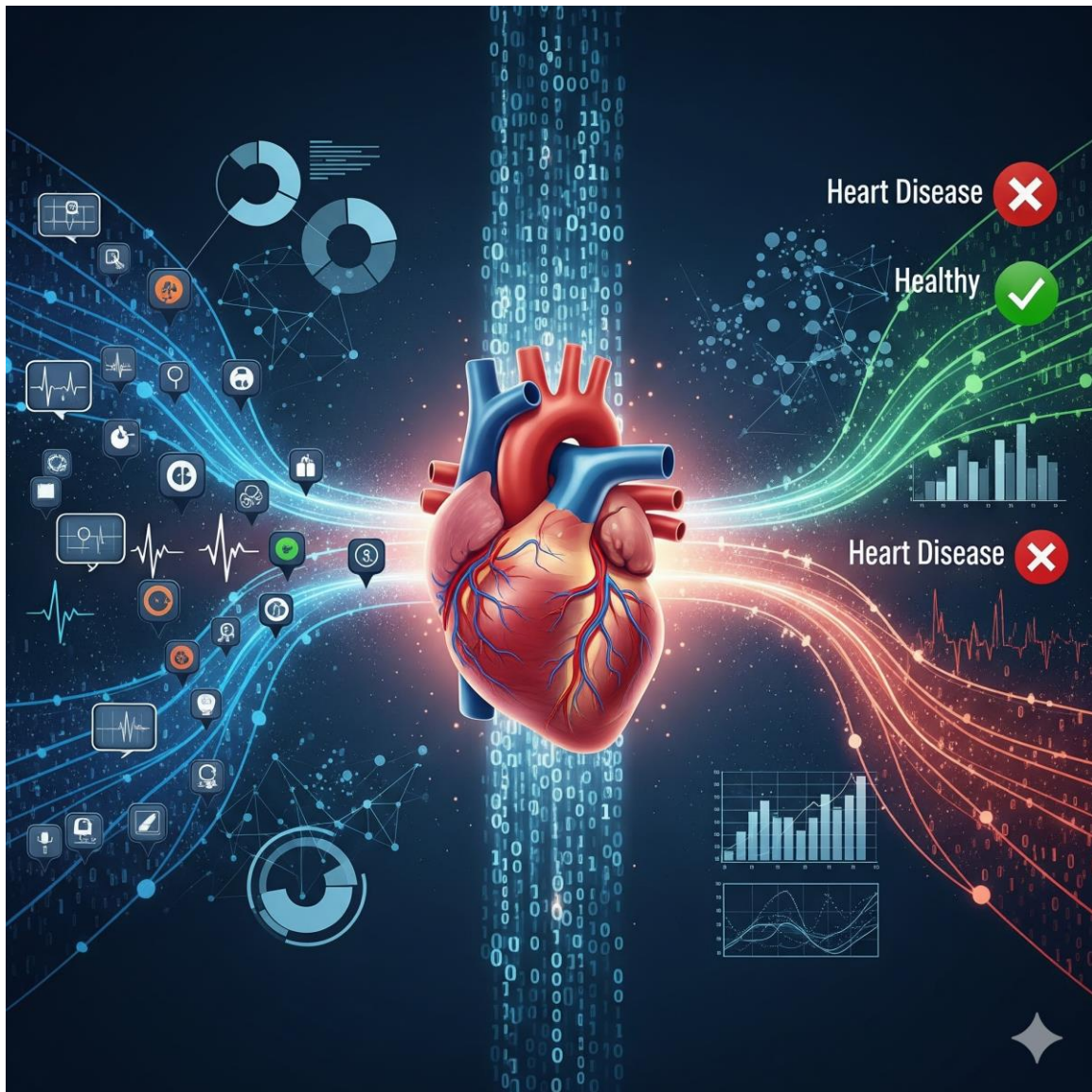


# PORTOFOLIO APLIKASI KLASIFIKASI BINER PENYAKIT JANTUNG

Proyek: Aplikasi Klasifikasi Penyakit Jantung Berbasis Streamlit

Bahasa Pemrograman: Python

Framework: Streamlit



## 1. Deskripsi Proyek

Aplikasi Klasifikasi Penyakit Jantung ini dikembangkan menggunakan Streamlit dan Python. Tujuan utama aplikasi adalah membantu pengguna atau tenaga medis untuk memprediksi kemungkinan terjadinya penyakit jantung berdasarkan data pasien seperti usia, tekanan darah, kolesterol, dan detak jantung. Model machine learning yang digunakan antara lain Random Forest dan algoritma lain yang dibandingkan untuk menentukan akurasi terbaik.

## 2. Struktur Folder dan File

Berikut struktur folder dari proyek:

- extracted\_project/
  - App\_Klasifikasi\_Penyakit\_Jantung/
    - Hasil\_Prediksi\_Random\_Forest.xlsx
    - heart.csv
    - heart.xlsx
    - Klasifikasi\_Penyakit\_Jantung.py
    - requirements.txt
    - utils.py
  - assets/
    - logo.png
  - pages/
    - 1\_Upload Dataset.py
    - 2\_Analisis Algoritma.py
    - 3\_Input Data User.py
    - 4\_Informasi Dataset.py
  - \_\_pycache\_\_/
    - utils.cpython-38.pyc

## 3. Kebutuhan Sistem (Requirements)

Untuk menjalankan aplikasi ini, diperlukan:

- Python 3.11
- Streamlit
- Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
- File requirements.txt berisi daftar dependensi lengkap

## 4. Langkah Instalasi & Menjalankan Aplikasi

1. Ekstrak file ZIP proyek ke dalam direktori pilihan Anda.

2. Buka terminal atau command prompt, lalu masuk ke folder proyek:

```
cd App_Klasifikasi_Penyakit_Jantung
```

3. Instal semua dependensi menggunakan perintah:

```
pip install -r requirements.txt
```

4. Jalankan aplikasi Streamlit dengan perintah:

```
streamlit run Klasifikasi_Penyakit_Jantung.py
```

5. Aplikasi akan terbuka di browser default, biasanya di alamat <http://localhost:8501>

## 5. Penjelasan Modul dan File Utama

- `Klasifikasi_Penyakit_Jantung.py`: File utama Streamlit yang menjalankan aplikasi, menampilkan halaman utama, dan memanggil fungsi dari `utils.py`.
- `utils.py`: Berisi fungsi-fungsi bantu seperti preprocessing data, training model, dan prediksi hasil.
- `pages/`: Folder berisi halaman tambahan Streamlit (fitur multi-page), seperti upload dataset, analisis algoritma, input data user, dan informasi dataset.
- `assets/logo.png`: Logo atau aset gambar yang digunakan di tampilan aplikasi.
- `requirements.txt`: File berisi daftar pustaka Python yang dibutuhkan untuk menjalankan aplikasi.
- `heart_disease.csv`: Dataset utama yang digunakan untuk melatih model prediksi penyakit jantung.

## 6. Screenshot Tampilan Aplikasi

Bagian ini disediakan untuk menambahkan screenshot masing-masing fitur aplikasi.


- Halaman Utama Aplikasi (Home Page)



- Halaman Analisis Algoritma

Data setelah dilakukan proses upload, selanjutnya akan mengambilnya kemudian dilakukan pada menu analisis algoritma. Disini juga dilakukan proses Data Understanding, Data Preparation.

Klasifikasi Penyakit Jantung  
**Analisis Algoritma**  
Input Data User  
Informasi Dataset




## Analisis Algoritma (CRISP-DM)


### Data Understanding

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303	303	303	303	303	303	303	303	303	303	303	303	303	303
mean	54.3663	0.6832	0.967	131.6238	246.264	0.1485	0.5281	149.6469	0.3267	1.0396	1.3993	0.7294	2.3135	0.5446
std	9.0821	0.466	1.0321	17.5381	51.8308	0.3562	0.5259	22.9052	0.4698	1.1611	0.6162	1.0226	0.6123	0.4988
min	29	0	0	94	126	0	0	71	0	0	0	0	0	0
25%	47.5	0	0	120	211	0	0	133.5	0	0	1	0	2	0
50%	55	1	1	130	240	0	1	153	0	0.8	1	0	2	1
75%	61	1	2	140	274.5	0	1	166	1	1.6	2	1	3	1
max	77	1	3	200	564	1	2	202	1	6.2	2	4	3	1

Memilih Analisis Algoritma yang akan digunakan diantaranya Random Forest, Support Vector Machine, Logistics Regression, K-Nearest Neighbors, Gradient Boosting. Dalam proses ini mengambil konsep training 80% dan testing 20%:

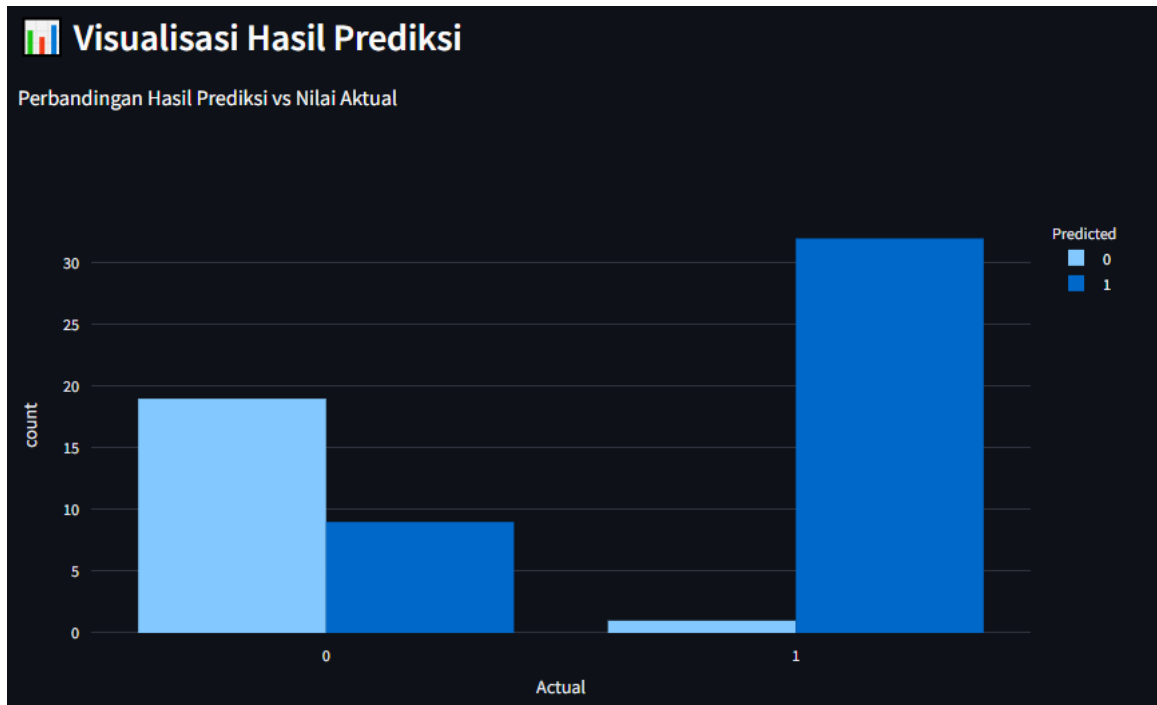
Pilih algoritma  
Random Forest

 Jalankan Analisis

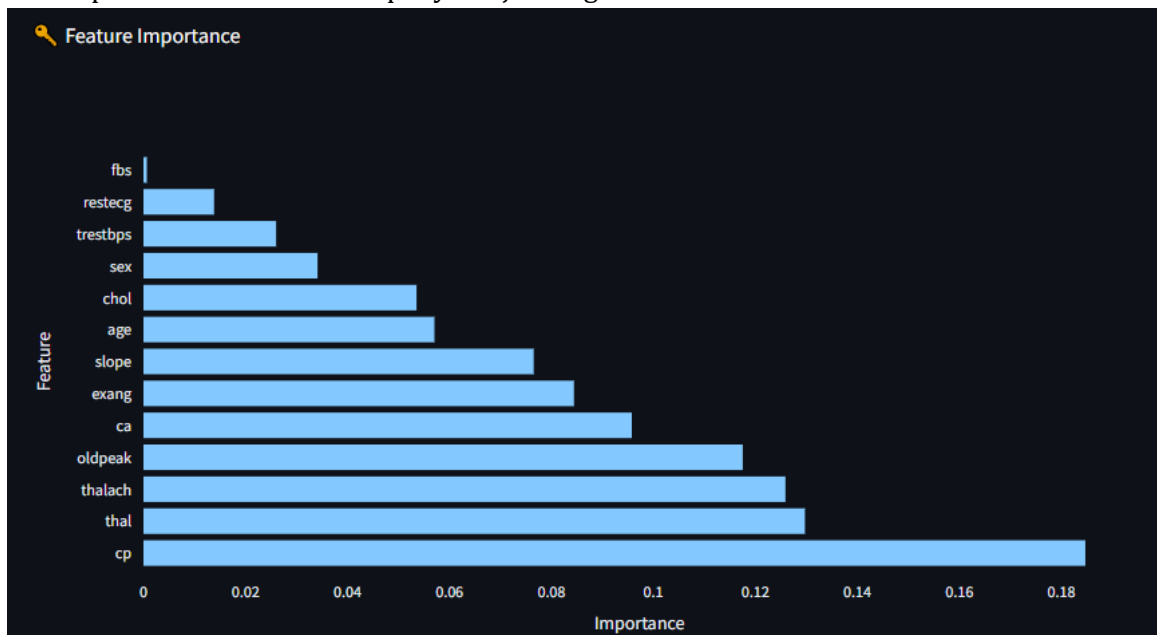
 **Evaluasi Model**  
Akurasi: 0.8361  
precision recall f1-score support  
0 0.95 0.68 0.79 28  
1 0.78 0.97 0.86 33  
accuracy 0.84 61  
macro avg 0.87 0.82 0.83 61  
weighted avg 0.86 0.84 0.83 61

Visualisasi antara Hasil Prediksi vs Nilai Aktual

Digunakan untuk membandingkan hasil prediksi model dengan nilai sebenarnya (aktual). Tujuannya adalah untuk melihat seberapa baik model memprediksi kelas atau nilai target dengan cara visual bukan hanya angka metrik seperti akurasi.



Visualisasi Feature Importance yang menunjukkan fitur atau variabel apa yang penting dalam proses klasifikasi biner penyakit jantung.



ROC (Receiver Operating Characteristic) adalah grafik yang digunakan untuk mengevaluasi kinerja model klasifikasi biner. Grafik ini menggambarkan kemampuan model dalam membedakan antara kelas positif dan negatif pada berbagai threshold (ambang batas) probabilitas.

ROC Curve digunakan untuk mengevaluasi performa model klasifikasi bukan hanya seberapa sering benar, tapi seberapa baik model membedakan antara kelas positif dan negatif. Mengambil probabilitas prediksi kelas positif (1) dari model.

ROC membutuhkan nilai probabilitas ini, bukan hasil klasifikasi langsung.

Menghitung:

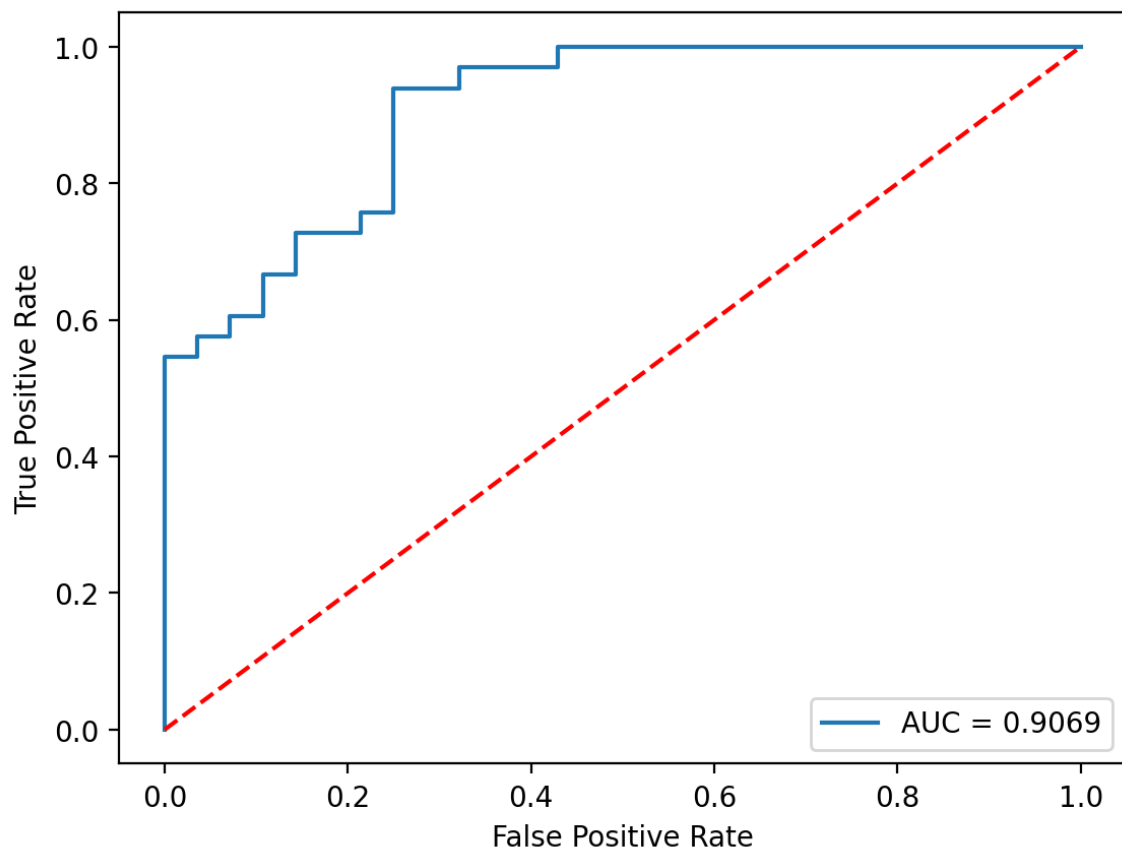
- TPR (True Positive Rate) = seberapa banyak positif yang terdeteksi benar
- FPR (False Positive Rate) = seberapa banyak negatif yang salah diklasifikasikan sebagai positif
- ROC curve adalah plot antara TPR vs FPR pada berbagai ambang batas (threshold) prediksi.

Menghitung AUC (Area Under Curve), yaitu luas di bawah kurva ROC.

Nilai AUC ini menjadi angka ringkas untuk menilai performa model secara keseluruhan:

- $AUC = 1 \rightarrow$  model sempurna
- $AUC = 0.5 \rightarrow$  model tidak lebih baik dari tebak acak
- $AUC < 0.5 \rightarrow$  model justru salah arah

ROC Curve (AUC = 0.9069) Artinya: model memiliki kemampuan membedakan positif dan negatif dengan akurasi area 90.69% ini sangat baik.



Precision-Recall (PR) Curve membantu melihat seberapa akurat model mendeteksi kelas positif tanpa banyak menghasilkan kesalahan. Digunakan untuk mengevaluasi performa model klasifikasi khususnya pada data yang tidak seimbang (imbalanced), yaitu ketika jumlah kelas positif dan negatif tidak sama (misalnya 90% negatif dan hanya 10% positif).

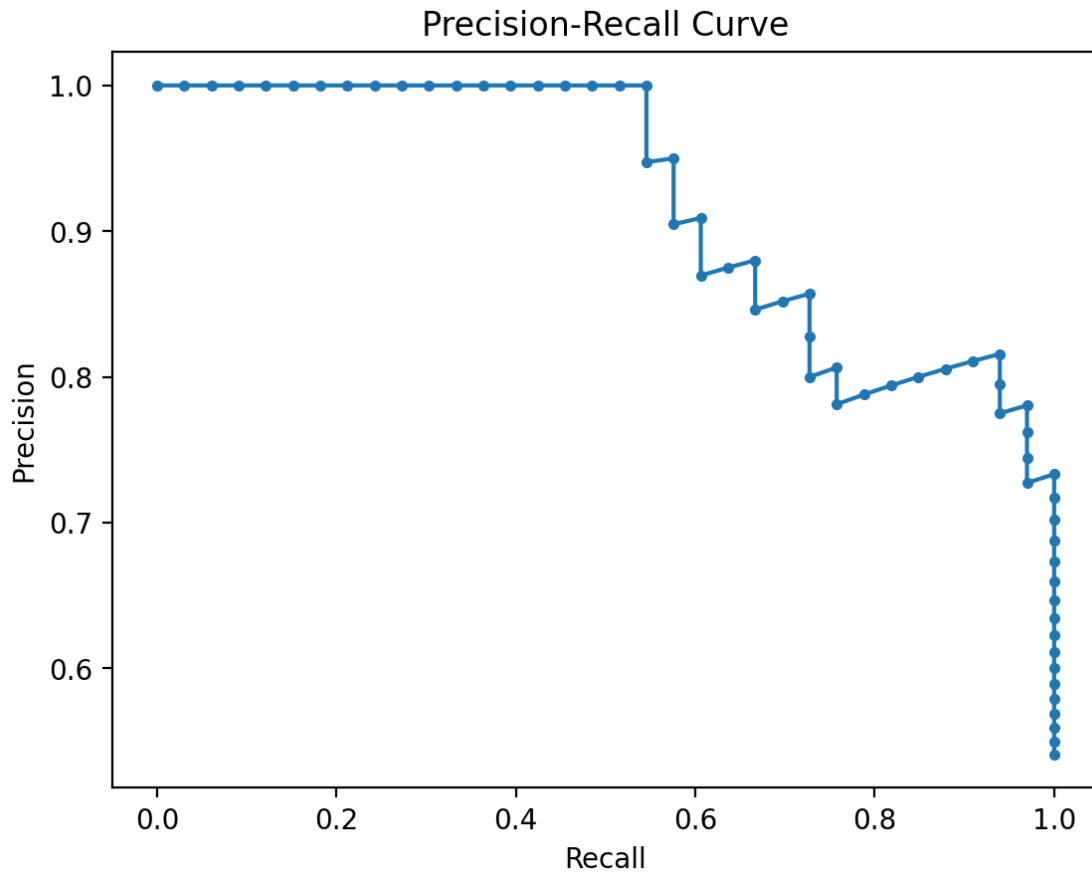
Menghitung dua metrik utama berdasarkan hasil probabilitas:

- Precision = dari semua yang diprediksi positif, berapa yang benar-benar positif.  
Rumus:  $TP / (TP + FP)$
- Recall (Sensitivity) = dari semua data positif yang sebenarnya, berapa yang berhasil dideteksi model.  
Rumus:  $TP / (TP + FN)$

ROC Curve bisa menyesatkan bila data tidak seimbang, jadi Precision-Recall Curve lebih cocok karena fokus pada kinerja model di kelas positif (minoritas).

Makna Precision-Recall Curve:

- Kurva yang mendekati sudut kanan atas (precision dan recall tinggi) menunjukkan model yang sangat baik.
- Kurva yang datar mendekati bawah berarti model banyak melakukan kesalahan atau tidak mampu mendeteksi kelas positif dengan baik.
- Bila digunakan bersama dengan AUC-PR (Area Under Curve), nilai mendekati 1.0 menunjukkan performa tinggi.



Dari proses analisis algoritma hasilnya langsung dapat didownload dalam format excel.

**Download Hasil Prediksi (Excel)**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	Actual	Predicted					
1	57	1	0	150	276	0	0	112	1	0,6	1	1	1	0	0					
2	67	1	0	125	254	1	1	163	0	0,2	1	2	3	0	0					
3	46	1	0	140	311	0	1	120	1	1,8	1	2	3	0	0					
4	60	1	2	140	185	0	0	155	0	3	1	0	2	0	1					
5	50	1	2	140	233	0	1	163	0	0,6	1	1	3	0	1					
6	70	1	2	160	269	0	1	112	1	2,9	1	1	3	0	0					
7	56	1	1	120	240	0	1	169	0	0	0	0	2	1	1					
8	62	0	0	140	268	0	0	160	0	3,6	0	2	2	0	0					
9	60	0	2	120	178	1	1	96	0	0	2	0	2	1	1					
10	59	1	3	170	288	0	0	159	0	0,2	1	0	3	0	1					
11	53	1	0	142	226	0	0	111	1	0	2	0	3	1	0					
12	45	0	0	138	236	0	0	152	1	0,2	1	0	2	1	1					
13	50	1	0	144	200	0	0	126	1	0,9	1	0	3	0	0					
14	63	0	1	140	195	0	1	179	0	0	2	2	2	1	1					
15	54	0	2	160	201	0	1	163	0	0	2	1	2	1	1					
16	42	1	3	148	244	0	0	178	0	0,8	2	2	2	1	1					
17	51	1	2	110	175	0	1	123	0	0,6	2	0	2	1	1					
18	53	0	0	130	264	0	0	143	0	0,4	1	0	2	1	1					
19	65	0	2	140	417	1	0	157	0	0,8	2	1	2	1	1					
20	63	1	3	145	233	1	0	150	0	2,3	0	0	1	1	1					



- **Form Input Data Pasien**

User langsung memasukkan data inputan pengujian. Proses masih menggunakan konsep algoritma yang terakhir dipilih pada menu Analisis Algoritma dalam hal ini masih menggunakan Algoritma Random Forest.

Setelah menekan tombol jalankan prediksi, maka hasilnya langsung memunculkan diagnosa berisiko atau tidak berisiko terkena penyakit jantung. Sekaligus terdapat rekomendasi menjaga kesehatan jantung.

Hasil diagnosa Pasien Tidak Berisiko Terkena Penyakit Jantung.


## Hasil diagnosa Pasien Berisiko Terkena Penyakit Jantung.

Klasifikasi Penyakit Jantung

Analisis Algoritma

Input Data User

Informasi Dataset



Thalassemia (0 = Normal, 1 = Fixed Defect, 2 = Reversible Defect)

2

Jalankan Prediksi

Hasil Prediksi

Hasil Prediksi: Pasien berisiko terkena penyakit jantung.

Rekomendasi Tindak Lanjut:

1. Segera konsultasi dengan dokter spesialis jantung.
2. Terapkan pola makan sehat (kurangi garam & lemak jenuh).
3. Lakukan aktivitas fisik rutin.
4. Hentikan kebiasaan merokok dan batasi alkohol.
5. Kontrol berat badan & tekanan darah secara berkala.
6. Rutin cek kesehatan (kolesterol, gula darah, tekanan darah).

Catatan: Ini hanya prediksi berbasis data, bukan diagnosis resmi.

## 7. Catatan Pengembangan

Aplikasi ini masih dapat dikembangkan lebih lanjut dengan menambahkan fitur-fitur seperti:

- Validasi input yang lebih ketat
- Pilihan model machine learning tambahan
- Penyimpanan hasil prediksi ke database
- Integrasi API untuk akses data eksternal

## 8. Profil dan Kontak

Nama: Arjun Yuda Firwanda

Role: Team Program Liraa Job

Email: (arjunyudafirwanda@gmail.com)

LinkedIn: (<https://www.linkedin.com/in/arjun-yuda-firwanda-8095261a4/>)

GitHub: (<https://github.com/arjunyudafirwanda/>)