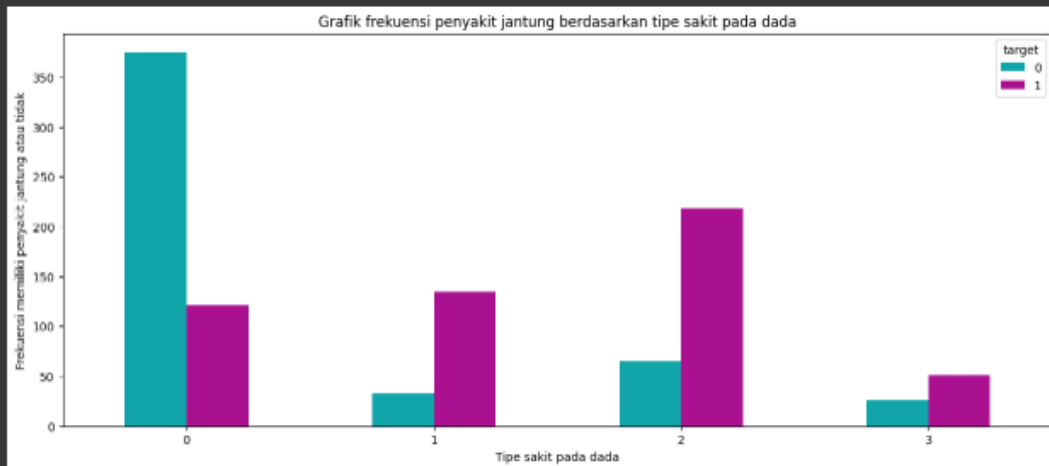
```
[14] pd.crosstab(df.fbs, df.target).plot(kind="bar", figsize=(15,6), color=["#FFC300", "#581845"])
plt.title("Grafik frekuensi penyakit jantung berdasarkan FBS")
plt.xlabel("FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)")
plt.xticks(rotation = 0)
plt.legend(["Tidak Memiliki Penyakit Jantung", "Memiliki Penyakit Jantung"])
plt.ylabel("Frekuensi memiliki penyakit jantung atau tidak")
plt.show()
```

```
[14] pd.crosstab(df.fbs,df.target).plot(kind="bar",figsize=(15,6),color=['#FFC300','#581845' ])
plt.title('Grafik frekuensi penyakit jantung berdasarkan FBS')
plt.xlabel('FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')
plt.xticks(rotation = 0)
plt.legend(['Tidak Memiliki Penyakit Jantung', 'Memiliki Penyakit Jantung'])
plt.ylabel('Frekuensi memiliki penyakit jantung atau tidak')
plt.show()
```



▼ fbs - (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

```
[15] pd.crosstab(df.cp,df.target).plot(kind="bar",figsize=(15,8),color=['#11A5AA','#AA1190' ])
plt.title('Grafik frekuensi penyakit jantung berdasarkan tipe sakit pada dada')
plt.xlabel('Tipe sakit pada dada')
plt.xticks(rotation = 0)
plt.ylabel('Frekuensi memiliki penyakit jantung atau tidak')
plt.show()
```



cp - tipe sakit pada dada

2.5. Scaling

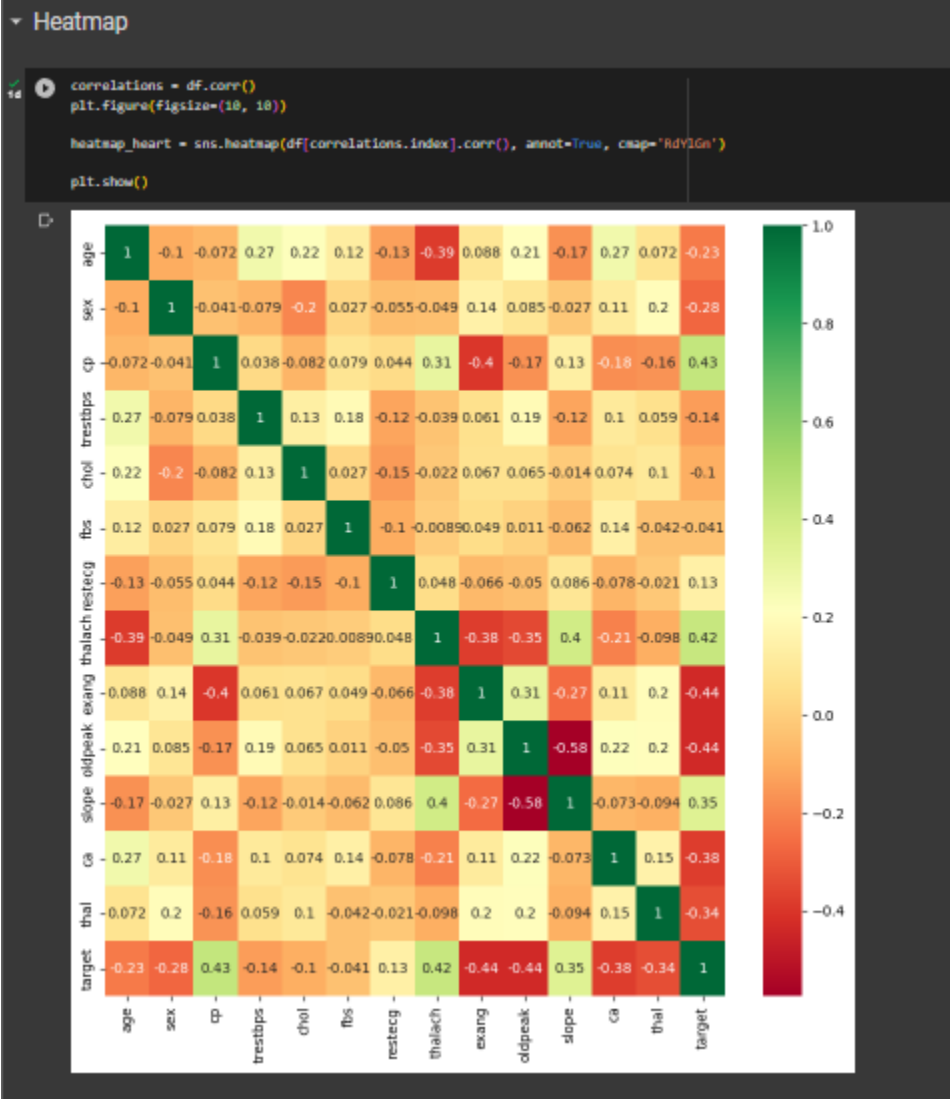
▼ Scaling

```
[27] scaler = MinMaxScaler()
normalized_array = scaler.fit_transform(df[['sex','chol','fbs','restecg','exang','slope','ca','thal']])
normalized_df = pd.DataFrame(normalized_array,columns=['sex','chol','fbs','restecg','exang','slope','ca','thal'])
df[['sex','chol','fbs','restecg','exang','slope','ca','thal']] = normalized_df
```

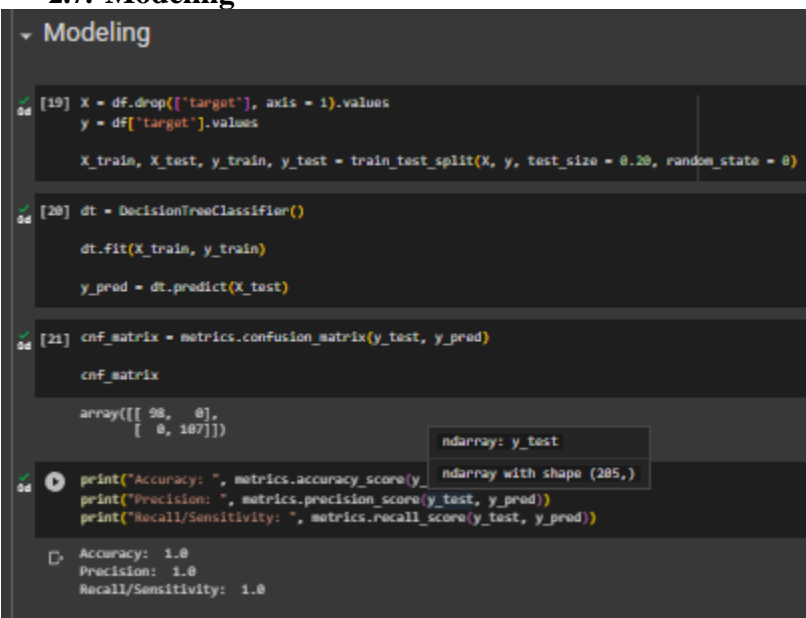
```
[28] df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1.0	0	125	0.196347	0.0	0.5	168	0.0	1.0	1.0	0.50	1.000000	0
1	53	1.0	0	140	0.175799	1.0	0.0	155	1.0	3.1	0.0	0.00	1.000000	0
2	70	1.0	0	145	0.109589	0.0	0.5	125	1.0	2.6	0.0	0.00	1.000000	0
3	61	1.0	0	148	0.175799	0.0	0.5	161	0.0	0.0	1.0	0.25	1.000000	0
4	62	0.0	0	138	0.383562	1.0	0.5	106	0.0	1.9	0.5	0.75	0.666667	0

2.6. Heatmap



2.7. Modeling



2.8. Predicting

```

Predicting

[23] smote = SMOTE(random_state = 45)

s_X_train, s_y_train = smote.fit_resample(X_train, y_train)
s_X_test, s_y_test = smote.fit_resample(X_test, y_test)

[24] s_dt = DecisionTreeClassifier()

s_dt.fit(s_X_train, s_y_train)

s_y_pred = dt.predict(s_X_test)

[25] s_cnf_matrix = metrics.confusion_matrix(s_y_test, s_y_pred)

s_cnf_matrix

array([[187,  0],
       [ 0, 187]])

Data Akan berubah setiap run

• 22 pasien diprediksi bahwa mereka akan memiliki penyakit jantung, prediksinya benar (True-Positive)
• 24 pasien diprediksi bahwa mereka tidak akan memiliki penyakit jantung, prediksinya benar (True-Negative)
• 5 pasien diprediksi bahwa mereka akan memiliki penyakit jantung, namun prediksinya salah (False-Positive)
• 10 pasien diprediksi bahwa mereka tidak akan memiliki penyakit jantung, namun hasil prediksinya salah (False-Negative)

[26] print("Accuracy: ", metrics.accuracy_score(s_y_test, s_y_pred))
print("Precision: ", metrics.precision_score(s_y_test, s_y_pred))
print("Recall/Sensitivity: ", metrics.recall_score(s_y_test, s_y_pred))

Accuracy: 1.0
Precision: 1.0
Recall/Sensitivity: 1.0
```

BAB III

KESIMPULAN & SARAN

Model yang dibuat menggunakan metode Decision Tree mampu mendiagnosa apakah seseorang mengidap penyakit jantung. Data yang 22 pasien diprediksi bahwa mereka akan memiliki penyakit jantung, prediksinya benar (True-Positive), 24 pasien diprediksi bahwa mereka tidak akan memiliki penyakit jantung, prediksinya benar (True-Negative), 5 pasien diprediksi bahwa mereka akan memiliki penyakit jantung, namun prediksinya salah (False-Positive), dan 10 pasien diprediksi bahwa mereka tidak akan memiliki penyakit jantung, namun hasil prediksinya salah (False-Negative).

DAFTAR PUSTAKA

<https://colab.research.google.com/drive/1DlGCHjPSaWqCY-BNFif22BljVLW6Docn?usp=sharing>
<https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>