JIPSI: Jurnal Ilmu Pendidikan dan Sains Islam Interdisipiner
Vol. 5, No. 3, Agustus 2026, pp. 2725-2739
http://journal.amorfati.id/index.php/jipsi || ISSN 2962-9187

Interdisciplinary Scope of Science and Education, Islamic Education and Science, Integration of Science and Islam, Islamic Thought and Philosophy, Integration of Social Sciences, Humanities, and Islamic Studies, Islamic Economics and Business Studies, and Islamic Ethics in Science and Technology

Sistem Prediksi Risiko Penyakit Jantung Menggunakan Metode Logistic Regression Berbasis Optimasi Data

Robieth Sohiburoyyan[a,1,*], Devan Herdiansyah Nugroho[b,2], Salamet Nur Himawan[c,3], Vera Wati[d,4], Rosena Shintabella[e,5]

[*a,b,c,d,e] Politeknik Negeri Indramayu, Indonesia.
[1]Robieth.s@polindra.ac.id; [2]Devanherdiansyah74@gmail.com; [3]snhimawan@polindra.ac.id; [4]vera.w@polindra.ac.id; [5]rosenasbella@polindra.ac.id
*Correspondent Author; Robieth.s@polindra.ac.id

ARTICLE INFO

Article history
Received:
19-06-2026
Revised:
24-07-2026
Accepted:
31-08-2026

Keywords

Penyakit Jantung, Machine Learning, Logistic Regression, Supervised Learning, Klasifikasi.

ABSTRACT

Penyakit jantung masih menjadi penyebab kematian yang signifikan diberbagai negara. Kondisi tersebut menunjukkan pentingnya suatu sistem pendukung keputusan yang dapat membantu tenaga medis dalam mendeteksi risiko penyakit jantung sejak dini secara cepat dan akurat. Penelitian ini berfokus pada pengembangan sistem berbasis *machine learning* untuk memprediksi risiko penyakit jantung melalui penerapan teknik penyeimbangan data, pemilihan fitur, serta optimalisasi model. Dengan pembagian dataset menjadi 80% data pelatihan serta 20% data pengujian dengan metode train test split. Pendekatan supervised learning digunakan dengan membandingkan enam algoritma klasifikasi, yaitu K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM), Naive Bayes, dan Logistic Regression. Ketidakseimbangan distribusi data ditangani menggunakan Synthetic Minority Over-sampling Technique (SMOTE), sedangkan penentuan fitur yang relevan dilakukan melalui Embedded Methods dengan Random Forest. Selanjutnya, GridSearchCV digunakan untuk menentukan konfigurasi hyperparameter terbaik. Kinerja setiap model dianalisis berdasarkan accuracy, precision, recall, F1-score, dan ROC-AUC. Berdasarkan hasil pengujian dengan dua hasil klasifikasi berupa berisiko dan tidak berisiko, Logistic Regression sebelum proses hyperparameter tuning menghasilkan performa paling baik, dengan accuracy sebesar 82,06%, precision 85,45%, recall 84,68%, F1-score 85,06%, serta ROC-AUC 87,26%. Performa dari Model terbaik tersebut selanjutnya diintegrasikan ke dalam aplikasi berbasis web dan mobile sebagai sarana pendukung untuk melakukan identifikasi dini terhadap risiko penyakit jantung.

Pendahuluan

Penyakit jantung merupakan gangguan pada struktur atau fungsi jantung yang dapat melibatkan pembuluh darah, katup, otot, maupun lapisan pelindung jantung. Kondisi tersebut dapat dipengaruhi oleh berbagai faktor, seperti penumpukan lemak pada pembuluh koroner, hipertensi, diabetes yang tidak terkontrol, serta infeksi (Alodokter et al., 2026). Penyakit ini menjadi salah satu masalah kesehatan global karena penyakit kardiovaskular menyebabkan sekitar 17,9 juta kematian setiap tahun di dunia (WHO, 2025). Di Indonesia, penyakit jantung juga dilaporkan sebagai salah satu penyebab kematian utama sehingga upaya deteksi risiko sejak dini menjadi penting untuk mencegah komplikasi dan menekan angka kematian (Sadikin, 2025).

Risiko penyakit jantung dipengaruhi oleh beberapa faktor, usia dan jenis kelamin termasuk faktor yang tetap atau tidak dapat diubah, sedangkan tekanan darah, kadar kolesterol, dan gula darah merupakan beberapa indikator klinis yang dapat dikendalikan (Silalahi, 2024; Laranjo et al., 2024). Selain itu, nyeri dada saat beraktivitas, perubahan denyut jantung maksimum, serta depresi segmen ST pada pemeriksaan Elektrokardiogram (EKG) dapat menjadi informasi penting dalam menilai kemungkinan adanya gangguan pada jantung (Dixon et al., 2023).

Proses deteksi penyakit jantung secara konvensional masih memiliki sejumlah keterbatasan. Interpretasi EKG, pengamatan gejala, serta penggunaan kalkulator risiko kardiovaskular membutuhkan ketelitian dan dapat menghadapi kendala ketika pasien menunjukkan gejala yang tidak spesifik (*Cardiovascular Diseases (CVDs) Key Facts*, 2025). Di sisi lain, banyaknya variabel medis dan hubungan antarfaktor yang kompleks menyebabkan pengolahan data secara manual menjadi kurang efisien. Oleh karena itu, diperlukan pendekatan komputasional yang dapat membantu proses skrining secara lebih cepat, konsisten, dan objektif (Alvin Rajkomar, Jeffrey Dean, 2024).

Perkembangan *data science* memberikan peluang untuk memanfaatkan *machine learning* dalam menganalisis data kesehatan dan membangun model prediksi penyakit. Pendekatan ini mampu mempelajari pola dari sejumlah variabel klinis untuk memperkirakan risiko penyakit jantung (Hidayat et al., 2024). Kemampuannya dalam mengolah data dalam jumlah besar dan membangun pola prediktif juga mendorong semakin luasnya penerapan *machine learning* pada bidang kesehatan (Anita et al., 2025).

Sejumlah algoritma klasifikasi telah digunakan dalam penelitian prediksi penyakit jantung, antara lain *Naive Bayes, K-Nearest Neighbors* (KNN), *Decision Tree, Random Forest, Support Vector Machine* (SVM), dan *Logistic Regression*. Penelitian menggunakan dataset Cleveland mengungkapkan bahwa *Naive Bayes* memperoleh akurasi 84,67%, sedangkan *Logistic Regression* dan *Random Forest* masing-masing mencapai 84,30% dan 81,70% (Pringsewu, 2025). Studi lain yang membandingkan *Naive Bayes, Decision Tree*, dan KNN menghasilkan akurasi tertinggi sebesar 86,88% pada *Naive Bayes* (Larassati et al., 2022). Sementara itu, perbandingan beberapa algoritma seperti *Random Forest*, SVM, *Gradient Boosting Machines*, XGBoost, dan LightGBM menunjukkan adanya perbedaan performa antaralgoritma dalam memprediksi penyakit jantung (Nugraha, 2023). Temuan tersebut memperlihatkan bahwa pemilihan serta pengoptimalan model masih menjadi aspek penting dalam memperoleh kemampuan prediksi yang baik.

Berdasarkan permasalahan dan penelitian terdahulu, penelitian ini menggunakan *Logistic Regression* sebagai model utama untuk mengklasifikasikan risiko penyakit jantung. Algoritma tersebut dipilih karena sesuai untuk permasalahan klasifikasi biner dan mampu menghasilkan probabilitas keanggotaan suatu data terhadap dua kelas melalui fungsi sigmoid. Penelitian difokuskan pada peningkatan kemampuan generalisasi model

sehingga dapat memberikan prediksi yang lebih baik terhadap data yang belum pernah dipelajari sebelumnya.

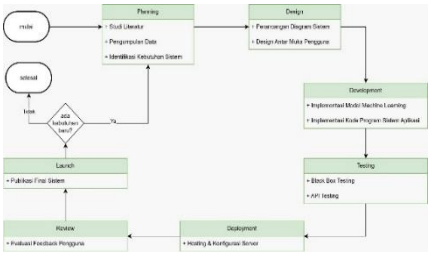
Pengembangan model juga mempertimbangkan karakteristik data medis yang dapat mengalami perbedaan proporsi kelas dan mengandung fitur dengan tingkat relevansi berbeda. *Synthetic Minority Over-sampling Technique* (SMOTE) diterapkan untuk menyeimbangkan distribusi kelas melalui pembentukan sampel sintetis pada kelas minoritas (Setiawan et al., 2025). Selanjutnya, *feature selection* menggunakan *Embedded Methods* berbasis *Random Forest* digunakan untuk memilih fitur yang memberikan kontribusi lebih besar terhadap prediksi (Merceedi & Abdulazeez, 2025). Tahap optimasi dilakukan melalui *hyperparameter tuning* menggunakan *GridSearchCV* untuk menentukan konfigurasi parameter yang memberikan performa terbaik (Alemerien, 2024). Dengan rangkaian tahapan tersebut, penelitian ini diarahkan untuk menghasilkan model prediksi risiko penyakit jantung yang memiliki performa baik sekaligus lebih efisien dalam memanfaatkan fitur klinis.

Metode

Penelitian ini mengadopsi pendekatan kuantitatif yang didasarkan pada pengolahan data numerik yang bersumber dari rekam medis pasien. Data klinis tersebut diolah dan dimodelkan menggunakan *machine learning* untuk menentukan tingkat risiko penyakit jantung. Kinerja setiap model diukur berdasarkan *accuracy, precision, recall, F1-Score*, dan *ROC-AUC* sebagai dasar penentuan model klasifikasi terbaik.

Model yang diperoleh selanjutnya diintegrasikan ke dalam aplikasi berbasis *website*. Pengembangan aplikasi menggunakan Agile Development yang memungkinkan proses pengerjaan dilakukan secara bertahap dan dapat disesuaikan dengan kebutuhan selama penelitian. Pelaksanaannya menggunakan beberapa siklus (*sprint*) yang mencakup tahapan berikut:

- *Planning & Design.* Tahap awal mencakup kajian literatur, pengumpulan dataset, analisis kebutuhan, serta penyusunan rancangan sistem. Hasil perencanaan kemudian dituangkan dalam pemodelan UML dan *mockup* antarmuka aplikasi.
- *Development & Testing.* Rancangan sistem diimplementasikan melalui pengembangan model *machine learning* dan pembuatan aplikasi. Setiap fungsi yang telah dibangun kemudian diverifikasi menggunakan *Black Box Testing* dan pengujian API.
- *Deployment, Review & Launch.* Sistem yang telah selesai diuji diterapkan pada server dan dievaluasi kembali. Temuan pada proses *review* digunakan sebagai dasar penyempurnaan pada iterasi berikutnya. Setelah seluruh fungsi memenuhi kebutuhan yang ditetapkan, aplikasi memasuki tahap *launch* dan siap digunakan.



Gambar 1 Tahapan Penelitian

1. *Planning*

Tahap *Planning* diawali dengan melakukan analisis kebutuhan sistem yang digunakan untuk mengenali permasalahan serta menetapkan ruang lingkup penelitian, serta merumuskan kebutuhan pengguna terhadap sistem yang akan dikembangkan. Pada tahap ini dilakukan studi literatur, pengumpulan data, serta identifikasi kebutuhan sistem baik fungsional maupun non-fungsional. Selain itu, ditentukan juga spesifikasi lingkungan

pengembangan, baik pada lingkungan lokal maupun *cloud environment*, guna mendukung proses pelatihan model *machine learning* dan pengembangan sistem berbasis *website*. Dalam pendekatan Agile, tahap perencanaan ini bersifat fleksibel dan dapat diperbarui pada setiap iterasi sesuai dengan hasil evaluasi dan umpan balik yang diperoleh selama proses pengembangan.

a. Studi Literatur

Penelitian diawali dengan Studi Literatur yang bertujuan untuk mengkaji teori dari penelitian terdahulu terkait prediksi risiko penyakit jantung menggunakan *machine learning*. Studi ini mencakup pembahasan algoritma klasifikasi, penanganan ketidakseimbangan data menggunakan SMOTE, optimasi model dengan *hyperparameter tuning* menggunakan metode *GridSearchCV,* serta *feature selection* menggunakan metode *Embedded Methods* sebagai dasar dalam perancangan sistem prediksi penyakit jantung berbasis *website*.

b. Pengumpulan Data

Penelitian ini menggunakan dataset yang bersumber dari Kaggle sebagai data pelatihan (*training data*), yaitu dataset Heart Disease UCI yang terdiri dari 920 data pasien dengan 16 atribut klinis. Label target pada dataset tersebut dibagi menjadi dua kelas, yaitu negatif (0) dan positif (1). Selain itu, dilakukan pengumpulan data pengujian (*testing data*) sebanyak 300 data melalui observasi langsung pada mitra penelitian, yaitu Rumah Sakit Umum Daerah (RSUD) Indramayu yang berlokasi di Jl. Murahnara No.7, Sindang, Kec. Sindang, Kabupaten Indramayu, Jawa Barat 45222. Kegiatan ini bertujuan untuk mendukung validasi dan evaluasi model yang telah dikembangkan. Data yang diperoleh merupakan data sekunder berupa rekapitulasi kasus penyakit jantung yang dihimpun dari arsip dan dokumen resmi rumah sakit.

c. Identifikasi Kebutuhan Sistem

Identifikasi kebutuhan sistem dilakukan untuk memastikan bahwa aplikasi yang dikembangkan mampu memenuhi tujuan penelitian sekaligus memberikan kemudahan bagi pengguna dalam melakukan klasifikasi espon penyakit jantung. Pada tahap ini, kebutuhan sistem dibagi dua, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

1). Kebutuhan Fungsional

- Sistem harus dapat menerima input berupa data gejala klinis pasien.
- Sistem harus memproses input tersebut dengan memanfaatkan model *machine learning* terbaik melalui API berbasis Flask.
- Sistem harus menampilkan hasil klasifikasi risiko penyakit jantung secara langsung pada antarmuka *website*
- Sistem harus menyimpan setiap hasil prediksi yang dilakukan oleh petugas ke dalam basis data sebagai espons klasifikasi.
- Sistem harus menyediakan fitur authentikasi agar hanya pengguna yang berwenang yang dapat mengakses aplikasi.
- Sistem harus memungkinkan petugas dan admin untuk melihat kembali respons klasifikasi yang telah dilakukan.

2). Kebutuhan Non-Fungsional

- Sistem harus memiliki antarmuka yang sederhana, responsive, dan mudah dipahami oleh pengguna.
- Sistem harus mampu memberikan hasil klasifikasi dengan cepat dan akurat.
- Sistem harus dapat dijalankan pada lingkungan berbasis *website* menggunakan browser populer, seperti Google Chrome.
- Sistem harus memiliki tingkat keamanan yang memadai untuk melindungi data pengguna dan hasil klasifikasi.

- Sistem harus mudah dipelihara dan dikembangkan untuk kebutuhan di masa mendatang.

Dengan adanya identifikasi kebutuhan sistem ini, proses perancangan dan implementasi dapat lebih terarah sehingga aplikasi yang dihasilkan mampu berfungsi sesuai tujuan penelitian serta memberikan manfaat nyata bagi pengguna.

d. *Design* (Perancangan Sistem)

Pada tahapan ini memberikan ambaran menyeluruh mengenai sistem klasifikasi penyakit jantung yang akan dibangun. Pada tahap ini disajikan rancangan sistem berupa *Diagram* alur kerja serta pemodelan dengan *Unified Modeling Language* (UML), yang mencakup *Use Case Diagram, Activity Diagram,* dan *Sequence Diagram*. Selain itu, dilakukan pula perancangan desain antarmuka pengguna (*Mockup*) beserta struktur menu yang tersedia dalam aplikasi.

2. *Development*

Pada tahap ini, sistem dikembangkan berdasarkan spesifikasi desain yang telah dirancang sebelumnya. Proses implementasi dibagi menjadi dua bagian utama, yaitu pengembangan model *machine learning* dan pengembangan sistem aplikasi. Pengembangan model fokus pada pelatihan algoritma menggunakan dataset yang sudah dikumpulkan, sementara pengembangan sistem memastikan fungsionalitas aplikasi berjalan sesuai dengan alur yang telah ditentukan.

a. Pengembangan Model *Machine Learning*

Pengembangan model *machine learning* dilakukan melalui beberapa tahapan utama, yaitu *preprocessing* data, *splitting* data, pelatihan, evaluasi, dan penyimpanan model. Dilakukan tahap *preprocessing* berupa penanganan *missing values,* penghapusan data duplikat, analisis distribusi fitur, serta normalisasi agar data siap digunakan. Dataset dibagi menjadi data latih (80%) dan data uji (20%) dengan metode train_test_split.

Proses pelatihan dilakukan pada Jupyter Notebook dengan enam algoritma algoritma klasifikasi, yaitu Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, Decision Tree, dan Naive Bayes. Evaluasi model menggunakan lima metrik evaluasi, yaitu *Accuracy, Precision, Recall, F1-Score,* dan *ROC-AUC* serta *classification report* dan *confusion matrix* untuk memperoleh hasil yang lebih representatif.

Model dengan performa terbaik kemudian disimpan menggunakan joblib bersama objek StandardScaler, Important Features, X_train columns, serta nilai median untuk fitur numerik dan mode untuk fitur kategorikal, sehingga dapat diintegrasikan ke dalam aplikasi berbasis *website* untuk memberikan prediksi klasifikasi risiko penyakit jantung secara cepat dan akurat.

b. Pengembangan Sistem Aplikasi

Pada bagian ini akan menguraikan secara rinci proses pengembangan sistem aplikasi, termasuk tahapan-tahapan yang dilakukan menggunakan metode agile. Metode ini digunakan karena memiliki sifat yang adaptif dan fleksibel terhadap perubahan kebutuhan fungsionalitas sistem. Pada tahap ini, sistem dikembangkan berdasarkan spesifikasi desain yang telah dirancang sebelumnya. Proses pengembangan dibagi menjadi dua bagian utama, yaitu *frontend* dan *backend,* sehingga pemrograman dilakukan secara terintegrasi menggunakan Visual Studio Code, sementara pengolahan basis data memanfaatkan PostgreSQL yang dijalankan melalui pgAdmin. Pendekatan ini memastikan setiap fungsi berjalan sesuai harapan, di mana proses *debugging* dan pengujian fungsional dilakukan melalui browser Google Chrome.

Dalam implementasi sistem aplikasi ini bertujuan untuk merealisasikan hasil penelitian ke dalam bentuk aplikasi berbasis *website* yang dapat digunakan langsung oleh

pengguna. Model *machine learning* terbaik yang telah disimpan dalam bentuk file dipanggil kembali melalui API berbasis Flask, yang berfungsi menerima input data klinis, serta menghasilkan keluaran berupa klasifikasi risiko penyakit jantung. API ini kemudian dihubungkan dengan sistem *frontend* yang dikembangkan menggunakan Laravel, di mana pengguna dapat memasukkan data melalui *form* input, dan hasil prediksi ditampilkan secara langsung pada antarmuka.

3. *Testing* (Pengujian Sistem)

Pada tahap ini, dilakukan pengujian untuk mengevaluasi sistem, berfungsi agar sesuai dengan kebutuhan yang ditentukan. Pengujian dilakukan dengan pendekatan *Black Box Testing*, yaitu dengan fungsionalitas aplikasi dari sisi pengguna tanpa melihat bagian dalam atau kode programnya. Metode ini berfokus pada pemberian input dan pengamatan terhadap output, hal ini untuk memastikan setiap fitur, seperti menginput pasien, menginput klasifikasi, dan menginput artikel, dapat berjalan dengan baik.

4. Deployment

Pada tahap ini, dilakukan tahap *deployment* yang merupakan proses penerapan sistem yang telah selesai dikembangkan dan diuji agar dapat digunakan oleh pengguna. Pada tahap ini dilakukan konfigurasi server aplikasi, integrasi antara antarmuka web dengan API model prediksi, serta pengaturan database untuk menyimpan data pasien, hasil klasifikasi, dan artikel kesehatan. Dengan demikian, sistem dapat diakses dan digunakan sesuai dengan kebutuhan pengguna serta memungkinkan dilakukan pemeliharaan dan pembaruan sistem secara berkelanjutan.

5. Review

Tahap ini dilakukan untuk meninjau kembali sistem yang telah dikembangkan dan diuji. Pada tahap ini dilakukan evaluasi terhadap fungsi sistem, kinerja model prediksi, serta kesesuaian fitur dengan kebutuhan pengguna. Hasil dari proses *review* digunakan sebagai bahan perbaikan sistem apabila masih ditemukan kekurangan.

6. Launch

Pada tahap ini, sistem akan masuk kedalam tahap *launch* yang merupakan tahap akhir dalam proses pengembangan, yaitu ketika sistem siap digunakan secara resmi oleh pengguna. Pada tahap ini aplikasi telah melalui proses pengujian, *deployment*, dan *review*, sehingga sistem dapat digunakan untuk melakukan klasifikasi risiko penyakit jantung serta mengelola data yang tersedia pada aplikasi.

Hasil dan Pembahasan

1. Hasil Pengumpulan Data

Penelitian ini menggunakan dua sumber data, yaitu data pelatihan (*training data*) dan data pengujian (*testing data*). Data pelatihan diperoleh dari Heart Disease UCI Dataset yang tersedia di Kaggle dengan total 920 data dan 16 atribut klinis yang digunakan sebagai fitur prediksi. Dataset tersebut terdiri atas dua kelas, yaitu berisiko penyakit jantung (positif) dan tidak berisiko penyakit jantung (negatif).

Data pengujian diperoleh melalui observasi di RSUD Indramayu sebanyak 300 data pasien yang digunakan untuk memvalidasi performa model pada data nyata. Data yang diperoleh merupakan data sekunder dari rekapitulasi kasus penyakit jantung yang masih mengandung beberapa permasalahan, seperti *missing value* dan ketidakkonsistenan format data.

Oleh karena itu, sebelum digunakan dalam proses pemodelan, seluruh data melalui tahap *preprocessing* yang meliputi pembersihan data, penanganan *missing value,* standarisasi format, dan transformasi data. Tahapan ini bertujuan untuk meningkatkan

kualitas data sehingga model *machine learning* dapat menghasilkan prediksi risiko penyakit jantung yang lebih akurat dan andal.

Fitur	Keterangan
age	Numerik
sex	Kategorikal (0 = Female, 1 = Male)
cp	Kategorikal (0 = typical angina, 1 = atypical angina, 2 = non-anginal, 3 = asymptomatic)
trestbps	Tekanan Darah Istirahat (mmHg) - Numerik
chol	Kadar Kolesterol Serum (mg/dl) - Numerik
fbs	Kategorikal (1 = ≥ 120 mg/dl, 0 = < 120 mg/dl)
restecg	Kategorikal (0 = normal, 1 = st-t abnormality, 2 = lv hypertrophy)
thalch	Numerik
exang	Kategorikal (0 = Tidak, 1 = Ya)
oldpeak	Numerik
target	Kategorikal (0 = Tidak Berisiko, 1 = Berisiko)

Tabel 1. Fitur untuk Prediksi Risiko Penyakit Jantung

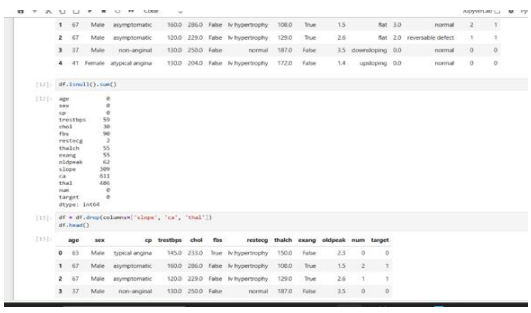
2. Hasil Preprocessing

Preprocessing data merupakan proses penting sebelum data digunakan dalam pelatihan model *machine learning*. Tujuan dari tahap ini adalah untuk memastikan bahwa data dalam kondisi bersih, konsisten, dan memiliki format yang sesuai agar dapat diproses secara optimal oleh model. Data yang diperoleh dari Kaggle dan RSUD Kabupaten Indramayu berupa data mentah yang mengandung berbagai permasalahan, seperti ketidakkonsistenan format, nilai yang hilang (*missing value*), data duplikat, serta beberapa fitur yang belum siap digunakan secara langsung. Oleh karena itu, dilakukan serangkaian proses *preprocessing* berikut untuk meningkatkan kualitas data sebelum digunakan pada tahap pemodelan:

a. Penanganan Data Kosong (*Missing Value*)

Pada tahap ini dilakukan proses identifikasi nilai kosong (*missing value*), berdasarkan hasil identifikasi, ditemukan beberapa entri yang memiliki data tidak lengkap, khususnya pada kolom yang berkaitan dengan informasi klinis. Data yang tidak lengkap berpotensi menurunkan performa model serta menghasilkan prediksi yang tidak akurat. Oleh sebab itu, dilakukan penanganan dengan cara menghapus (*drop*) data yang memiliki nilai kosong pada fitur-fitur utama, seperti usia, jenis kelamin, dan parameter pemeriksaan lainnya. Sehingga hanya data yang lengkap yang digunakan dalam proses modeling.

Setelah proses penanganan *missing value* dilakukan, dataset menjadi lebih bersih dan siap digunakan untuk tahap selanjutnya. Hasil pemeriksaan terhadap *missing value* dapat dilihat pada gambar berikut.



Gambar 2. Hasil Pemeriksaan *Missing Value*

Pada Gambar 2 menunjukan bahwa hasil pemeriksaan, ditemukan fitur *trestbps, chol, fbs, restecg, thalch, exang, oldpeak, slope, ca,* dan *thal* memiliki sejumlah nilai yang kosong. Sementara itu, fitur-fitur lainnya seperti *age, sex, cp* dan target tidak memiliki nilai yang

hilang. Penemuan missing value ini menunjukan bahwa data perlu melalui tahapan penanganan lebih lanjut sebelum dapat digunakan untuk melatih model. Berdasarkan hasil identifikasi *missing value* sebelumnya, dilakukan tahapan penanganan untuk memastikan kualitas data yang optimal.

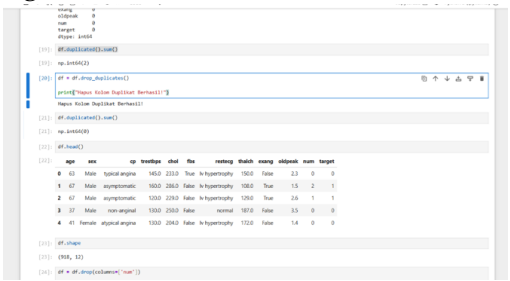
Nilai yang hilang pada fitur *trestbps, chol, thalch,* dan *oldpeak* diatasi dengan menggunakan metode imputasi median. Metode ini dipilih karena nilai median lebih tahan terhadap pengaruh *outlier* dibandingkan nilai rata-rata (*mean*), sehingga mampu menjaga distribusi data tetap stabil. Sementara itu, nilai yang hilang pada fitur *fbs, restecg,* dan *exang* diatasi dengan menggunakan metode imputes *mode*. Karena kedua fitur tersebut bersifat kategorikal, sehingga nilai yang paling sering muncul dianggap paling representatif untuk menggantikan data yang hilang.

Selain itu, fitur *slope, ca,* dan *thal* memiliki jumlah nilai kosong yang sangat tinggi (*extreme missing*), sehingga berpotensi menurunkan kualitas data dan performa model. Oleh karena itu, fitur-fitur tersebut dihapus dari dataset agar tidak memengaruhi hasil proses klasifikasi. Melalui tahapan ini, diharapkan data yang digunakan menjadi lebih bersih, konsisten, dan optimal untuk proses pelatihan model.

b. Penanganan Data Duplikat

Berdasarkan hasil identifikasi, ditemukan beberapa entri yang memiliki data duplikat, khusunya pada data dengan informasi klinis yang sama. Keberadaan data duplikat berpotensi menyebabkan bias pada model serta menurunkan kualitas hasil prediksi. Oleh sebab itu, dilakukan penanganan dengan cara menghapus (*drop*) data yang teridentifikasi sebagai duplikat, sehingga hanya menyisakan data yang unik.

Setelah proses penanganan data duplikat dilakukan, dataset menjadi lebih bersih, tidak redundan, dan lebih representatif untuk digunakan pada tahap selanjutnya. Hasil pemeriksaan terhadap data duplikat menunjukan bahwa hasil pemeriksaan data duplikat, dimana ditemukan sebanyak dua data yang teridentifikasi sebagai duplikat berdasarkan hasil pengecekan. Oleh karena itu, berdasarkan hasil identifikasi tersebut, dilakukan proses penghapusan data duplikat untuk memastikan bahwa dataset yang digunakan memiliki kualitas yang lebih baik, tidak redundan, serta dapat meningkatkan performa model dalam menghasilkan prediksi yang akurat.

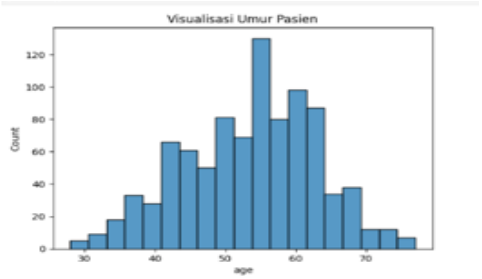


Gambar 3. Hasil Penanganan Data Duplikat

c. *Explanatory Data Analyst* (EDA)

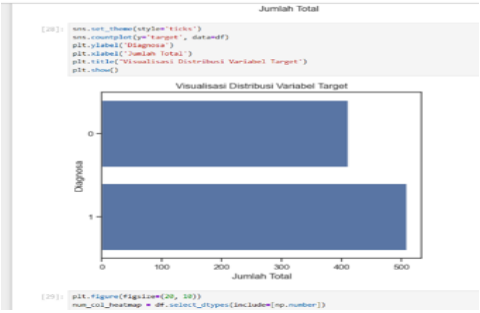
Pada tahap ini dilakukan proses *explanatory data analysis* (EDA) terhadap dataset menggunakan *library seaborn* dan *matplotlib.* Tahapan ini bertujuan untuk memahami karakteristik data, distribusi setiap fitur, serta hubungan antar variabel yang dapat memengaruhi hasil klasifikasi. Analisis dilakukan melalui berbagai visualisasi seperti histogram, *boxplot,* dan *heatmap* korelasi untuk mengidentifikasi pola, tren, serta keberadaan *outlier* dalam data.

Melalui proses EDA, diperoleh wawasan awal mengenai fitur-fitur yang memiliki pengaruh signifikan terhadap target, sehingga dapat membantu dalam pemilihan fitur (*feature selection*) dan meningkatkan performa model yang akan dibangun. Selain itu, EDA juga berperan dalam memastikan bahwa data telah siap dan sesuai untuk digunakan pada tahap pemodelan *machine learning*.



Gambar 4 Visualisasi Data Umur Pasien

Gambar 4 menyajikan distribusi umur pasien berdasarkan hasil *Exploratory Data Analysis* (EDA) dalam bentuk histogram. Visualisasi tersebut memperlihatkan bahwa pasien berusia **50–60 tahun** memiliki frekuensi paling tinggi dibandingkan kelompok umur lainnya. Dengan demikian, data penelitian didominasi oleh pasien pada rentang usia tersebut, yang menggambarkan bahwa kelompok usia paruh baya lebih banyak terwakili dalam dataset.



Gambar 5 Visualisasi Data Variabel Target

Gambar 5 memperlihatkan komposisi kelas pada variabel *target* melalui visualisasi *countplot*. Hasil pengamatan mengindikasikan bahwa data pasien yang termasuk kategori berisiko memiliki jumlah lebih besar daripada kategori tidak berisiko. Perbedaan tersebut mengindikasikan adanya ketidakseimbangan kelas (*class imbalance*) dalam dataset. Untuk mengurangi dampaknya terhadap proses pembelajaran model, SMOTE (*Synthetic Minority Over-sampling Technique*) diterapkan pada data pelatihan guna menambah sampel sintetis pada kelas minoritas. Dengan distribusi kelas yang lebih seimbang, model diharapkan dapat mempelajari karakteristik kedua kelas secara lebih proporsional dan mengurangi kecenderungan prediksi terhadap kelas mayoritas.

Gambar 6 Visualisasi Heatmap Korelasi

Gambar 6 menyajikan *heatmap* korelasi sebagai bagian dari *Exploratory Data Analysis* (EDA) untuk menggambarkan keterkaitan antarvariabel dalam dataset. Nilai korelasi yang mendekati 1 menunjukkan hubungan positif yang semakin kuat, sedangkan nilai yang mendekati -1 mengindikasikan hubungan negatif yang kuat. Sementara itu, nilai di sekitar 0 menunjukkan hubungan linear yang relatif lemah.

Hasil analisis memperlihatkan bahwa tidak terdapat korelasi antarfitur yang melebihi 0,8. Dengan demikian, indikasi multikolinearitas yang kuat tidak ditemukan pada data tersebut. Beberapa fitur menunjukkan keterkaitan dengan variabel *target*, yaitu *exang* dan *oldpeak* dengan arah korelasi positif, sedangkan *thalch* memiliki arah korelasi negatif. Informasi dari analisis korelasi ini selanjutnya dapat digunakan sebagai salah satu pertimbangan dalam menentukan fitur yang relevan pada tahap *feature selection* sebelum proses pemodelan.

d. Encoding Fitur Kategorikal

Sebelum pemodelan, variabel kategorikal dikonversi ke bentuk numerik agar dapat dikenali oleh algoritma *machine learning*. Fitur biner seperti *sex, fbs*, dan *exang* ditransformasikan menggunakan *Label Encoding*. Sementara itu, *One Hot Encoding* diterapkan pada *cp* dan *restecg* karena keduanya mempunyai lebih dari dua kategori. Hasil transformasi tersebut menghasilkan representasi numerik yang tetap mempertahankan informasi kategori dan dapat digunakan sebagai masukan pada proses pelatihan model.

Fitur	Sebelum Encoding	Sesudah Encoding
sex (Jenis kelamin)	Male, Female	0 = Female, 1 = Male
fbs (Gula darah puasa)	True, False	0 = False, 1 = True
exang (Nyeri dada saat aktivitas)	True, False	0 = False, 1 = True

Tabel 2 Label Encoding

Tabel 2 memperlihatkan transformasi data kategorikal pada variabel *sex, fbs*, dan *exang* menggunakan *Label Encoding*. Pada *sex*, kategori *Female* direpresentasikan dengan angka 0 dan *Male* dengan angka 1. Sementara itu, nilai *False* dikodekan sebagai 0 dan *True* sebagai 1 pada fitur biner terkait. Transformasi tersebut menghasilkan data numerik yang dapat digunakan sebagai masukan dalam pemodelan *machine learning*.

Fitur	Sebelum Encoding	Sesudah Encoding
cp (Jenis nyeri dada)	asymptomatic, non-anginal, atypical angina, typical angina	cp_atypical angina, cp_non-anginal, cp_typical angina (asymptomatic sebagai referensi)
restecg (Hasil EKG)	normal, lv hypertrophy, st-t abnormality	restecg_normal, restecg_st-t abnormality (lv hypertrophy sebagai referensi)

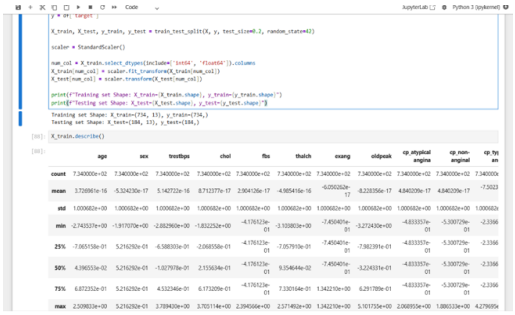
Tabel 3 One Hot Encoding

Tabel 3 menyajikan transformasi fitur *cp* dan *restecg* menggunakan metode *One Hot Encoding*. Masing-masing kategori diubah menjadi variabel biner bernilai 0 atau 1. Dalam proses tersebut, kategori *asymptomatic* pada fitur *cp* dan *lv hypertrophy* pada *restecg* ditetapkan sebagai kategori acuan. Penggunaan kategori referensi ini bertujuan untuk mencegah terjadinya *dummy variable trap* yang dapat memicu multikolinearitas pada data hasil transformasi.

e. Splitting Data

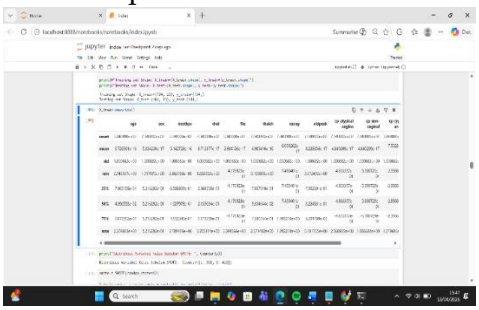
Dataset dipisahkan menjadi dua kelompok dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian model. Setelah pembagian dilakukan, variabel numerik ditransformasikan menggunakan *StandardScaler* berbasis Z-score. Standardisasi tersebut

menghasilkan fitur dengan nilai rata-rata di sekitar 0 dan standar deviasi mendekati 1. Dengan skala yang lebih seragam, perbedaan rentang nilai antarfitur dapat diminimalkan sehingga proses pembelajaran model dapat berlangsung lebih efektif.



Gambar 7 Hasil Proses Splitting Data

Gambar 7 memperlihatkan hasil pemisahan data dengan komposisi **80:20**, yaitu 734 sampel sebagai data latih (*X_train*) dan 184 sampel sebagai data uji (*X_test*), dengan masing-masing memuat 13 fitur. Jumlah label pada *y_train* dan *y_test* mengikuti pembagian tersebut, yakni 734 dan 184 data. Pembagian ini memungkinkan model mempelajari pola dari sebagian besar dataset, sedangkan data yang tidak dilibatkan selama pelatihan digunakan untuk mengukur kemampuan model dalam melakukan prediksi.



Gambar 8 Hasil Standarisasi Data

Gambar 8 memperlihatkan hasil transformasi fitur numerik menggunakan *StandardScaler*. Setelah proses tersebut, setiap fitur memiliki rata-rata yang berada di sekitar 0 dengan standar deviasi mendekati 1. Hasil ini mengindikasikan bahwa skala data telah berhasil diseragamkan, sehingga diharapkan dapat mendukung proses pembelajaran model dan menghasilkan kinerja prediksi yang lebih optimal.

f. *Synthetic Minority Over-sampling Technique* (SMOTE)

Teknik *Synthetic Minority Over-sampling Technique* (SMOTE) digunakan untuk memperbaiki perbedaan jumlah sampel antar kelas pada data pelatihan. Metode ini menambahkan sampel sintetis pada kelompok dengan jumlah data lebih sedikit hingga komposisi kelas menjadi lebih proporsional. Penyeimbangan tersebut dilakukan untuk mengurangi kecenderungan model dalam memprioritaskan kelas mayoritas selama proses klasifikasi.

Fitur	Sebelum SMOTE	Setelah SMOTE
0 =Hasil Prediksi Pasien Tidak Berisiko Terkena Penyakit Jantung	337	397
1 =Hasil Prediksi Pasien Berisiko Terkena Penyakit Jantung	397	397

Tabel 4 Distribusi Data Sebelum SMOTE

Tabel 4 memperlihatkan kondisi kelas sebelum dilakukan SMOTE, yaitu **337 sampel pada kelas 0** dan **397 sampel pada kelas 1**. Selisih jumlah tersebut mengindikasikan bahwa

distribusi data belum seimbang dan dapat menyebabkan model lebih cenderung mempelajari karakteristik kelas dengan jumlah sampel lebih besar.

Penerapan SMOTE menghasilkan komposisi data yang sama pada kedua kelas. Kondisi yang lebih seimbang ini diharapkan membuat model mampu mengenali pola dari masing-masing kelas secara lebih proporsional, sehingga kecenderungan prediksi terhadap kelas mayoritas dapat diminimalkan.

g. Feature Selection Menggunakan Embedded Methods

Seleksi fitur dilakukan melalui pendekatan *Embedded Methods* dengan menggunakan nilai *feature importance* yang dihasilkan oleh algoritma Random Forest. Nilai tersebut digunakan untuk mengukur besarnya kontribusi setiap variabel terhadap proses klasifikasi. Fitur kemudian diseleksi menggunakan batas *cumulative importance* ≤ 0,95, sehingga variabel yang memiliki kontribusi lebih besar dapat dipertahankan dan jumlah fitur dalam model dapat dikurangi.

No.	feature	importance	No	feature	importance
1.	chol	0.166412	8.	sex	0.050578
2.	thalch	0.147296	9.	cp_non-anginal	0.040060
3.	oldpeak	0.129319	10.	fbs	0.019746
4.	age	0.123868	11.	restecg_normal	0.018717
5.	exang	0.111822	12.	cp_typical angina	0.018110
6.	trestbps	0.090257	13.	restecg_st-t abnormality	0.015667
7.	cp_atypical angina	0.068147			

Tabel 6 Distribusi Data Setelah SMOTE

Tabel 6 menyajikan tingkat kontribusi masing-masing fitur berdasarkan nilai *feature importance*. Hasil pengukuran menempatkan *chol, thalch, oldpeak*, dan *age* sebagai variabel dengan kontribusi paling dominan terhadap klasifikasi. Sebaliknya, fitur dengan nilai kepentingan yang lebih rendah memberikan kontribusi yang relatif kecil terhadap keputusan model.

Penerapan batas *cumulative importance* ≤ 0,95 menghasilkan penyederhanaan fitur dari 13 menjadi 10 variabel. Fitur yang dipertahankan meliputi *chol, thalch, oldpeak, age, exang, trestbps, cp_atypical angina, sex, cp_non-anginal*, dan *fbs*. Pengurangan jumlah variabel tersebut bertujuan menghasilkan model yang lebih efisien dengan tetap mempertahankan informasi yang relevan untuk proses prediksi.

3. Hasil Pengembangan Model *Machine Learning*

a. Training Model

Data yang telah selesai dipersiapkan melalui *preprocessing*, pembagian dataset, penerapan SMOTE, dan seleksi fitur kemudian digunakan sebagai masukan dalam proses pelatihan. Eksperimen melibatkan enam metode klasifikasi, yaitu Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression, Random Forest, dan Decision Tree. Agar hasil evaluasi dapat dibandingkan secara konsisten, seluruh model dilatih menggunakan komposisi data dan kumpulan fitur yang sama.

Algoritma	Parameter
K-Nearest Neighbors (KNN)	n_neighbors=5, weights='distance'
Decision Tree	random_state=42, max_depth=None
Random Forest	n_estimators=300, random_state=42, n_jobs=-1
Support Vector Machine (SVM)	probability=True, random_state=42, kernel='rbf'
Naive Bayes	

Logistic Regression	max_iter=1000, solver='lbfgs'

Tabel 7 Training Model

Tabel 7 merangkum konfigurasi parameter yang diterapkan pada setiap algoritma selama pelatihan. KNN menggunakan n_neighbors=5, sedangkan Random Forest menerapkan n_estimators=300. Pada SVM digunakan *kernel* RBF, sementara Logistic Regression diatur dengan max_iter=1000. Decision Tree menggunakan random_state=42 dan max_depth=None, sedangkan Naive Bayes tetap menggunakan konfigurasi *default*. Penetapan konfigurasi tersebut digunakan untuk memperoleh kinerja model yang stabil selama proses pelatihan dan evaluasi.

***b.* Evaluasi Model**

Kinerja setiap model klasifikasi diukur menggunakan lima indikator, yaitu *accuracy, precision, recall, F1-score*, dan *ROC-AUC*. Kombinasi metrik tersebut digunakan untuk menilai performa model dari berbagai aspek sehingga kemampuan dalam membedakan pasien berisiko dan tidak berisiko penyakit jantung dapat dievaluasi secara lebih menyeluruh.

Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*	*ROC-AUC*
K-Nearest Neighbors (KNN)	0.815217	0.859813	0.828829	0.844037	0.852277
Decision Tree (DT)	0.739130	0.818182	0.729730	0.771429	0.741577
Random Forest (RF)	0.798913	0.849057	0.810811	0.829493	0.865729
Support Vector Machine (SVM)	0.793478	0.847619	0.801802	0.824074	0.854622
Naive Bayes (NB)	0.809783	0.845455	0.837838	0.841629	0.877700
Logistic Regression (LR)	0.820652	0.854545	0.846847	0.850679	0.872640

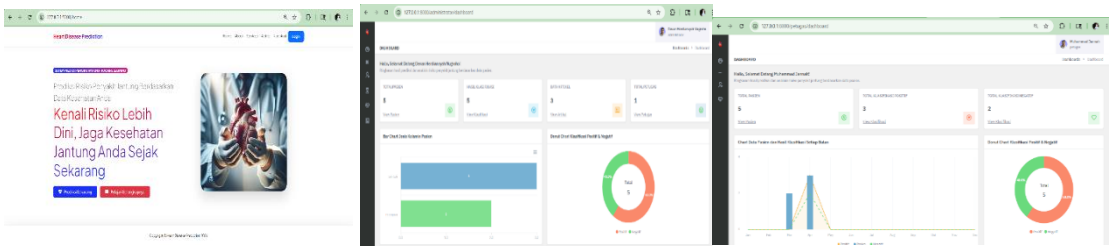
Skenario	*Accuracy*	*Precision*	*Recall*	*F1-Score*	*ROC-AUC*
Baseline (Tanpa SMOTE & Tanpa *Feature Selection*)	0.798913	0.836364	0.828829	0.832579	0.868567
Tanpa SMOTE, pakai *Feature Selection*	0.809783	0.833333	0.855856	0.844444	0.866839
Pakai SMOTE, tanpa *Feature Selection*	0.804348	0.844037	0.828829	0.836364	0.867580
Model Final (Pakai SMOTE, Pakai *Feature Selection*)	0.820652	0.854545	0.846847	0.850679	0.872640

Gambar 9 Evaluasi Model Sebelum dan sesudah

Berdasarkan hasil evaluasi pada Gambar 9, Logistic Regression menunjukkan kinerja paling unggul dibandingkan model lainnya. Algoritma ini menghasilkan *accuracy* 82,07%, *precision* 85,45%, *recall* 84,68%, *F1-Score* 85,07%, dan *ROC-AUC* 87,26%. Capaian yang relatif seimbang pada seluruh indikator evaluasi tersebut menjadi dasar pemilihan Logistic Regression sebagai model akhir untuk klasifikasi risiko penyakit jantung**.**

***c.* Hasil Pengembangan Sistem**

Gambar berikut menyajikan hasil implementasi antarmuka aplikasi yang terdiri atas *landing page*, *dashboard* admin, dan *dashboard* petugas. *Landing page* berfungsi sebagai halaman awal ketika pengguna membuka aplikasi. Sementara itu, *dashboard* admin digunakan untuk mengawasi aktivitas sistem secara keseluruhan. Pada sisi petugas, *dashboard* menyediakan informasi terkait aktivitas operasional, termasuk jumlah klasifikasi yang telah dilakukan dan total data pasien yang telah dimasukkan ke dalam sistem.



Gambar 10. Halaman *Home - Landing Page, dashboad admin dan dashboard petugas*

Kesimpulan

Penelitian ini menghasilkan sistem berbasis *machine learning* untuk mengidentifikasi risiko penyakit jantung dengan membandingkan enam metode klasifikasi, yaitu K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM), Naive Bayes, dan Logistic Regression. Berdasarkan pengujian, Logistic Regression menunjukkan kinerja paling baik dengan *accuracy* **82,06%**, *precision* **85,45%**, *recall* **84,68%**, *F1-Score* **85,06%**, dan *ROC-AUC* **87,26%**. Atas dasar hasil tersebut, algoritma ini ditetapkan sebagai model akhir.

Ketidakseimbangan kelas pada data ditangani menggunakan *Synthetic Minority Over-sampling Technique* (SMOTE), sedangkan *Embedded Methods* digunakan untuk menyaring fitur berdasarkan tingkat relevansinya. Pengujian konfigurasi melalui GridSearchCV menunjukkan bahwa proses *hyperparameter tuning* tidak memberikan peningkatan kinerja dibandingkan model awal. Oleh sebab itu, Logistic Regression sebelum *tuning* dipertahankan sebagai model terbaik. Model tersebut selanjutnya diintegrasikan dengan Laravel dan Flask API ke dalam aplikasi berbasis web sehingga data klinis pasien dapat diproses untuk menghasilkan klasifikasi risiko penyakit jantung secara *real-time*.

Daftar Pustaka

Al-Zewairi, M., & Qaddoura, A. (2025). Modeling and Design of Complex Systems Using UML Activity Diagrams. *International Journal of Computer Science and Software Engineering, 11(3), 45-53.*

Alemerien, K. (2024). *Diagnosing Cardiovascular Diseases using Optimized Machine Learning Algorithms with GridSearchCV*. *5*(4), 1539–1552.

Alodokter, C., Berdebar, J., & Penyebab, K. (2026). *Artikel Terkait Dokter Terkait Chat lebih dari 1 . 000 dokter di Aplikasi Alodokter ! Respons Cepat , Jawaban Akurat ! Alodokter*. 2026.

Alvin Rajkomar, Jeffrey Dean, I. K. (2024). *Machine Learning in Medicine*. 1347–1358. https://doi.org/10.1056/NEJMra1814259

Anita, M., Grecea, I., Yulianti, D., Pasaribu, S. V., Studi, P., Informatika, T., Widya, U., Pontianak, D., Machine, S. V., Bayes, N., Tree, D., & Machine, S. V. (2025). *KLASIFIKASI FAKTOR RISIKO PENYAKIT JANTUNG MENGGUNAKAN MACHINE*. *16*(c), 68–78.

Dixon, D. L., Harm, P. D., & Taqueti, V. R. (2023). 2023AHA / ACC / ACCP / ASPC / NLA / PCNA Guideline for the Management of Patients With Chronic Coronary Disease. *Journal of the American College of Cardiology*, *82*(9), 833–955. https://doi.org/10.1016/j.jacc.2023.04.003

Hidayat, R., Sy, Y. S., Sujana, T., Husnah, M., & Saputra, H. T. (2024). *Implementasi Machine Learning Untuk Prediksi Penyakit Jantung Menggunakan Algoritma Support Vector Machine*. *5*(2), 161–168.

Larassati, D., Zaidiah, A., & Afrizal, S. (2022). *Sistem prediksi penyakit jantung koroner menggunakan metode naïve bayes*. *07*, 533–546.

Merceedi, K. J., & Abdulazeez, A. M. (2025). *Feature Selection Methods of Gene Expression Based on Machine Learning : A Review*. *5*(1), 105–138.

Nugraha, W. (2023). Prediksi penyakit jantung cardiovascular menggunakan model algoritma klasifikasi. *Jurnal SIGMATA, 9*(2), 78–84.

Pringsewu, U. A. (2025). *Volume 6 Issue 1 Aisyah Journal of Informatics and Electrical Engineering Aisyah Journal of Informatics and Electrical Engineering Aisyah Journal of Informatics and Electrical Engineering*. *6*(1), 145–150.

Pulungan, M. P., Purnomo, A., Kurniasih, A., Korespondensi, P., Class, I., & Technique, S. M. O. (2024). *Penerapan Smote Untuk Mengatasi Imbalance Class Dalam Klasifikasi Kepribadian Mbti Menggunakan Naive Bayes Application Of Smote To Overcome Class Imbalance In The Mbti Personality Classification Using The Naïve Bayes Classifier*. *11*(5), 1033–1042. https://doi.org/10.25126/jtiik.2024117989

Raras, C., Widiawati, A., Nurazizah, L., & Yunita, I. R. (2024). *Implementasi Algoritma Logistic Regression pada Pembuatan Website Sederhana untuk Prediksi Penyakit Jantung*. *15*(01), 117–122. https://doi.org/10.35970/infotekmesin.v15i1.2048

Riani, A., Susianto, Y., Rahman, N., & Ali, U. D. (2025). *Implementasi Data Mining Untuk Memprediksi Penyakit Jantung Mengunakan Metode Naive Bayes Data Mining Implementation to Predict Heart Disease using Naive Bayes Method*. *1*(01), 25–34. https://doi.org/10.35970/jinita.v1i01.64

Sadikin, B. G. (2025). Menkes : Stroke dan Jantung Kematian per Tahun di Indonesia. *Https://Nasional.Kompas.Com/Read/2025/05/17/16104461/Menkes-300000-Orang-Meninggal-Dunia-Karena-Stroke-per-Tahun*.

Silalahi, S. M. M. (2024). *Analisis Faktor Risiko Penyakit Jantung Pada Penduduk Usia Produktif Di Indonesia (Analisis Data SKI 2023)*. 61–62.

Siswanto, E., Kom, S., Kom, M., Siswanto, E., Kom, S., & Kom, M. (2025). *Kupas Tuntas Pemprograman PHP*.

Wibowo, A. C., Lestari, S. A., Informasi, S., Komputer, F. I., Duta, U., & Surakarta, B. (2024). *Analisis Penggunaan Machine Learning Dalam Klasifikasi Penentuan Penyakit Jantung*. *9*(2), 9–13.

Ying, X. (2023). *An Overview of Overfitting and its Solutions An Overview of Overfitting and its Solutions*. https://doi.org/10.1088/1742-6596/1168/2/022022