

Implementasi Naïve Bayes dalam Flask Framework untuk Sistem Informasi Klasifikasi Penyakit Jantung

Akas Bagus Setiawan¹, Dzakiyyan Nasyatha Adlin², Mas'ud Hermansyah³

^{1,2}Manajemen Informatika, Politeknik Negeri Jember

³Bisnis Digital, Politeknik Negeri Jember

Email: ¹akabagus_s@polije.ac.id, ²dzakiyyanyannas@gmail.com, ³mas_udhermansyah@polije.ac.id

Email Penulis Korespondensi: akabagus_s@polije.ac.id

Abstrak

Penyakit jantung merupakan salah satu penyebab utama kematian di Indonesia, dengan tren kasus yang terus meningkat setiap tahunnya. Deteksi dini sangat penting untuk mencegah risiko yang lebih parah, namun keterbatasan akses terhadap layanan kesehatan menjadi kendala di beberapa wilayah. Penelitian ini bertujuan untuk membangun sistem klasifikasi penyakit jantung berbasis algoritma *Naive Bayes* yang diimplementasikan dalam API menggunakan *Flask Framework Python* sebagai *backend* cerdas. *Dataset* yang digunakan diperoleh dari *Kaggle* dan telah melalui tahapan *preprocessing*, seleksi atribut, transformasi data, serta pembagian data latih dan uji. Model *Naive Bayes* dipilih karena kesederhanaannya serta kemampuannya dalam menangani data berskala besar. Evaluasi model menunjukkan performa yang cukup baik, dengan *accuracy* mencapai 73,77%, *precision* 67,57%, dan *recall* 86,21%. Sistem yang dikembangkan diintegrasikan dalam layanan *web* dan *mobile*, sehingga dapat diakses oleh pengguna secara luas. Hasil penelitian ini menunjukkan bahwa integrasi algoritma *machine learning* dan API dapat memberikan solusi deteksi dini penyakit jantung yang ringan, cepat, dan mudah digunakan. Penelitian ini diharapkan dapat menjadi dasar pengembangan sistem pendukung keputusan dalam bidang kesehatan digital.

Kata Kunci: *Naive Bayes*, penyakit jantung, *Flask API*, klasifikasi, *Dataset*

Abstract

Heart disease is one of the leading causes of death in Indonesia, with cases continuing to increase every year. Early detection is essential to prevent more severe risks, but limited access to health services is an obstacle in some areas. This study aims to build a heart disease classification system based on the *Naive Bayes* algorithm implemented in an API using the *Python Flask Framework* as an intelligent backend. The dataset used was obtained from *Kaggle* and has gone through the stages of preprocessing, attribute selection, data transformation, and division of training and test data. The *Naive Bayes* model was chosen because of its simplicity and ability to handle large-scale data. The model evaluation showed quite good performance, with an accuracy of 73.77%, precision of 67.57%, and recall of 86.21%. The developed system is integrated into web and mobile services, so that it can be accessed by users widely. The results of this study indicate that the integration of machine learning and API can provide a lightweight, fast, and easy-to-use early detection solution for heart disease. This research is expected to be the basis for the development of decision support systems in the field of digital health.

Keywords: *Naive Bayes*, heart disease, *Flask API*, classification, *Dataset*

1. PENDAHULUAN

Penyakit jantung masih menjadi salah satu penyebab utama kematian di Indonesia. Berdasarkan data *Global Burden of Disease* dan *Institute for Health Metrics and Evaluation (IHME)* tahun 2014–2019, penyakit ini tercatat sebagai penyumbang kematian tertinggi secara nasional. Data *Riskesmas* tahun 2013 dan 2018 juga menunjukkan tren peningkatan prevalensi penyakit jantung, dari 0,5% menjadi 1,5% dalam lima tahun. Tak hanya berdampak pada kesehatan, penyakit jantung juga menjadi beban ekonomi terbesar bagi negara, dengan pembiayaan mencapai Rp7,7 triliun menurut data BPJS Kesehatan tahun 2021. Kondisi ini menimbulkan kebutuhan mendesak akan sistem yang dapat memberikan informasi dan klasifikasi dini terhadap penyakit jantung sehingga dapat membantu deteksi dan penanganan lebih awal [1].

Akan tetapi, masih banyak masyarakat yang tidak memiliki akses terhadap sistem pendukung keputusan (*decision support system*) yang cepat, akurat, dan mudah diakses, baik melalui *web* maupun perangkat *mobile*. Di sisi lain, pengembangan sistem berbasis kecerdasan buatan untuk klasifikasi penyakit jantung seringkali terbatas pada platform *desktop* atau tidak terintegrasi dengan baik ke dalam sistem berbasis *web* dan *mobile*. Hal ini menimbulkan kebutuhan akan solusi *backend* cerdas yang ringan, mudah di-deploy, dan dapat diakses secara luas. Penelitian ini mengusulkan pembangunan sebuah sistem *backend* cerdas menggunakan algoritma *Naive Bayes* yang diimplementasikan dalam *framework Flask*, dan disediakan dalam bentuk API (*Application Programming Interface*) yang dapat digunakan oleh aplikasi *frontend* berbasis *web* dan *mobile*. Sistem ini akan memungkinkan klasifikasi penyakit jantung secara *real-time* berdasarkan input data pasien, dan diakses secara luas oleh pengguna dari berbagai perangkat.

Naive Bayes adalah salah satu algoritma klasifikasi yang sederhana namun efektif, terutama dalam konteks prediksi medis [2]. Studi-studi sebelumnya [3], [4], [5] telah menunjukkan bahwa *Naive Bayes* mampu memberikan akurasi yang

kompetitif dalam klasifikasi data kesehatan, termasuk penyakit jantung. *Flask*, sebagai *microframework Python*, telah menjadi pilihan populer dalam pengembangan backend API karena fleksibilitas dan performanya yang ringan [6]. Namun, implementasi terintegrasi antara *Naive Bayes*, *Flask*, dan platform *web-mobile* untuk klasifikasi penyakit jantung masih jarang ditemukan dalam literatur.

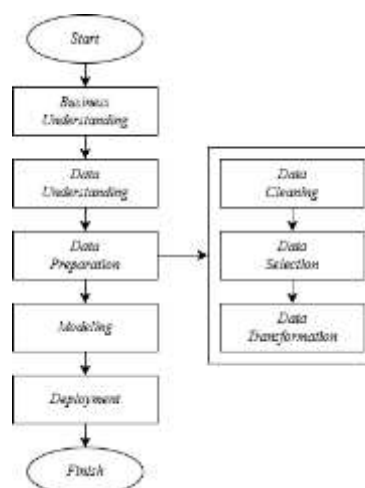
Beberapa penelitian sebelumnya telah menunjukkan potensi algoritma *Naive Bayes* dalam klasifikasi penyakit jantung. Penelitian oleh [7], [8] melakukan analisis performa beberapa algoritma, termasuk *Naive Bayes*, dalam memprediksi penyakit jantung, dan menemukan bahwa algoritma ini cukup kompetitif dari segi akurasi. Namun, penelitian mereka hanya sebatas evaluasi model tanpa pengembangan sistem implementatif yang dapat digunakan langsung oleh pengguna. Sementara itu, [9] mengembangkan model prediksi penyakit jantung berbasis *machine learning* menggunakan *Python*, namun hasilnya hanya terbatas pada lingkungan lokal, tanpa adanya integrasi dengan *framework web* atau *API*. Penelitian lain oleh [10] menggabungkan teknologi *IoT (Internet of Things)* dan pembelajaran mesin untuk sistem monitoring kesehatan. Meskipun inovatif, penelitian ini tidak secara spesifik membahas klasifikasi penyakit jantung menggunakan *Naive Bayes* atau pengembangan backend cerdas berbasis *Flask*. Selanjutnya, [11] mengembangkan sistem prediksi penyakit jantung berbasis *web* menggunakan algoritma *machine learning*. Meskipun cukup relevan, penelitian tersebut tidak mendalami aspek pengembangan API modular yang dapat diakses oleh berbagai platform, khususnya *mobile*. Penerapan *Python* dalam pengembangan sistem juga dilakukan oleh penelitian [12] menggunakan *Spyder* dan *Flask* yang merupakan lingkungan pengembangan terpadu sumber terbuka yang dirancang khusus untuk pemrograman ilmiah dalam bahasa *Python* untuk membangun antarmuka web bagi sistem prediksi penyakit jantung, tetapi implementasinya masih bersifat monolitik dan belum menyediakan backend dalam bentuk API yang dapat digunakan ulang oleh aplikasi lain.

Berangkat dari keterbatasan-keterbatasan penelitian sebelumnya, penelitian ini hadir dengan pendekatan yang lebih modular dan aplikatif. Fokus utama penelitian ini adalah pada implementasi algoritma *Naive Bayes* ke dalam API menggunakan *Flask Framework Python* sebagai backend cerdas yang dapat dikonsumsi oleh aplikasi berbasis *web* dan *mobile*. Penelitian ini menggunakan data penyakit jantung yang bersumber dari platform *Kaggle*, yang telah diproses dan disesuaikan untuk klasifikasi. Inovasi dari penelitian ini terletak pada perancangan sistem backend yang tidak hanya cerdas dan efisien, tetapi juga bersifat *reusable* dan *cross-platform*, memungkinkan akses klasifikasi data secara *real-time* dari berbagai jenis aplikasi *frontend*. Dengan pendekatan ini, penelitian ini menawarkan solusi konkret dan inovatif dalam sistem pendukung keputusan medis berbasis teknologi informasi yang mudah diintegrasikan dan dikembangkan lebih lanjut.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan *CRISP-DM (Cross-Industry Standard Process for Data Mining)* sebagai metodologi utama dalam proses pengembangan sistem klasifikasi penyakit jantung. *CRISP-DM* terdiri dari enam tahapan utama yang dilakukan secara iteratif dan fleksibel [13]. Berikut adalah alur penelitian yang dilakukan oleh peneliti seperti terlihat pada Gambar 1.



Gambar 1. *CRISP-DM (Cross-Industry Standard Process for Data Mining)*

CRISP-DM terdiri dari enam tahapan utama [14]:

- Business Understanding*
Memahami tujuan dan kebutuhan bisnis secara menyeluruh serta merumuskan masalah yang akan diselesaikan melalui *data mining*.
- Data Understanding*

Mengumpulkan data awal dan melakukan eksplorasi awal terhadap data untuk memahami struktur, kualitas, dan informasi penting di dalamnya.

c. *Data Preparation*

Menyiapkan data agar sesuai untuk pemodelan, termasuk pembersihan data, transformasi atribut, penggabungan data, dan seleksi atribut.

d. *Modeling*

Membangun model *machine learning* menggunakan algoritma yang sesuai, seperti *Naive Bayes*, *Decision Tree*, *SVM*, dll. Pada tahap ini, juga dilakukan pengujian dan tuning parameter model.

e. *Deployment*

Mengimplementasikan model ke dalam sistem nyata, misalnya dalam bentuk *API*, aplikasi *web*, atau *dashboard*, agar dapat digunakan oleh pengguna akhir.

2.2 Business Understanding

Pada tahap ini, permasalahan utama yang diangkat adalah tingginya prevalensi penyakit jantung di Indonesia serta perlunya sistem pendukung keputusan yang cepat, ringan, dan mudah diakses oleh masyarakat melalui perangkat *web* maupun *mobile*. Tujuan dari penelitian ini adalah membangun sistem klasifikasi penyakit jantung yang akurat, berbasis *Naive Bayes*, dan diimplementasikan ke dalam *backend* cerdas menggunakan *Flask Framework* dalam bahasa *Python*. *Backend* ini dirancang agar dapat diakses oleh berbagai *platform frontend* sebagai bagian dari sistem monitoring atau diagnosis awal penyakit jantung.

2.3 Data Understanding

Dataset yang digunakan dalam penelitian ini diambil dari *platform Kaggle* dengan nama dataset *Heart Disease UCI*, yang berisi atribut-atribut penting seperti usia, jenis kelamin, tekanan darah, kolesterol, detak jantung maksimum, serta status target (terindikasi penyakit jantung atau tidak). Tahapan ini mencakup eksplorasi data, identifikasi atribut prediktor, distribusi nilai, serta hubungan antaratribut untuk memahami karakteristik data lebih dalam.

2.4 Data Preparation

Tahap ini mencakup proses *preprocessing* data yang bertujuan untuk mempersiapkan data mentah menjadi bentuk yang sesuai dan optimal untuk pelatihan model klasifikasi. Langkah awal yang dilakukan adalah pembersihan data, yaitu dengan menangani nilai-nilai kosong atau tidak valid yang dapat memengaruhi akurasi model. Selanjutnya, dilakukan proses konversi atribut kategorikal menjadi bentuk numerik melalui teknik *encoding* agar dapat dikenali oleh algoritma *machine learning*. Selain itu, data juga dinormalisasi atau dilakukan proses *scaling* untuk menyamakan skala antar atribut sehingga dapat meningkatkan performa model. Tahap akhir dalam proses persiapan data adalah membagi dataset menjadi dua bagian, yaitu data latih dan data uji, yang umumnya menggunakan proporsi 80:20. Data yang telah melalui seluruh tahapan *preprocessing* ini kemudian digunakan sebagai input dalam proses pelatihan model *Naive Bayes* untuk klasifikasi penyakit jantung.

2.5 Modelling

Pada tahap pemodelan, dilakukan evaluasi perbandingan terhadap beberapa algoritma *data mining* guna mengidentifikasi algoritma dengan tingkat akurasi terbaik menggunakan *dataset* yang sama. Setelah algoritma dengan performa paling optimal ditemukan, algoritma tersebut kemudian diterapkan sebagai metode klasifikasi untuk memprediksi risiko penyakit jantung berdasarkan atribut-atribut yang telah ditentukan sebelumnya. Model ini dibangun menggunakan pustaka *Python* seperti *scikit-learn* yang menyediakan fungsi-fungsi praktis dalam proses pelatihan dan evaluasi model. Setelah model dilatih dengan data latih, langkah selanjutnya adalah melakukan validasi dan evaluasi performa model menggunakan beberapa metrik evaluasi, yaitu *accuracy*, *precision*, dan *recall*. Evaluasi ini penting untuk mengukur kemampuan model dalam membedakan antara individu yang memiliki risiko penyakit jantung dan yang tidak. Tahapan ini bertujuan untuk memastikan bahwa model yang dihasilkan memiliki performa yang andal dan akurat sebelum diimplementasikan lebih lanjut ke dalam sistem aplikasi.

2.6 Deployment

Tahap akhir dari penelitian adalah implementasi model ke dalam *API* menggunakan *Flask Framework Python*. Model disimpan dalam format *.pkl (pickle)* dan di-load oleh *backend Flask* untuk menyajikan *endpoint API*. *Backend* ini disiapkan untuk diintegrasikan ke aplikasi berbasis *web* dan *mobile*, sehingga pengguna dapat memasukkan data parameter kesehatan dan langsung mendapatkan hasil klasifikasi. Sistem ini memungkinkan integrasi lintas platform dan memberikan layanan klasifikasi secara *real-time*.

3. HASIL DAN PEMBAHASAN

3.1 Dataset

Dataset yang digunakan dalam penelitian ini diperoleh dari situs *Kaggle*, dengan nama "*Heart Disease UCI*". *Dataset* ini berisi 1.025 data pasien dengan 13 atribut sebagai fitur dan 1 atribut sebagai target untuk melakukan klasifikasi. Atribut tersebut adalah *age*, *sex*, *Chest pain Type (Cp)*, *Resting blood pressure (Testbps)*, *Serum Cholesterol (Chol)*, *Fasting Blood Sugar (Fbs)*, *Resting electrocardiographic result (Restecg)*, *Maximum Heart Rate (Thalach)*, *Does the patient experience angina during physical activity? (Exang)*, *ST depression due exercise (Oldpeak)*, *The slope of the peak exercise ST (Slope)*, *Number of major vessels (Ca)*, *The patient's thalassemia category (Thal)*, and *Cardiac Disease (Target)*. Ringkasan atribut *dataset* dapat dilihat pada Tabel 1.

Tabel 1. Atribut dan Tipe Data *Dataset*

Kode	Atribut	Tipe Data
a1	<i>Age</i>	<i>int</i>
a2	<i>Sex</i>	<i>int</i>
a3	<i>Chest pain Type (Cp)</i>	<i>int</i>
a4	<i>Resting blood pressure (Testbps)</i>	<i>int</i>
a5	<i>Serum Cholesterol (Chol)</i>	<i>int</i>
a6	<i>Fasting Blood Sugar (Fbs)</i>	<i>int</i>
a7	<i>Resting electrocardiographic result (Restecg)</i>	<i>int</i>
a8	<i>Maximum Heart Rate (Thalach)</i>	<i>int</i>
a9	<i>Does the patient experience angina during physical activity? (Exang)</i>	<i>int</i>
a10	<i>ST depression due exercise (Oldpeak)</i>	<i>float</i>
a11	<i>The slope of the peak exercise ST (Slope)</i>	<i>int</i>
a12	<i>Number of major vessels (Ca)</i>	<i>int</i>
a13	<i>The patient's thalassemia category (Thal)</i>	<i>int</i>
a14	<i>Cardiac Disease (Target)</i>	<i>int</i>

3.2 Data Processing

Tahapan ini mencakup seluruh proses pengolahan data sebelum digunakan dalam pelatihan model. Langkah-langkah yang dilakukan meliputi pembersihan data, seleksi atribut, transformasi data numerik dan kategorikal. Rincian *dataset* awal dapat dilihat pada Tabel 2.

Tabel 2. *Dataset* Awal

No.	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14
1.	52	1	0	125	212	0	1	168	0	1	2	2	3	0
2.	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
3.	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
4.	61	1	0	148	203	0	1	161	0	0	2	1	3	0
5.	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1.024	50	0	0	110	254	0	0	159	0	0	2	0	2	1
1.025	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

3.2.1 Data Cleaning

Tahapan pembersihan data mencakup identifikasi nilai yang hilang (*missing values*) serta pendeteksian data yang tidak valid atau tidak sesuai. Selain itu, proses ini juga melibatkan penghapusan data duplikat dengan menyisakan satu entri dari setiap pasangan yang teridentifikasi. Setelah dilakukan pembersihan, ditemukan sebanyak 723 data yang merupakan duplikasi dan telah dihapus agar data siap untuk diproses ke tahap selanjutnya.

3.2.2 Data Selection

Pemanfaatan atribut dalam pelatihan model dilakukan secara selektif, tidak semua atribut dalam *dataset* digunakan dalam proses pelatihan model, melainkan hanya atribut yang memiliki relevansi dan kontribusi signifikan terhadap proses klasifikasi penyakit jantung yang dipertahankan. Atribut-atribut penting yang digunakan antara lain *age* (usia), *sex* (jenis kelamin), *cp* (jenis nyeri dada), *restbps* (tekanan darah saat istirahat), *chol* (kadar kolesterol), *thalach* (detak jantung maksimum), *exang* (angina yang diinduksi oleh olahraga), dan *cardiac disease* (target). Beberapa atribut dalam *dataset* klasifikasi penyakit jantung dihilangkan karena pertimbangan statistik dan relevansi informasi terhadap model prediktif, diantaranya:

- Atribut *fbs* (*Fasting Blood Sugar* > 120 mg/dl) dihapus karena memiliki distribusi yang sangat tidak seimbang, di mana sebagian besar data bernilai 0, sehingga kontribusinya terhadap akurasi model sangat rendah. Selain itu, korelasinya dengan variabel target sering kali tidak signifikan dan dianggap redundan secara statistik.
- Atribut *restecg* (hasil *elektrokardiografi* saat istirahat) juga dieliminasi karena merupakan data kategorikal dengan variasi nilai yang terbatas (0, 1, 2), serta memiliki kemungkinan informasi yang tumpang tindih dengan atribut lain seperti *cp* (nyeri dada) atau *thalach* (denyut jantung maksimum).
- Atribut *oldpeak* (penurunan segmen ST akibat olahraga) juga tidak digunakan karena sering menunjukkan distribusi yang tidak normal atau memiliki nilai pencilan ekstrem (*outlier*) yang berpotensi menurunkan performa model.
- Atribut *slope*, yang merepresentasikan kemiringan segmen ST, juga dianggap kurang informatif karena kecenderungannya bersifat redundan terhadap variabel lain seperti *oldpeak* dan *thalach*, serta kontribusi statistiknya yang rendah.
- Atribut *ca* (jumlah pembuluh darah utama yang terlihat melalui *fluoroskopi*) dikeluarkan dari proses pemodelan karena sering ditemukan nilai kosong atau tidak valid, serta kurangnya variasi data yang mengakibatkan minimnya relevansi.
- Atribut *thal* (*thalassemia*) juga tidak dimasukkan dalam pelatihan model karena bersifat kategorikal ordinal, dan jika tidak ditangani dengan teknik *encoding* yang tepat, dapat menyebabkan bias atau kesalahan interpretasi pada model. Selain itu, atribut ini kerap menunjukkan distribusi kelas yang tidak seimbang dan potensi *multikolinearitas* dengan atribut lain seperti *slope* atau *exang*, sehingga dipertimbangkan untuk dihilangkan dalam proses seleksi fitur.

Setelah dilakukan proses seleksi atribut dan pembersihan data, diperoleh 8 atribut dan sebanyak 302 data pasien yang siap digunakan dalam tahap pemodelan. Data yang terseleksi ini diharapkan mampu meningkatkan akurasi dan efisiensi dalam proses pelatihan serta pengujian model. Seluruh *dataset* digunakan dalam pemodelan ini karena setiap atribut dianggap saling berhubungan dan memiliki tingkat relevansi yang signifikan terhadap hasil klasifikasi. Setelah dilakukan proses seleksi atribut dan pembersihan data, diperoleh sebanyak 302 data bersih dan valid yang siap digunakan dalam tahap pemodelan. Data yang terseleksi ini diharapkan mampu meningkatkan akurasi dan efisiensi dalam proses pelatihan serta pengujian model.

3.2.3 Data Transformation

Pada tahap *data transformation*, sejumlah atribut diubah ke dalam format yang sesuai agar dapat diolah secara optimal oleh algoritma *Naïve Bayes*. Atribut *sex* yang awalnya bersifat kategorikal (*male/female*) diubah menjadi format *numerik biner*, yaitu 0 untuk perempuan dan 1 untuk laki-laki. Selanjutnya, atribut *chest pain type* (*cp*) yang memiliki empat kategori dikonversi menjadi angka, yakni 0 untuk *asymptomatic*, 1 untuk *typical angina*, 2 untuk *atypical angina*, dan 3 untuk *non-anginal pain*. Selanjutnya, atribut *exercise-induced angina* (*exang*) yang semula berupa nilai *boolean* (*true/false*) diubah menjadi 1 untuk “ya” dan 0 untuk “tidak”. Transformasi ini dilakukan untuk memastikan setiap atribut memiliki format yang seragam dan kompatibel dengan model klasifikasi yang digunakan. Rincian *dataset* akhir dapat dilihat pada Tabel 3.

Tabel 3. *Dataset* Akhir untuk Pemodelan

No.	a1	a2	a3	a4	a5	a8	a9	a14
1.	52	1	0	125	212	168	0	0
2.	53	1	0	140	203	155	1	0
3.	70	1	0	145	174	125	1	0
4.	61	1	0	148	203	161	0	0
5.	62	0	0	138	294	106	0	0
...	...	...	...	...	...	...	...	...
301	51	1	2	94	227	154	1	1
302	55	0	1	132	342	166	0	1

3.3 Pemodelan Algoritma

Pada tahap pemodelan algoritma, dilakukan proses evaluasi dan perbandingan kinerja beberapa algoritma klasifikasi yang umum digunakan dalam bidang *data mining*. Algoritma yang dibandingkan meliputi *Naive Bayes*, *Logistic Regression*, *Decision Tree*, *Random Forest*, *Support Vector Machine*, dan *K-Nearest Neighbors*. Setiap model diuji menggunakan *dataset* yang sama, dan kinerjanya dianalisis berdasarkan metrik evaluasi seperti *accuracy*, *precision*, dan *recall*. Hasil pengujian menunjukkan bahwa algoritma *Naive Bayes* memberikan performa yang paling baik dalam mengklasifikasikan data penyakit jantung. Berdasarkan hasil evaluasi tersebut, *Naive Bayes* dipilih sebagai algoritma

utama yang akan digunakan pada tahap implementasi sistem klasifikasi berbasis *web* dan *mobile*. Pemilihan ini didasarkan pada pertimbangan efisiensi, kemudahan integrasi dengan sistem *backend*, serta relevansi model terhadap karakteristik *dataset* yang digunakan. Hasil evaluasi performa pemodelan dari tiap algoritma dapat dilihat secara rinci pada Tabel 4.

Tabel 4. Perbandingan Pemodelan Algoritma

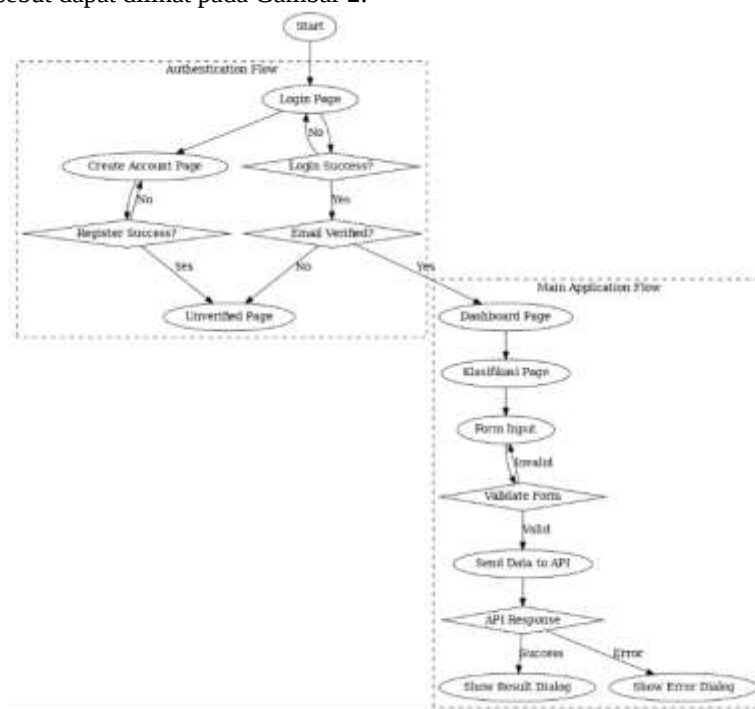
Algoritma	Accuracy	Precision	Recall
Naive Bayes	73.77%	67.57%	86.21%
Logistic Regression	72.13%	68.75%	75.86%
Decision Tree	63.93%	62.96%	58.62%
Random Forest	70.49%	66.67%	75.86%
Support Vector Machine	60.66%	55.81%	82.76%
K-Nearest Neighbors	63.93%	58.14%	86.21%

3.4 Rancangan Sistem

Perancangan sistem merupakan tahapan penting dalam proses pengembangan aplikasi, karena berfungsi sebagai dasar dalam membangun struktur dan fungsionalitas sistem yang sesuai dengan kebutuhan. Dalam penelitian ini, perancangan dilakukan dengan menggunakan beberapa komponen utama untuk menggambarkan alur kerja sistem, interaksi pengguna, struktur data, serta konfigurasi teknis dari sistem yang akan diimplementasikan. Adapun komponen perancangan sistem yang digunakan meliputi *Flowchart System*, *Use Case Diagram*, *Entity Relationship Diagram (ERD)*, dan *Arsitektur Sistem*.

3.4.1 Flowchart System

Flowchart digunakan untuk menggambarkan alur logika proses secara sistematis mulai dari *input data* oleh pengguna hingga hasil klasifikasi yang dihasilkan oleh sistem. Diagram ini membantu dalam memahami aliran data dan tahapan proses klasifikasi penyakit jantung yang dilakukan oleh model *Naive Bayes* di *backend Flask*. Penjelasan lebih lanjut mengenai *flowchart* tersebut dapat dilihat pada Gambar 2.



Gambar 2. Flowchart System

Flowchart ini menggambarkan alur proses sistem yang terbagi menjadi dua bagian utama, yaitu *Authentication Flow* dan *Main Application Flow*.

a. Authentication Flow

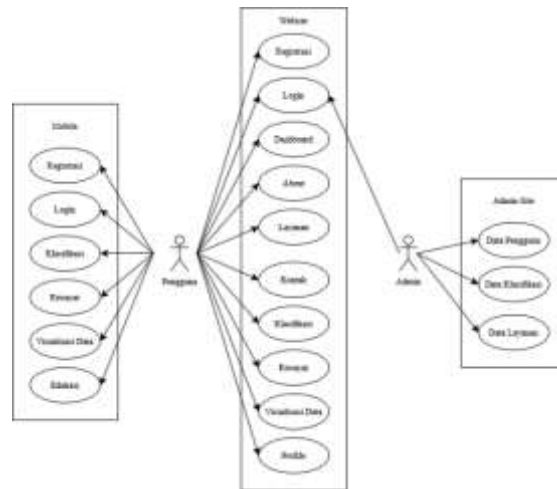
Proses diawali dari halaman login. Jika pengguna belum memiliki akun, mereka akan diarahkan ke halaman pembuatan akun. Setelah registrasi berhasil, pengguna akan masuk ke halaman *Unverified* jika *email* belum diverifikasi. Jika sudah memiliki akun dan berhasil *login*, sistem akan memeriksa status verifikasi *email*. Pengguna hanya bisa melanjutkan ke aplikasi utama jika *email* sudah diverifikasi.

b. Main Application Flow

Setelah proses otentikasi dan verifikasi berhasil, pengguna diarahkan ke halaman *Dashboard*, lalu menuju halaman *Klasifikasi*. Di sini pengguna mengisi *form input* yang kemudian divalidasi. Jika data tidak valid, pengguna kembali ke *form*. Jika valid, data dikirim ke *API*. Berdasarkan respons *API*, jika berhasil akan ditampilkan *Result Dialog*, jika gagal akan muncul *Error Dialog*.

3.4.2 Use Case Diagram

Use case diagram digunakan untuk mendeskripsikan interaksi antara aktor (pengguna) dan sistem. Diagram ini menjelaskan berbagai fungsi atau layanan utama yang dapat diakses oleh pengguna, seperti memasukkan data gejala, mengirim permintaan klasifikasi, serta menerima hasil prediksi status jantung. *Use case* ini memetakan kebutuhan fungsional sistem secara visual. Penjelasan lebih lanjut mengenai *use case diagram* tersebut dapat dilihat pada Gambar 3.



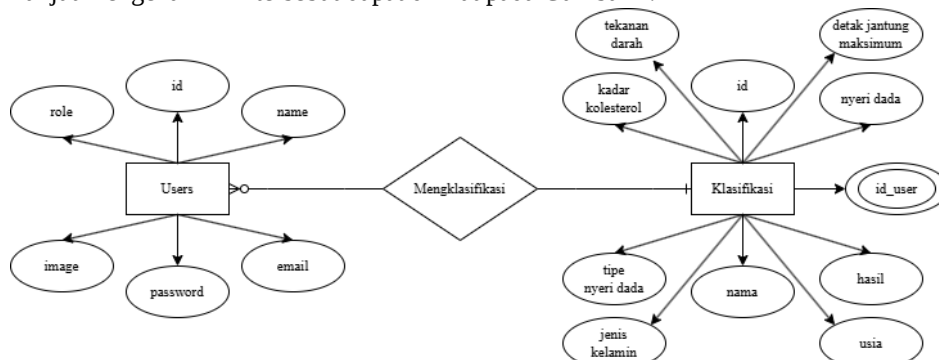
Gambar 4. *Use Case Diagram*

Use case diagram tersebut menggambarkan interaksi antara aktor dan sistem yang mencakup tiga komponen utama, yakni antarmuka *Mobile*, *Website*, dan *Admin-Site*.

- Pengguna dapat melakukan proses registrasi, login, klasifikasi penyakit, melihat riwayat klasifikasi, serta visualisasi data baik melalui aplikasi *mobile* maupun *website*. Khusus untuk versi *mobile*, pengguna juga difasilitasi dengan fitur edukasi untuk meningkatkan pengetahuan tentang penyakit jantung.
- Pada sisi *website*, pengguna mendapatkan fitur tambahan seperti *dashboard*, profil, kontak, layanan, dan informasi *about* yang memperkaya interaksi dengan sistem.
- Sementara itu, Admin memiliki akses ke panel *admin-site* yang memungkinkan untuk mengelola data pengguna, data klasifikasi, dan data layanan. Hal ini menunjukkan adanya manajemen data terpusat dan otorisasi yang jelas antara pengguna umum dan pihak pengelola sistem.

3.4.3 Entity Relationship Diagram

ERD digunakan untuk merancang struktur basis data yang akan digunakan dalam sistem. Diagram ini menjelaskan entitas-entitas utama (seperti pengguna dan hasil klasifikasi) beserta atribut-atributnya, serta relasi antar entitas. Dengan perancangan ERD, integritas data dapat terjaga dan mendukung proses penyimpanan dan pengambilan data secara efisien. Penjelasan lebih lanjut mengenai ERD tersebut dapat dilihat pada Gambar 4.



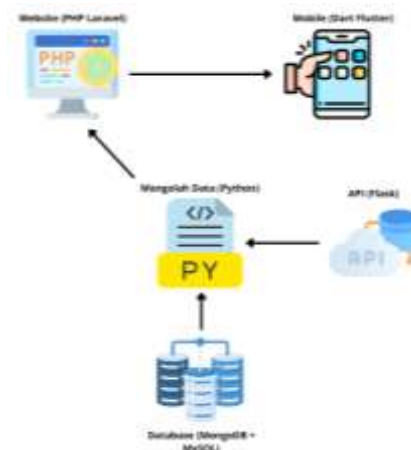
Gambar 4. *Entity Relationship Diagram*

Berdasarkan gambar *Entity Relationship Diagram (ERD)* yang ditampilkan, dapat dijelaskan bahwa sistem klasifikasi penyakit jantung ini melibatkan dua entitas utama, yaitu *Users* dan *Klasifikasi*, yang saling terhubung melalui relasi *Mengklasifikasi*. Entitas *Users* menyimpan informasi pengguna yang terdiri dari atribut *id*, *name*, *email*, *password*, *role*, dan *image*. Setiap pengguna dapat melakukan aktivitas klasifikasi data penyakit jantung, yang direpresentasikan oleh relasi *Mengklasifikasi*. Entitas *Klasifikasi* memuat data medis yang menjadi dasar proses klasifikasi, meliputi atribut:

- usia, jenis kelamin, tipe nyeri dada, tekanan darah, kadar kolesterol, detak jantung maksimum, serta hasil klasifikasi.
- atribut *id_user* pada entitas *Klasifikasi* berfungsi sebagai foreign key yang merujuk ke entitas *Users*, menandakan bahwa setiap data klasifikasi dikaitkan dengan pengguna yang melakukan input data.

3.4.4 Arsitektur Sistem

Arsitektur sistem menjelaskan komponen-komponen teknis dan hubungan antar bagian dalam sistem, termasuk *frontend* (*web* dan *mobile*), *backend API* yang dibangun dengan *Flask*, serta integrasi dengan model *machine learning Naive Bayes*. Diagram ini juga menggambarkan bagaimana data mengalir antara pengguna, *server*, dan model klasifikasi secara menyeluruh. Pendekatan ini mendukung fleksibilitas sistem agar dapat diakses melalui berbagai platform serta memudahkan pengembangan lebih lanjut. Penjelasan lebih lanjut mengenai arsitektur sistem tersebut dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur Sistem

Berdasarkan gambar arsitektur sistem yang ditampilkan, sistem klasifikasi penyakit jantung berbasis *web* dan *mobile* ini dirancang dengan pendekatan teknologi terdistribusi, yang mengintegrasikan beberapa komponen utama, yaitu *front-end* (*mobile* dan *web*), *back-end API*, *engine* pemrosesan data, dan *database*. Berikut adalah penjelasan teknis dan ilmiah dari arsitektur tersebut:

- Antarmuka Pengguna (User Interface)**
Sistem ini mendukung dua platform:
 - Website* dikembangkan menggunakan *PHP Laravel*, sebuah *framework* berbasis (*Model-View-Controller*) *MVC* yang handal dalam membangun sistem *web* dinamis.
 - Mobile application* dibangun menggunakan *Dart Flutter*, *framework* yang mendukung *cross-platform development*, memungkinkan aplikasi berjalan di *Android* maupun *iOS* dari satu basis kode yang sama.
- API Backend (Flask)**
Inti pemrosesan sistem berada pada *API* yang dibangun menggunakan *Flask Framework (Python)*. *Flask* dipilih karena ringan, fleksibel, dan cocok untuk mengembangkan *API* berbasis *machine learning*. Komponen *API* ini menjadi jembatan antara aplikasi (*web/mobile*) dengan proses klasifikasi penyakit jantung berbasis *Naive Bayes Algorithm*.
- Engine Pemrosesan Data**
Proses klasifikasi dilakukan pada bagian pemrosesan data menggunakan *Python*, di mana data dari pengguna akan melalui tahapan *preprocessing*, transformasi, dan prediksi menggunakan algoritma *Naive Bayes*. Hasil klasifikasi kemudian dikirim kembali melalui *API* untuk ditampilkan ke pengguna.
- Basis Data**
Sistem menggunakan kombinasi *MongoDB* dan *MySQL*:
 - MongoDB* digunakan untuk menyimpan data tidak terstruktur atau semi-terstruktur seperti histori klasifikasi dalam format dokumen *JSON*.

2. MySQL digunakan untuk menyimpan data terstruktur seperti data pengguna, *login*, dan metadata klasifikasi.

3.5 Implementasi Sistem

Tahap implementasi sistem merupakan proses merealisasikan rancangan yang telah dibuat ke dalam bentuk sistem yang dapat dijalankan. Pada penelitian ini, sistem klasifikasi penyakit jantung dikembangkan dengan menggunakan *Flask Framework* sebagai backend untuk mengelola logika aplikasi dan proses klasifikasi berbasis algoritma *Naive Bayes*. *Backend* ini kemudian diintegrasikan dengan antarmuka pengguna berbasis *web* dan *mobile* untuk memudahkan pengguna dalam melakukan input data dan memperoleh hasil klasifikasi secara *real-time*. Seluruh komponen sistem dikembangkan dengan mempertimbangkan efisiensi, skalabilitas, serta kemudahan akses, guna mendukung kebutuhan pengguna dalam mendeteksi potensi penyakit jantung secara cepat dan akurat. Implementasi ini juga mencakup koneksi ke model prediktif, pengelolaan basis data, dan penyusunan *API endpoint* sebagai penghubung antara sistem *frontend* dan *backend*.

3.5.1 Hasil Tampilan Mobile

- a. Awal Aplikasi *Mobile* dan Halaman *Register*



Gambar 6. Tampilan Awal Aplikasi *Mobile*



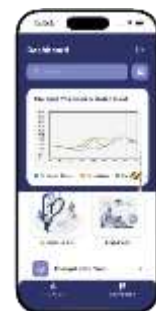
Gambar 7. Halaman *Register*

Gambar 6 merupakan tampilan awal aplikasi (*mobile*) yang menampilkan dua tombol utama, yaitu *Login* dan *Register*. Tampilan ini menyajikan ilustrasi medis dengan nuansa profesional yang mendukung konsep sebagai aplikasi pendamping manajemen jantung. Tampilan ini bertujuan untuk memudahkan pengguna memulai penggunaan aplikasi dengan memilih untuk masuk atau mendaftar terlebih dahulu. Sementara itu, gambar 7 memungkinkan pengguna baru untuk mendaftar ke dalam aplikasi. Tampil dengan desain modern dan minimalis, halaman ini dilengkapi ilustrasi medis yang mendukung tema aplikasi kesehatan jantung. Pengguna diminta mengisi *form* yang terdiri dari Nama, *Email*, dan *Password*. Terdapat *icon* untuk menampilkan atau menyembunyikan password yang meningkatkan kenyamanan dan keamanan pengguna saat mengisi data sensitif.

- b. Halaman *Login* dan Halaman *Dashboard*



Gambar 8. Halaman *Login*



Gambar 9. Halaman *Dashboard*

Gambar 8 halaman masuk bagi pengguna yang sudah memiliki akun. Pengguna diminta mengisi *Email* dan *Password*. Sementara itu, gambar 9 menampilkan *dashboard* utama yang berfungsi sebagai pusat kontrol dan pemantauan kesehatan jantung. Tampilan ini dirancang dengan berbagai fitur penting, seperti *search bar* yang memungkinkan pengguna mencari data atau fitur tertentu dengan cepat dan efisien. Selain itu, terdapat visualisasi data dalam bentuk grafik yang menampilkan parameter kesehatan seperti tekanan darah, kadar kolesterol, dan detak jantung maksimum dengan penggunaan garis berwarna untuk memudahkan interpretasi. *Dashboard* ini juga dilengkapi dengan fitur riwayat klasifikasi, yang menampilkan daftar hasil klasifikasi sebelumnya sebagai referensi kondisi kesehatan pengguna. Terakhir, tersedia navigasi cepat yang mempermudah pengguna untuk langsung mengakses halaman klasifikasi maupun edukasi.

- c. Halaman Klasifikasi dan Halaman Riwayat Klasifikasi



Gambar 10. Halaman Klasifikasi



Gambar 11. Halaman Riwayat Klasifikasi

Gambar 10 digunakan untuk melakukan klasifikasi kondisi jantung pengguna berdasarkan beberapa parameter kesehatan. Sementara itu, gambar 11 menampilkan daftar hasil riwayat klasifikasi penyakit jantung berdasarkan input pengguna yang sebelumnya telah dimasukkan melalui halaman klasifikasi. Setiap *entri* dalam riwayat ini mencerminkan hasil dari proses klasifikasi lengkap dengan data-data yang telah digunakan.

d. Halaman Edukasi



Gambar 12. Halaman Edukasi

Gambar 12 menampilkan informasi sebagai edukasi yang berfokus pada kesehatan jantung. Di bawah kolom pencarian, terdapat beberapa bagian yang menampilkan topik-topik edukasi dalam bentuk card yang masing-masing terdapat tombol untuk menampilkan penjelasan yang lebih lengkap.

3.5.2 Hasil Tampilan Website

a. Halaman Beranda Website dan Halaman Informasi



Gambar 13. Halaman Beranda Website



Gambar 14. Halaman Informasi

Gambar 13 merupakan tampilan awal aplikasi (*website*) yang terdapat logo atau nama aplikasi yaitu "HeartGuard." dan beberapa menu halaman. Sementara itu, gambar 14 berfungsi untuk memberikan penjelasan tentang penyakit jantung dan beberapa jenis penyakit jantung.

b. Halaman Layanan



Gambar 15. Halaman Layanan



Gambar 16. Halaman Kontak

Gambar 15 berfungsi sebagai pusat informasi dengan tujuan utama halaman ini adalah untuk memberikan jawaban dari pertanyaan-pertanyaan umum yang sering ditanyakan mengenai penyakit jantung. Sementara itu, gambar 16 berfungsi sebagai sarana bagi pengguna untuk berinteraksi dan berkonsultasi lebih lanjut terkait kondisi penyakit jantung.

c. Halaman *Sign Up* dan Halaman *Sign In*



Gambar 17. Halaman *Sign Up*



Gambar 18. Halaman *Sign In*

Gambar 17 adalah formulir untuk mendaftar akun. Pengguna diminta memasukkan nama, *email* dan *password* untuk mendaftarkan akun mereka yang nantinya digunakan saat ingin masuk ke dalam akun. Sementara itu, gambar 18 adalah formulir untuk masuk ke dalam akun. Pengguna diminta memasukkan *email* dan *password* untuk mengakses akun mereka.

d. Halaman Klasifikasi dan Halaman *Dashboard*



Gambar 19. Halaman Klasifikasi



Gambar 20. Halaman *Dashboard*

Gambar 19 berisi formulir yang digunakan untuk melakukan klasifikasi dengan meminta pengguna untuk memasukkan semua inputan yang diperlukan dan nantinya pengguna akan mendapatkan hasil klasifikasi secara langsung. Sementara itu, gambar 20 adalah *dashboard admin*. Menu navigasi juga menyertakan opsi lain seperti Riwayat dan User. Di pojok kanan atas, terdapat ikon yang digunakan untuk pengaturan atau profil pengguna.

e. Halaman Riwayat Klasifikasi dan Halaman Data Pengguna



Gambar 21. Halaman Riwayat Klasifikasi



Gambar 22. Halaman Data Pengguna

Gambar 21 adalah bagian Riwayat dari sistem admin, halaman ini berisi tabel hasil klasifikasi yang menunjukkan bahwa halaman ini menampilkan catatan dari klasifikasi pengguna yang telah dilakukan sebelumnya. Di bagian atas tabel, terdapat kolom pencarian yang digunakan admin untuk mencari data spesifik dalam riwayat. Sementara itu, gambar 22 adalah bagian manajemen *User*. Halaman ini berisi tabel data pengguna yang menunjukkan bahwa halaman ini menampilkan data-data pengguna yang mengakses aplikasi untuk melakukan klasifikasi penyakit jantung.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa implementasi algoritma *Naive Bayes* ke dalam *API Flask Framework Python* mampu menghasilkan sistem *backend* cerdas yang efektif dan efisien dalam melakukan klasifikasi penyakit jantung. Penggunaan dataset dari *Kaggle* yang telah melalui proses *preprocessing*, seleksi fitur, transformasi data, dan pembagian data uji serta data latih, memberikan dasar yang kuat bagi pengembangan model

klasifikasi yang akurat. Model *Naive Bayes* yang dibangun menunjukkan performa yang cukup baik dengan *accuracy* mencapai 73,77%, *precision* 67,57%, dan *recall* 86,21%. Hal ini menunjukkan bahwa model mampu melakukan klasifikasi dengan tingkat ketepatan yang cukup tinggi dan dapat digunakan sebagai alat bantu dalam mendeteksi risiko penyakit jantung secara dini. Integrasi model ke dalam sistem berbasis *API* menggunakan *Flask* memungkinkan pemanfaatan model secara fleksibel pada berbagai platform aplikasi, baik *web* maupun *mobile*. Hal ini mendukung pengembangan solusi digital dalam bidang kesehatan yang dapat diakses secara luas oleh masyarakat maupun tenaga medis. Dengan pendekatan ini, penelitian memberikan kontribusi nyata dalam penyediaan sistem pendukung keputusan berbasis kecerdasan buatan untuk deteksi penyakit jantung.

UCAPAN TERIMAKASIH

Penulis menyampaikan rasa syukur kepada Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga penelitian dan penulisan laporan ini dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, arahan, dan bantuan selama proses ini. Secara khusus, penulis menyampaikan apresiasi kepada Ketua Jurusan Teknologi Informasi serta Koordinator Program Studi Manajemen Informatika Politeknik Negeri Jember atas bimbingan, fasilitas, dan kesempatan yang diberikan selama pelaksanaan penelitian.

DAFTAR PUSTAKA

- [1] Kemenkes, "Penyakit Jantung Penyebab Utama Kematian, Kemenkes Perkuat Layanan Primer," Sehat Negeriku. Accessed: Nov. 06, 2025. [Online]. Available: <https://sehatnegeriku.kemkes.go.id>
- [2] M. Afriansyah, J. Saputra, V. Y. Ardhana, and Y. Sa'adati, "Algoritma Naive Bayes Yang Efisien Untuk Klasifikasi Buah Pisang Raja Berdasarkan Fitur Warna," *JISMDB J. Inf. Syst. Manag. Digit. Bus.*, vol. 1, no. 2, pp. 236–248, 2024, doi: 10.59407/jismdb.v1i2.438.
- [3] D. Lestari, "Metode Naïve Bayes Dalam Machine Learning Untuk Memprediksi Penyakit Jantung Dalam Tubuh," *J. Tek. J. Teknol. Komput. dan Sist. Informasi*, vol. 2, no. 1, pp. 23–28, 2022, [Online]. Available: <http://jurnal.goretanpena.com/index.php/teknisi>
- [4] L. Risqia and Z. Fatah, "Implementasi Analisa Penyakit Jantung Menggunakan Naïve Bayes," *Jamastika*, vol. 4, no. 1, pp. 102–107, 2025.
- [5] R. Hidayatur and N. A. Arinda, "Penerapan Algoritma Naïve Bayes Untuk Mendiagnosis Dan K-Means Untuk Mengelompokkan Penyakit Gagal Jantung," *Just IT J. Sist. Informasi, Teknol. Inf. dan Komput.*, vol. 15, no. 1, pp. 273–283, 2024.
- [6] C. Wijayanto and Y. A. Susetyo, "Implementasi Flask Framework Pada Pembangunan Aplikasi Sistem Informasi Helpdesk (SIH)," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 3, pp. 858–868, 2022, doi: 10.29100/jupi.v7i3.3161.
- [7] P. A. Sihotang and D. Sitanggang, "Perbandingan Algoritma C4.5 Dengan Naive Bayes Untuk Memprediksi Penyakit Jantung," *J. TEKINKOM*, vol. 7, no. 2, pp. 899–908, 2024, doi: 10.37600/tekinkom.v7i2.1535.
- [8] V. Simamora, A. Desiani, and I. Irmeilyana, "Klasifikasi Penyakit Hati Menggunakan Perbandingan Implementasi Algoritma Naïve Bayes Dan K-Nearest Neighbor," *J. Tek. Elektro dan Komputasi*, vol. 6, no. 1, pp. 12–17, 2024, doi: 10.32528/elkom.v6i1.18424.
- [9] A. Putranto, N. L. Azizah, and A. I. Ratna Ika, "Sistem Prediksi Penyakit Jantung Berbasis Web Menggunakan Metode SVM dan Framework Streamlit," *J. Penerapan Sist. Inf. (Komputer Manajemen)*, vol. 4, no. 2, pp. 442–452, 2023, [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/heart+disease>
- [10] I. Irpanudin, R. Reka, R. Nur Anggraeni, P. Pratama, A. Sujjada, and A. Fergina, "Prediksi Penyakit Jantung Menggunakan Metode Deep Neural Network dengan Memanfaatkan Internet of Things," *J. Inf. dan Teknol.*, vol. 5, no. 2, pp. 45–55, 2023, doi: 10.37034/jidt.v5i2.330.
- [11] C. A. WidiawatiRaras, L. Nurazizah, and I. R. Yunita, "Implementasi Algoritma Logistic Regression pada Pembuatan Website Sederhana untuk Prediksi Penyakit Jantung," *Infotekmesin*, vol. 15, no. 01, pp. 117–122, 2024, doi: 10.35970/infotekmesin.v15i1.2048.
- [12] D. Larassati, A. Zaidiah, and S. Afrizal, "Sistem Prediksi Penyakit Jantung Koroner Menggunakan Metode Naive Bayes," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 2, pp. 533–546, 2022, doi: 10.29100/jupi.v7i2.2842.
- [13] D. Ruswanti, D. Susilo, and R. Riani, "Implementasi CRISP-DM pada Data Mining untuk Melakukan Prediksi Pendapatan dengan Algoritma C4.5," *GO INFOTECH J. Ilm. STMIK AUB*, vol. 30, no. 1, pp. 111–121, 2024, doi: 10.36309/goi.v30i1.266.
- [14] A. Rianti, N. Wachid, A. Majid, and A. Fauzi, "CRISP-DM: Metodologi Proyek Data Science," in *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB) 2023*, 2023, pp. 107–114.