

**IMPLEMENTASI GRADIENT BOOSTING
MACHINE UNTUK MENDETEKSI
PENYAKIT JANTUNG**

SKRIPSI

DISUSUN OLEH

SAL SYABILA
NPM. 2109020037



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025**

IMPLEMENTASI GRADIENT BOOSTING MACHINE UNTUK MENDETEKSI PENYAKIT JANTUNG

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu Komputer
dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

**SAL SYABILA
NPM 2109020037**

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025**

LEMBAR PENGESAHAN

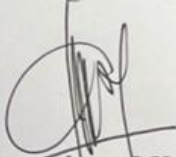
Judul Skripsi : IMPLEMENTASI GRADIENT BOOSTING MACHINE
UNTUK MENDETEKSI PENYAKIT JANTUNG
Nama Mahasiswa : SAL SYABILA
NPM : 2109020037
Program Studi : TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing



(Fatma Sari Hutagalung, S.Kom, M.Kom)
NIDN. 0117019301

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom, M.Kom)
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

IMPLEMENTASI GRADIENT BOOSTING MACHINE UNTUK MENDETEKSI PENYAKIT JANTUNG

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 22 Juli 2025

Yang membuat pernyataan



SAL SYABILA

NPM. 2109020037

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Sal Syabila
Npm : 2109020037
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**IMPLEMENTASI GRADIENT BOOSTING MACHINE UNTUK
MENDETEKSI PENYAKIT JANTUNG**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 22 Juli 2025
Yang membuat pernyataan



SAL SYABILA
NPM 2109020037

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : SAL SYABILA
Tempat dan Tanggal Lahir : MEDAN, 06 MAI 2003
Alamat Rumah : JL MARELAN IX LINGK-04 T 600
Telepon/Faks/HP : 0895 3270 09461
E-mail : salsyabila65@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : SDS HARAPAN MULIA MEDAN TAMAT: 2014
SMP : SMPN TERBUKA 38 MEDAN TAMAT: 2017
SMA : YP SMKS BINA SATRIA MEDAN TAMAT: 2020

KATA PENGANTAR



Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, S.Kom, M.Kom Ketua Program Studi Teknologi Informasi
4. Bapak Mhd. Basri, S.Si, M.Kom Sekretaris Program Studi Teknologi Informasi
5. Pembimbing skripsi Ibu Fatma Sari Hutagalung, S.Kom, M.Kom
6. Abangku yang hebat Iqnu Anggara Terimakasih yang tak terhingga atas semua pengorbanan serta sudah merangkap peran sebagai ayah, mendukung, menyemangati, membiayai semuanya, tanpa abang mungkin penulis tidak akan sampai dititik ini, Skripsi ini, kelulusan ini awal dari janji untuk membuat abang bangga serta bukti nyata dari kasih sayang dan kerja keras inshaallah dibalas allah dengan banyaknya kebaikan dan rezeki yang berlipat ganda Aamiin.
7. Mama Elya Masita Terimakasih karena semua pencapaianku ini hasil dari doa-doa dan dukunganya.
8. Diri penulis, Keluarga, Orang terdekat, Teman Seangkatan serta semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu terimakasih telah membantu dalam penyelesaian skripsi ini.

IMPLEMENTASI *GRADIENT BOOSTING MACHINE* UNTUK MENDETEKSI PENYAKIT JANTUNG

ABSTRAK

Penyakit jantung merupakan salah satu penyebab utama kematian di tingkat global dan menjadi ancaman kesehatan yang serius di Indonesia. Keterbatasan akses dan biaya tinggi seringkali menjadi penghambat dalam upaya deteksi dini. Penelitian ini bertujuan untuk mengimplementasikan algoritma *Gradient Boosting Machine* (GBM) guna membangun model klasifikasi risiko penyakit jantung, serta merancang sistem prediksi berbasis web yang dapat diakses oleh masyarakat umum sebagai alat bantu deteksi dini non-diagnostik. Metode penelitian mencakup beberapa tahapan utama, mulai dari akuisisi dan pra-pemrosesan data sekunder dari platform Kaggle, pelatihan model GBM dengan optimasi *hyperparameter* menggunakan *Grid Search*, hingga implementasi model ke dalam sistem terintegrasi. Sistem ini dikembangkan dengan arsitektur *frontend* (HTML, PHP, MySQL) dan *backend* (Python, Flask) yang terhubung melalui REST API. Hasil evaluasi model menunjukkan performa yang sangat tinggi dengan akurasi global mencapai **98,54%**. Secara spesifik, model ini berhasil mencapai nilai *Recall* **1.0000** untuk kelas "Sakit Jantung", yang berarti mampu mengidentifikasi seluruh kasus positif tanpa ada yang terlewat, dan nilai *Precision* **1.0000** untuk kelas "Sehat". Penelitian ini berhasil menghasilkan sebuah sistem prediksi fungsional yang tidak hanya akurat tetapi juga praktis, dilengkapi antarmuka ramah pengguna untuk publik dan dasbor komprehensif bagi admin untuk mengelola model dan konten edukasi secara mandiri.

Kata Kunci: *Gradient Boosting Machine*, Penyakit Jantung, *Machine Learning*, Sistem Prediksi, Klasifikasi.

IMPLEMENTATION OF GRADIENT BOOSTING MACHINE TO DETECT HEART DISEASE

ABSTRACT

Heart disease is a leading cause of death globally and a serious health threat in Indonesia. Limited access and high costs often hinder early detection efforts. This research aims to implement the Gradient Boosting Machine (GBM) algorithm to build a heart disease risk classification model and to design a web-based prediction system accessible to the general public as a non-diagnostic early detection tool. The research method includes several main stages, from the acquisition and preprocessing of secondary data from the Kaggle platform, training the GBM model with hyperparameter optimization using Grid Search, to the implementation of the model into an integrated system. The system was developed with a frontend (HTML, PHP, MySQL) and backend (Python, Flask) architecture connected via a REST API. The model evaluation results show very high performance, with a global accuracy reaching 98.54%. Specifically, the model successfully achieved a Recall value of 1.0000 for the "Heart Disease" class, meaning it is capable of identifying all positive cases without any being missed, and a Precision value of 1.0000 for the "Healthy" class. This research successfully produced a functional prediction system that is not only accurate but also practical, equipped with a user-friendly interface for the public and a comprehensive dashboard for admins to independently manage the model and educational content.

Keywords: *Gradient Boosting Machine, Heart Disease, Machine Learning, Prediction System, Classification.*

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PERNYATAAN PERSETUJUAN PUBLIKASI	iv
RIWAYAT HIDUP	v
KATA PENGANTAR	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	6
BAB II LANDASAN TEORI	7
2.1 Penyakit Jantung	7
2.2 Deteksi Dini Penyakit Jantung	7
2.3 Gejala dan Faktor Risiko Penyakit Jantung	7
2.4 Machine Learning	8
2.5 Algoritma Gradient Boosting Machine	9
2.6 Confussion Matrix	11
2.7 Pra-Pemrosesan Data	12
2.8 Hypertext Markup Language (HTML)	13
2.9 Hypertext Preprocessor (PHP)	13
2.10 MySQL	14
2.11 Python	14
2.12 Pandas	15
2.13 Scikit-Learn	15

2.14	Flask API.....	15
2.15	Penelitian Terdahulu	16
2.16	Unified Modelling Language (UML).....	17
BAB III METODE PENELITIAN.....		20
3.1	Pendekatan dan Alur Penelitian	20
3.2	Sumber dan Jenis Data	22
3.3	Tahapan Penelitian	23
3.3.1.	Akuisisi dan Pra-Pemrosesan Data	23
3.3.2.	Perhitungan Algoritma <i>Gradient Boosting Machine</i>	23
3.3.3.	Implementasi dengan <i>Python</i>	28
3.3.4.	Evaluasi Model	30
3.4	Desain Sistem.....	32
3.5	Desain Basis Data	51
BAB IV HASIL DAN PEMBAHASAN		54
4.1.	Perhitungan <i>Gradient Boosting Machine</i>	54
4.1.1.	Proses <i>Label Encoding</i>	56
4.1.2.	Analisa Korelasi Fitur	58
4.1.3.	Persiapan dan Pemisahan Data.....	60
4.1.4.	Standardisasi Data.....	62
4.1.5.	Implementasi <i>Gradient Boosting Machine</i> dengan <i>Python</i>	63
4.2.	Hasil Implementasi Sistem.....	70
4.2.1.	Tampilan Menu Pengguna	70
4.2.2.	Tampilan Menu Admin	73
4.3.	Pengujian Program.....	85
4.4.	Kelebihan dan Kekurangan Program	88
BAB V KESIMPULAN DAN SARAN.....		92
5.1	Kesimpulan	92
5.2	Saran.....	93
DAFTAR PUSTAKA		94

DAFTAR GAMBAR

Gambar 3. 1 Alur Penelitian.....	20
Gambar 3. 2 Flowchart Aplikasi	33
Gambar 3. 3 Use case diagram.....	34
Gambar 3. 4 Activity diagram Login	35
Gambar 3. 5 Activity diagram Dashboard	36
Gambar 3. 6 Activity diagram Kelola Model Prediksi	37
Gambar 3. 7 Activity diagram Lihat Riwayat Prediksi.....	38
Gambar 3. 8 Activity diagram Kelola Data Edukasi.....	39
Gambar 3. 9 Activity diagram Logout	40
Gambar 3. 10 Activity Diagram Form Prediksi	41
Gambar 3. 11 Activity Diagram Lihat Hasil.....	42
Gambar 3. 12 Activity Diagram Unduh Hasil	43
Gambar 3. 13 Activity Diagram Edukasi Kesehatan	44
Gambar 3. 14 Sequence Diagram Login	44
Gambar 3. 15 Sequence Diagram Dashboard	45
Gambar 3. 16 Sequence Diagram Kelola Model Prediksi	45
Gambar 3. 17 Sequence Diagram Lihat Riwayat Prediksi.....	46
Gambar 3. 18 Sequence Diagram Kelola Konten Edukasi	46
Gambar 3. 19 Sequence Diagram Logout	47
Gambar 3. 20 Sequence Diagram Form Prediksi.....	47
Gambar 3. 21 Sequence Diagram Lihat Hasil.....	48
Gambar 3. 22 Sequence Diagram Unduh Hasil	48
Gambar 3. 23 Sequence Diagram Edukasi Kesehatan Jantung.....	49
Gambar 3. 24 Class Diagram	50

DAFTAR TABEL

Tabel 3. 1 Penelitian Terdahulu	16
Tabel 3. 2 Simulasi Data Pasien.....	24
Tabel 3. 3 Tabel Perhitungan Residual	25
Tabel 3. 4 Tabel update logit.....	27
Tabel 3. 5 Tabel Probabilitas baru pada iterasi ke -1	27
Tabel 3. 6 Nilai Residual pada iterasi ke-1	27
Tabel 3. 7 Rancangan Tabel Admin.....	51
Tabel 3. 8 Rancangan Tabel Prediksi.....	52
Tabel 3. 9 Rancangan Tabel Model Prediksi	52
Tabel 3. 10 Rancangan Tabel Artikel Edukasi.....	53

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Penyakit jantung atau penyakit kardiovaskular merupakan salah satu penyebab utama kematian di dunia. Menurut data *World Health Organization* (WHO), sekitar 17,9 juta orang meninggal setiap tahun akibat penyakit kardiovaskular, yang mencakup serangan jantung dan stroke (Devia In Baliani et al., n.d.). Di Indonesia, penyakit jantung juga menjadi ancaman serius bagi Kesehatan masyarakat. Data kementerian Kesehatan republik Indonesia menunjukkan bahwa prevalensi penyakit jantung meningkat dari 0,5% pada tahun 2013 menjadi 1,5% pada tahun 2018. Peningkatan prevalensi ini disebabkan oleh berbagai faktor risiko, seperti merokok, pola makan tidak seimbang, kurangnya aktivitas fisik, obesitas, dan diabetes melitus. Perubahan gaya hidup yang tidak sehat ini berkontribusi terhadap tingginya angka kejadian penyakit jantung di Indonesia (Dinas Kesehatan Kalbar, 2024).

Penyakit jantung sendiri memiliki berbagai jenis, antara lain: Penyakit Jantung Koroner, Penyakit Jantung Hipertensif, Penyakit Jantung Rematik,. Penyakit Jantung Gagal, Penyakit Jantung Kardiomiopati (Rashad et al., 2023). Selain itu, keterbatasan akses terhadap fasilitas medis, biaya pemeriksaan, dan minimnya kesadaran masyarakat mengenai pentingnya pemeriksaan dini menjadi hambatan signifikan dalam pencegahan penyakit jantung di masyarakat umum.

Deteksi dini penyakit jantung memegang peranan krusial dalam sistem kesehatan karena menjadi langkah awal untuk mencegah komplikasi lebih lanjut serta menurunkan risiko kematian mendadak. Dengan melakukan skrining dan

diagnosis secara lebih awal, pasien yang berisiko tinggi dapat segera mendapatkan penanganan medis yang tepat. Deteksi dini juga memungkinkan intervensi gaya hidup sehat atau terapi farmakologis yang lebih cepat, sehingga hasil pengobatan cenderung lebih baik (Mayo Clinic, 2023). Proses deteksi ini biasanya melibatkan sejumlah alat diagnostik seperti el

Ektrokardiogram (EKG) untuk memantau aktivitas listrik jantung, echocardiography (USG jantung) untuk melihat struktur dan fungsi bilik jantung, treadmill test untuk mengevaluasi performa jantung saat aktivitas fisik, serta angiografi dan CT scan jantung untuk melihat pembuluh darah dan mendeteksi penyumbatan. Deteksi tersebut seringkali hanya tersedia di fasilitas medis tertentu, serta memerlukan biaya dan tenaga spesialis.

Perkembangan teknologi informasi dan kecerdasan buatan membuka peluang untuk mendukung solusi kesehatan berbasis data. Salah satu pendekatan yang dapat dimanfaatkan adalah *machine learning*, yaitu teknologi yang memungkinkan komputer mempelajari pola dari data dan membuat prediksi terhadap data baru secara otomatis (IBM, 2023). Dalam konteks prediksi penyakit, machine learning sangat berguna untuk mengidentifikasi pola risiko dari data pasien, seperti usia, tekanan darah, kadar kolesterol, dan parameter medis lainnya.

Dari berbagai algoritma yang tersedia dalam machine learning, Gradient Boosting Machine (GBM) merupakan salah satu metode yang terbukti efektif dalam menyelesaikan masalah klasifikasi. GBM bekerja dengan membangun beberapa model lemah (*weak learners*) secara bertahap dan memperbaiki kesalahan prediksi dari model sebelumnya melalui pendekatan *gradient descent*.

Model akhir yang dihasilkan mampu meminimalkan fungsi kerugian dan menghasilkan akurasi yang tinggi, terutama pada data yang kompleks dan tidak linear. Selain itu, GBM juga memiliki keunggulan dalam menangani data numerik maupun kategorikal tanpa banyak prasyarat (Mulyo & Khanif Zyen, 2025).

Namun, model prediktif yang dibangun dalam lingkungan pengembangan saja tidak cukup. Diperlukan sebuah sistem informasi yang dapat menjembatani hasil pemodelan machine learning agar dapat digunakan secara langsung oleh pengguna umum. Dalam hal ini, pengembangan sistem berbasis web menjadi pilihan utama. Platform web dipilih karena bersifat platform-independen, tidak memerlukan instalasi khusus, dapat diakses di berbagai perangkat, dan memungkinkan integrasi antar sistem secara efisien. Sistem web memungkinkan pengguna memasukkan data secara mandiri melalui antarmuka yang sederhana, lalu sistem akan memproses data tersebut melalui model machine learning yang terhubung di backend.

Penelitian ini mengimplementasikan algoritma GBM menggunakan bahasa pemrograman Python, yang dikenal karena sintaksnya yang sederhana dan didukung oleh pustaka-pustaka populer seperti pandas untuk manipulasi data, scikit-learn untuk pembelajaran mesin, dan Flask untuk membangun REST API yang menghubungkan model dengan sistem web. Sistem ini kemudian diintegrasikan dengan frontend berbasis HTML dan PHP serta basis data MySQL sebagai penyimpanan input-output prediksi. Dengan arsitektur ini, pengguna cukup mengakses halaman web, memasukkan data, dan sistem akan menampilkan hasil prediksi risiko penyakit jantung secara langsung.

Dengan pendekatan ini, hubungan antara machine learning dan sistem web terjalin secara fungsional dan aplikatif. Machine learning menjadi otak analisis di backend, sedangkan web menjadi medium interaksi antara pengguna dan sistem. Ini bukan hanya menjadikan penelitian bersifat teoritis, tetapi juga praktis, karena hasilnya dapat digunakan masyarakat sebagai alat bantu deteksi dini non-diagnostik yang murah, cepat, dan mudah diakses.

Oleh karena itu, penelitian ini tidak hanya bertujuan membangun model klasifikasi penyakit jantung menggunakan Gradient Boosting Machine, tetapi juga merancang sistem prediksi risiko penyakit jantung berbasis web yang terintegrasi dengan model machine learning. Sistem ini dirancang untuk dapat digunakan oleh masyarakat umum, dengan harapan dapat menjadi inovasi teknologi yang mendukung upaya promotif dan preventif di bidang kesehatan Masyarakat, khususnya dalam deteksi penyakit jantung.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana mengimplementasikan algoritma *Gradient Boosting Machine* (GBM) untuk melakukan klasifikasi risiko penyakit jantung berdasarkan data sekunder yang didapatkan dari *platform* Kaggle?
2. Bagaimana merancang sistem prediksi penyakit jantung berbasis web yang terintegrasi dengan model GBM yang dapat digunakan secara mandiri oleh Masyarakat?
3. Bagaimana mengevaluasi performa model GBM menggunakan metrik klasifikasi seperti akurasi, presisi, *recall*, dan *F1-Score*?

1.3 Batasan Masalah

Agar penelitian lebih terfokus, maka diterapkan batasan-batasan masalah seperti berikut ini :

1. Penelitian ini hanya menggunakan algoritma *Gradient Boosting Machine* (GBM) untuk proses klasifikasi risiko penyakit jantung, tanpa melakukan perbandingan langsung dengan algoritma *machine learning* lainnya.
2. Dataset yang digunakan merupakan data sekunder yang bersifat terbuka dan tersedia secara public melalui *platform* Kaggle, serta hanya mencakup fitur-fitur yang relevan untuk prediksi risiko penyakit jantung.
3. Sistem prediksi yang dibangun bersifat prototipe berbasis *web* dan dirancang untuk digunakan secara mandiri oleh masyarakat umum, tanpa integrasi langsung dengan sistem informasi rumah sakit atau rekam medis elektronik.
4. Sistem ini dibangun dengan dua komponen utama yaitu *frontend* dan *backend*. *Frontend* menggunakan HTML, PHP, dan MySQL untuk menampilkan tampilan *web* dan menerima *input* pengguna. *Backend* dibuat dengan *Python* dan *Flask* untuk menjalankan model *machine learning* (GBM). Keduanya dihubungkan melalui *REST API*, sehingga data dari web dikirim ke server *Python* untuk diproses, lalu hasil prediksi dikembalikan ke *web* untuk ditampilkan kepada pengguna.
5. Evaluasi model hanya menggunakan metrik seperti akurasi, presisi, recall, dan F1-score, tanpa membahas cara kerja model, keadilan, atau keamanan data.
6. Sistem ini hanya ditujukan sebagai alat bantu deteksi dini berbasis data dan bersifat non-diagnostik, sehingga tidak dimaksudkan untuk menggantikan peran atau keputusan dari tenaga medis profesional.

1.4 Tujuan Penelitian

Adapun yang menjadi tujuan dalam penelitian ini adalah sebagai berikut :

1. Mengimplementasikan algoritma Gradient Boosting Machine (GBM) dalam mendeteksi risiko penyakit jantung berdasarkan data sekunder yang tersedia secara publik melalui platform Kaggle.
2. Merancang dan membangun sistem prediksi penyakit jantung berbasis *web* yang dapat digunakan secara mandiri oleh masyarakat umum sebagai alat bantu deteksi dini non-diagnostik.
3. Mengevaluasi performa algoritma GBM dalam mengklasifikasikan risiko penyakit jantung menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score.

1.5 Manfaat Penelitian

Adapun yang menjadi manfaat dalam penelitian ini adalah sebagai berikut

1. Memberikan kontribusi dalam penerapan algoritma *Gradient Boosting Machine* untuk klasifikasi risiko penyakit jantung berdasarkan data sekunder, sehingga dapat dijadikan referensi dalam studi prediktif berbasis *machine learning* di bidang Kesehatan.
2. Menjadi dasar dalam perancangan sistem prediksi risiko penyakit jantung berbasis *web* yang dapat digunakan secara mandiri oleh Masyarakat umum sebagai alat bantu deteksi dini non-diagnostik
3. Menyediakan hasil evaluasi performa model GBM menggunakan metrik akurasi, presisi, recall, dan F1-score yang dapat digunakan sebagai acuan dalam pengembangan atau validasi model serupa di masa mendatang.

BAB II

LANDASAN TEORI

2.1 Penyakit Jantung

Penyakit jantung merupakan suatu istilah umum dari semua gangguan yang penyebabnya berasal dari hati. Penyakit Kardiovaskular (CDV) menjadi penyebab utama kasus kematian di dunia dibandingkan penyakit lainnya. Penyakit jantung adalah penyakit tidak menular yang menyebabkan kasus kematian paling besar. Berdasarkan laporan *World Health Organization* (WHO) pada tahun 2000, yang dapat membuat seseorang tidak dapat mengenali gejala awalnya sehingga awal pengobatan menjadi sulit (Rashad et al., 2023).

2.2 Deteksi Dini Penyakit Jantung

Deteksi dini penyakit jantung merupakan Langkah penting dalam pencegahan penyakit jantung koroner, khususnya bagi individu yang berisiko tinggi seperti penderita hipertensi. Salah satu metode manual yang sering digunakan adalah pemeriksaan tekanan secara berkala menggunakan tensimeter, yang disertai dengan edukasi mengenai risiko dan gejala penyakit jantung (Lina & Saraswati, 2020).

2.3 Gejala dan Faktor Risiko Penyakit Jantung

Penyakit jantung dapat menunjukkan berbagai gejala, tergantung pada jenis dan tingkat keparahannya. Beberapa gejala umum yang sering dialami meliputi:

1. Nyeri atau ketidaknyamanan di Dada seperti sensasi nyeri, tekanan, atau rasa terbakar di area dada
2. Sesak napas atau napas pendek, terutama saat beraktivitas

3. Merasa Lelah tanpa sebab yang jelas, bahkan saat melakukan aktivitas ringan
4. Merasa pusing, kepala ringan, atau kehilangan kesadaran
5. Detak jantung tidak teratur (Palpitasi) dan jantung berdebar-debar.
6. Pembengkakan pada Kaki atau pergelangan kaki (Kemenkes, 2024).

Adapun Beberapa faktor dapat meningkatkan risiko seseorang terkena penyakit jantung. Faktor-faktor tersebut dibagi menjadi dua kategori:

1. Faktor risiko yang tidak dapat diubah, seperti usia, jenis kelamin, dan Riwayat keluarga
2. Faktor risiko yang dapat diubah, seperti Merokok, Tekanan Darah Tinggi, Kadar Kolesterol Tinggi, Diabetes Mellitus, Obesitas, Kurang Aktivitas Fisik, Pola Makan Tidak Sehat, dan Stres (Kemenkes, 2022).

2.4 Machine Learning

Machine Learning (ML) adalah cabang dari kecerdasan buatan (Artificial Intelligence atau AI) yang memungkinkan sistem komputer untuk belajar dari data dan meningkatkan kinerjanya seiring waktu tanpa diprogram secara eksplisit. Machine learning (ML) adalah cabang AI dan ilmu komputer yang berfokus pada penggunaan data dan algoritma untuk memungkinkan AI meniru cara manusia belajar (IBM, 2023).

Machine Learning dapat dikategorikan menjadi beberapa jenis, antara lain :

1. Pembelajaran Terawasi (*Supervised Learning*): Model dilatih menggunakan data berlabel, di mana algoritma mempelajari hubungan antara input dan output untuk kemudian memprediksi output pada data baru.

2. Pembelajaran Tanpa Pengawasan (*Unsupervised Learning*): Model bekerja dengan data yang tidak berlabel, berusaha menemukan struktur atau pola tersembunyi dalam data tersebut.
3. Pembelajaran Penguatan (*Reinforcement Learning*): Model belajar melalui interaksi dengan lingkungan dan menerima umpan balik dalam bentuk reward atau penalti, bertujuan untuk memaksimalkan reward kumulatif.

2.5 Algoritma *Gradient Boosting Machine*

Gradient Boosting merupakan salah satu metode *ensemble learning* yang bekerja dengan menggabungkan beberapa model lemah (*weak learners*) biasanya berupa *decision tree*, untuk membentuk model yang kuat (*strong learner*) secara bertahap. Setiap model baru yang ditambahkan bertujuan untuk memperbaiki kesalahan prediksi dari model sebelumnya dengan cara meminimalkan fungsi kerugian (*loss function*) menggunakan pendekatan *gradient descent*. Metode ini sangat efektif untuk menangani permasalahan regresi dan klasifikasi, serta dikenal memiliki akurasi tinggi dalam berbagai aplikasi data mining dan machine learning.

Secara umum, proses *Gradient Boosting* dapat dinyatakan dalam rumus-rumus berikut :

1. Inisialisasi Model Awal

$$f_0(x) = \arg \min + \sum_{i=1}^n L(y_i, \gamma)$$

Pada Langkah awal, model GBM dimulai dengan mencari nilai konstanta γ yang meminimalkan kesalahan prediksi terhadap seluruh data pelatihan.

2. Hitung Pseudo-Residuals

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$$

Pseudo-residual merupakan selisih antara nilai actual dan prediksi model sebelumnya. Nilai ini digunakan sebagai target baru untuk model selanjutnya.

3. Latih Model Baru untuk Memprediksi Residual

$$h_m(x) \approx r_{im}$$

Model baru, seperti pohon keputusan sederhana, dilatih untuk mempelajari pola dari pseudo-residual. Tujuan model ini adalah memperbaiki kesalahan prediksi dari model sebelumnya.

4. Hitung Koefisien Optimal γ_m

$$\gamma_m = \arg \min \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

Setelah model $h_m(x)$ dibuat, ditentukan nilai γ_m yang paling optimal untuk mengurangi kesalahan prediksi saat model tersebut ditambahkan ke model sebelumnya.

5. Perbarui Model Keseluruhan

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

Model diperbarui dengan menambahkan kontribusi dari model baru. Proses ini dilakukan berulang kali hingga mencapai jumlah iterasi maksimal atau hingga *error* konvergen.

2.6 Confussion Matrix

Confusion Matrix adalah tabel yang menyatakan klasifikasi jumlah uji yang benar dan jumlah uji yang salah (Normawati & Prayogi, 2021). *Confusion Matrix* sendiri memiliki empat istilah :

1. *True Negative* (TN) yang berarti model klasifikasi memprediksi data ada di kelas *negative* namun yang sebenarnya data memang ada di kelas *negative*.
2. *True Positive* (TP) yang berarti model klasifikasi memprediksi data ada di kelas positif namun sebenarnya memang data berada di kelas positif.
3. *False Negative* (FN) yang berarti model klasifikasi memprediksi data ada di kelas *negative* namun yang sebenarnya data ada di kelas positif.
4. *False Positive* (FP) yang berarti model klasifikasi memprediksi data ada di kelas positif namun yang sebenarnya data ada di kelas *negative*.

Dari definisi *confusion matrix*, beberapa poin poin dalam *confusion matrix* digunakan untuk menghitung *precision*, *recall*, dan *f1 score*. *Precision* adalah perbandingan antara *True Positive* (TP) dengan banyaknya data yang diprediksi positif, secara matematis dapat dilihat dibawah ini

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(2.2)$$

Untuk *recall* sendiri adalah perbandingan antara *true positive* (TP) dengan banyaknya data yang sebenarnya positif. Dapat dinyatakan secara matematis seperti dibawah ini

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(2.3)$$

Sedangkan *F1 Score* adalah nilai tengah dari *precision* dan *recall*. Nilai terbaik dari *F1 Score* adalah 1 dan nilai terburuknya adalah 0, secara matematis dapat dituliskan seperti dibawah ini

$$\frac{1}{F1} = \frac{1}{2} \left(\frac{1}{precision} + \frac{1}{recall} \right) \dots\dots\dots(2.4)$$

Nilai *F1 Score* yang baik menandakan bahwa model klasifikasi kita punya *precision* dan *recall* yang baik (S. Setiawan, 2020).

2.7 Pra-Pemrosesan Data

Pra-pemrosesan data adalah tahapan untuk mengolah data mentah dengan menghilangkan beberapa permasalahan yang mengganggu saat pemrosesan data. Hal tersebut dikarenakan oleh data yang formatnya tidak konsisten. Melalui proses ini, pemodelan algoritma KNN akan berjalan lebih efektif dan efisien. Adapun tahapan tahapan dalam pra-pemrosesan data adalah :

1. Pertama-tama dalam tahap awal *preprocessing* data, langkah yang esensial adalah melakukan data cleaning. Proses ini melibatkan pemilihan kembali data mentah untuk menghilangkan entri yang tidak lengkap, tidak relevan, atau tidak akurat. Dengan melakukan ini, kita dapat menghindari kesalahpahaman saat melakukan analisis terhadap data tersebut.
2. Langkah berikutnya adalah *data integration*, yang diperlukan karena *preprocessing* data melibatkan penggabungan data dari berbagai sumber ke dalam satu dataset. Penting untuk memastikan bahwa data dari berbagai sumber memiliki format yang seragam.
3. Setelah itu, kita melanjutkan ke tahap transformasi data. Sebagaimana dijelaskan sebelumnya, data yang berasal dari sumber yang berbeda mungkin memiliki format yang beragam. Oleh karena itu, perlu dilakukan

penyesuaian format agar seluruh data yang terkumpul memiliki struktur yang seragam, memudahkan proses analisis data.

4. Tahap terakhir dalam *preprocessing* data adalah mengurangi jumlah data, yang dikenal sebagai *data reduction*. Tujuan utamanya adalah mengurangi sampel data tanpa mengubah hasil analisis. Terdapat tiga teknik yang dapat diterapkan pada tahap ini, yaitu pengurangan dimensi (*dimensionality reduction*), pengurangan jumlah (*numerosity reduction*), dan kompresi data (Binus Student Corner, 2022).

2.8 Hypertext Markup Language (HTML)

HTML atau *HyperText Markup Language* adalah Bahasa markup standar yang digunakan untuk membuat halaman *web*. HTML berfungsi untuk membuat struktur dari sebuah *website*. HTML digunakan untuk menandai bagian mana yang akan menjadi judul artikel, bagian mana yang berfungsi sebagai isi artikel, atau bagian mana yang disajikan dalam bentuk tabel (Andre Pratama, 2020).

2.9 Hypertext Preprocessor (PHP)

PHP adalah Bahasa *scripting* yang menyatu dengan HTML yang dijalankan pada server sedang yang dikirim ke browser hanya hasilnya saja. Ketika pengguna internet membuka situs yang menggunakan fasilitas *serverside scripting* PHP, maka terlebih dahulu server yang bersangkutan akan memproses semua perintah PHP lalu mengirimkan hasilnya dalam format HTML ke *web server* pengguna internet tadi. Sehingga kode asli yang ditulis dengan PHP tidak terlihat di *browser* pengguna. PHP memiliki fungsi untuk mengambil informasi dari *form* berbasis *web* dan menggunakannya untuk berbagai macam fungsi (Aniqsa, 2019).

2.10 MySQL

MySQL merupakan perangkat lunak basis data relasi atau *Relational Database Management System* (RDBMS) yang diterapkan menggunakan lisensi GPL (*General Public License*). MySQL memiliki kinerja, kecepatan proses yang tidak kalah dengan *database-database* besar lainnya (Aniqsa, 2019). Beberapa pertimbangan dalam memilih MySQL antara lain :

1. Kecepatan
2. Mudah Digunakan
3. Terbuka
4. Kapabilitas
5. Konektifitas dan Keamanan

2.11 Python

Python adalah salah satu Bahasa pemrograman yang populer, yang dibuat pertama kali pada tahun 1991 oleh Guido van Rossum. Bahasa Pemrograman Python sendiri dapat digunakan dalam pengembangan aplikasi *web*, pengembangan software, hingga menyelesaikan permasalahan matematika dari yang dasar dan kompleks dan bisa digunakan untuk pengembangan algoritma prediksi.

Keunggulan menggunakan Python adalah, dapat digunakan di berbagai platform, memiliki *syntax* yang mudah dan *simple*, dan dapat digunakan dalam pemrograman procedural, pemrograman berorientasi objek, maupun pemrograman fungsional.

2.12 *Pandas*

Pandas adalah *library* pada *Python* yang paling sering dipakai untuk menganalisis data. Cara kerja *pandas* cukup menarik, Data berupa format CSV atau SQL diubah menjadi objek dalam *Python* dengan bentuk baris dan kolom yang disebut sebagai *Dataframe*. Objek *Dataframe* ini akan terlihat sangat mirip dengan tabel yang terdapat dalam aplikasi pengolahan statistika pada umumnya.

Penggunaan *Library Pandas* juga dapat membantu dalam verifikasi data, pengolahan data, normalisasi data, penggabungan dan penyatuan data, inspeksi data, serta membuat dan menyimpan data .

2.13 *Scikit-Learn*

Scikit-Learn adalah *library* dalam bahasa pemrograman *python* yang membantu melakukan proses pada data ataupun melakukan pelatihan pada data untuk kebutuhan *data science* seperti klasifikasi, *clustering*, dan lain-lain. *Library* ini sangat populer dikarenakan banyak sekali model pembelajaran mesin yang dapat dipanggil menggunakan *scikit-learn* (Bisa AI Academy, 2023).

2.14 *Flask API*

Flask adalah sebuah *framework web* yang ditulis menggunakan bahasa pemrograman *Python*, yang termasuk dalam kategori *microframework*. Meskipun *Flask* tergolong ringan, *framework* ini menawarkan fleksibilitas yang tinggi bagi pengembang untuk membangun aplikasi *web* yang terstruktur. *Flask* memudahkan pengelolaan tampilan dan logika aplikasi *web*, sehingga cocok digunakan untuk proyek *web* sederhana hingga aplikasi yang cukup kompleks. Salah satu keunggulannya adalah kemudahan dalam penggunaannya, terutama bagi pemula dalam pengembangan aplikasi berbasis *web*. *Flask* juga memungkinkan integrasi yang baik dengan berbagai pustaka dan alat bantu

Python lainnya, seperti ORM (Object Relational Mapping) untuk *database* dan RESTful API untuk komunikasi antar server (Budianto et al., 2023).

2.15 Penelitian Terdahulu

Beberapa penelitian sebelumnya telah menggunakan algoritma *Gradient Boosting Machine* (GBM) untuk menyelesaikan berbagai masalah klasifikasi dan prediksi. GBM dikenal memiliki kemampuan yang baik dalam menghasilkan model yang akurat, baik untuk data numerik maupun kategorikal. Penelitian-penelitian tersebut menjadi dasar dan referensi penting dalam pengembangan sistem pada penelitian ini. Ringkasan dari beberapa penelitian yang relevan ditampilkan pada tabel berikut:

Tabel 3. 1 Penelitian Terdahulu

No	Peneliti & Tahun	Judul Penelitian	Metode / Model	Hasil Utama & Relevansi
1	Mulyo & Khanif Zyen (2025)	<i>Pengaruh Hyperparameter Tuning Gradient Boosting terhadap Prediksi Pemilihan Program Studi Mahasiswa Baru</i>	Gradient Boosting Machine + Tuning	Model GBM yang dioptimalkan menghasilkan akurasi hingga 99,84% . Menunjukkan GBM efektif untuk klasifikasi berbasis data kategori dan numerik.
2	Cha et al. (2021)	<i>Application of Machine Learning Algorithms for Construction Waste Prediction</i>	GBM, Decision Tree, SVM	GBM mampu bekerja optimal bahkan pada dataset kecil dengan variabel kategorikal.
3	Shahani et al. (2021)	<i>Prediction of Rock Strength Using Gradient Boosting Regression and Other Models</i>	GBM, Regression Tree	GBM menunjukkan akurasi tinggi dalam memprediksi data geoteknik, memperkuat fleksibilitasnya untuk berbagai domain.
4	(Yıldız & Kalayci, 2024)	<i>Gradient Boosting Decision Trees on Medical Diagnosis over Tabular Data</i>	XGBoost, LightGBM, CatBoost	GBDT mengungguli metode ML tradisional dan deep learning dalam klasifikasi medis berbasis data tabular.

2.16 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk menentukan, memvisualisasikan, membangun berbagai aspek dari sebuah sistem perangkat lunak. UML sendiri memiliki fungsi sebagai alat untuk menangkap pemahaman mengenai sistem yang perlu dibangun.

Bayangkan sistem sebagai kelompok objek yang berbeda yang bekerja sama untuk melakukan pekerjaan yang berguna untuk pengguna. Bagian struktur statis mendefinisikan jenis objek yang penting untuk sistem dan bagaimana mereka terkait satu sama lain. Bagian perilaku dinamis menggambarkan bagaimana objek berubah seiring waktu dan berkomunikasi satu sama lain untuk mencapai tujuan tertentu. Dengan memodelkan sistem dari berbagai sudut pandang yang terkait, kita dapat memahami sistem tersebut untuk berbagai keperluan (Rumbaugh et al., 2021).

Alat bantu yang digunakan dalam perancangan sistem menggunakan *Unified Modelling Language (UML)* adalah sebagai berikut :

1. *Use case diagram*

Use case diagram menunjukkan bagaimana perilaku suatu sistem ketika digunakan oleh orang lain, *Use case diagram* memecah fungsionalitas sistem menjadi tindakan yang bermakna bagi pengguna (*Actor*) yang menggunakan sistem. Pengguna (*Actor*) mencakup manusia, serta sistem dan proses komputer lainnya).

Sedangkan *Actor* adalah representasi ideal dari orang, proses, atau objek eksternal yang berinteraksi dengan suatu sistem, subsistem, atau kelas. Untuk identifikasi *Actor*, harus ditentukan tugas tugas yang berkaitan

dengan peran pada konteks sistem. Orang atau sistem bisa muncul dalam beberapa peran, perlu diketahui jika *Actor* berinteraksi dengan *use case*.

Selain *Actor*, terdapat simbol simbol lain yang bisa digunakan didalam *Use Case* untuk memodelkan fungsi apa saja dari sistem yang dibangun beserta hubungan antar fungsi nya. Adapun simbol-simbol dalam *Use Case* antara lain sebagai berikut :

2. Diagram Aktivitas (*Activity diagram*)

Diagram aktivitas (*Activity diagram*) menggambarkan aliran kerja dari sebuah sistem. Diagram aktivitas dapat mencakup cabang dan bercabangnya control dalam sebuah sistem yang berjalan secara bersamaan. Cabang cabang ini mewakili aktivitas yang dapat dilakukan secara bersamaan (Rumbaugh et al., 2021). Simbol-simbol yang digunakan dalam *Activity diagram* adalah sebagai berikut :

3. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antara objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu :

4. Class Diagram (Diagram Kelas)

Class diagram adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deksripsi *class*, atribut, metode, dan hubungan dari setiap objek. *Class Diagram* bersifat statis, dalam artian diagram ini tidak menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi.

Class Diagram memiliki tiga komponen penyusun. Dapat dilihat dalam gambar dibawah ini



Gambar 2. 1 Komponen dalam Class Diagram
(Sumber : R. Setiawan, 2021)

Berikut merupakan penjelasan komponen-komponen diatas :

1. Komponen Atas

Komponen ini berisikan nama *class*. Setiap class memiliki nama yang berbeda-beda

2. Komponen Tengah

Komponen ini berisikan atribut dari *class*, komponen ini digunakan untuk menjelaskan kualitas dari suatu kelas. Atribut ini dapat ditulis dengan detail, dengan cara memasukkan tipe nilai

3. Komponen Bawah

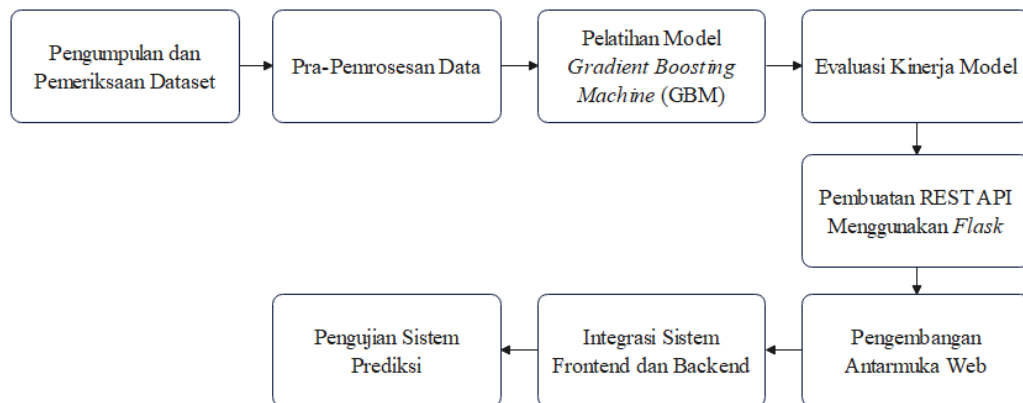
Komponen ini menyertakan operasi yang ditampilkan dalam bentuk daftar. Operasi ini dapat menggambarkan bagaimana suatu *class* dapat berinteraksi dengan data.

BAB III

METODE PENELITIAN

3.1 Pendekatan dan Alur Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan tujuan mengimplementasikan model klasifikasi berbasis *Gradient Boosting Machine* (GBM) dan menerapkannya dalam sistem prediksi penyakit jantung berbasis *web*. Untuk mendukung pendekatan tersebut, dilakukan serangkaian tahapan penelitian dapat dilihat pada gambar ini :



Gambar 3. 1 Alur Penelitian

1. Pengumpulan dan Pemeriksaan *Dataset*

Dataset penyakit jantung diunduh dari platform Kaggle. *Dataset* tersebut diperiksa terlebih dahulu untuk memastikan tidak ada nilai kosong atau data tidak valid yang dapat mengganggu proses pelatihan model.

2. Pra-Pemrosesan Data

Data dikonversi ke dalam format numerik jika masih dalam bentuk kategori (*encoding*), lalu dilakukan normalisasi agar skala fitur seragam. Setelah itu, data dibagi menjadi data latih dan data uji dengan rasio 80:20 secara acak.

3. Pelatihan Model *Gradient Boosting Machine* (GBM)

Model dibangun menggunakan bahasa pemrograman *Python* dengan pustaka *scikit-learn*. Parameter model seperti jumlah pohon dan learning rate ditentukan, lalu model dilatih menggunakan data latih.

4. Evaluasi Kinerja Model

Setelah model dilatih, dilakukan evaluasi menggunakan data uji. Hasil prediksi dibandingkan dengan label sebenarnya untuk menghasilkan metrik seperti akurasi, presisi, recall, dan F1-score.

5. Pembuatan REST API Menggunakan *Flask*

Model yang telah dilatih disimpan dalam format *.pkl* dan dihubungkan ke API menggunakan *framework Flask*. API ini berfungsi sebagai jembatan antara model dan sistem web.

6. Pengembangan Antarmuka Web

Halaman web dibangun menggunakan HTML dan PHP untuk menampilkan formulir input data serta hasil prediksi. Data yang dimasukkan pengguna akan dikirim melalui HTTP request ke API Python.

7. Integrasi Sistem *Frontend* dan *Backend*

Sistem web dan backend *machine learning* dihubungkan melalui REST API. Ketika pengguna mengisi data dan mengirimkan permintaan prediksi, data akan dikirim ke model Python untuk diproses, dan hasilnya dikembalikan ke web untuk ditampilkan.

8. Pengujian Sistem Prediksi

Sistem diuji menggunakan data simulasi untuk memastikan bahwa prediksi berhasil dijalankan, hasil tampil dengan benar, dan alur komunikasi antara frontend dan backend berjalan sesuai rencana.

3.2 Sumber dan Jenis Data

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari platform Kaggle dengan nama dataset “Heart Disease Dataset”. Dataset ini banyak digunakan dalam penelitian dan pelatihan model deteksi penyakit jantung karena berisi informasi medis dasar yang umum tersedia dalam pemeriksaan klinis. Dataset ini terdiri dari 14 atribut, dengan satu atribut target (target) yang menyatakan keberadaan penyakit jantung (1 = positif, 0 = negatif). Berikut fitur-fitur yang terdapat dalam dataset.

No	Nama Atribut	Deskripsi
1	age	Umur pasien (tahun)
2	sex	Jenis kelamin (1 = laki-laki, 0 = perempuan)
3	cp	Tipe nyeri dada (0 = typical angina, 1 = atypical angina, 2 = non-anginal, 3 = asymptomatic)
4	trestbps	Tekanan darah istirahat (mm Hg)
5	chol	Kadar kolesterol (mg/dl)
6	fbs	Gula darah puasa > 120 mg/dl (1 = ya, 0 = tidak)
7	restecg	Hasil elektrokardiogram saat istirahat (0–2)
8	thalach	Detak jantung maksimum yang dicapai
9	exang	Angina yang diinduksi oleh olahraga (1 = ya, 0 = tidak)
10	Oldpeak	Depresi ST akibat olahraga dibandingkan saat istirahat
11	slope	Kemiringan segmen ST saat latihan
12	ca	Jumlah pembuluh darah utama (0–3) yang diwarnai
13	thal	Status Thallium (1 = normal, 2 = fixed defect, 3 = reversible defect)
14	target	Diagnosis penyakit jantung (1 = ya, 0 = tidak)

3.3 Tahapan Penelitian

3.3.1. Akuisisi dan Pra-Pemrosesan Data

Langkah awal dalam penelitian ini adalah memuat dan memeriksa dataset, diikuti oleh tahap pra-pemrosesan, yang meliputi :

1. Menghapus nilai kosong (jika ada).
2. Melakukan *Encoding* terhadap atribut kategorikal
3. Normalisasi fitur numerik (jika dibutuhkan).
4. Memisahkan data menjadi data latih (80%) dan data uji (20%) secara acak.

Proses akuisisi dan pemrosesan data dilakukan menggunakan pustaka *pandas* untuk manipulasi dataset dalam bentuk *DataFrame*, dan *numpy* untuk operasi numerik. Untuk encoding data kategorikal, digunakan *LabelEncoder* dari *sklearn.preprocessing*. Proses normalisasi dilakukan menggunakan *StandardScaler* dari pustaka yang sama.

3.3.2. Perhitungan Algoritma *Gradient Boosting Machine*

Implementasi Algoritma *Gradient Boosting Machine* dilakukan dengan dua pendekatan, yaitu simulasi perhitungan manual untuk memahami prinsip dasarnya, serta implementasi penuh menggunakan Bahasa pemrograman *python* pada dataset asli secara keseluruhan. Dataset yang digunakan adalah *Heart Disease Dataset* dari Kaggle yang terdiri dari 1025 observasi dan 14 atribut, termasuk 13 fitur dan satu variabel target yang menyatakan apakah seorang pasien terdiagnosis penyakit jantung (1) atau tidak (0).

Untuk memahami prinsip dasar dari algoritma *Gradient Boosting Machine* (GBM), dilakukan simulasi perhitungan secara manual menggunakan enam sampel data dari *Heart Disease Dataset* yang tersedia di Kaggle. Sampel ini

dipilih agar mencerminkan kedua kelas target, yaitu 1 (positif penyakit jantung) dan 0 (negatif).

Tabel 3. 2 Simulasi Data Pasien

ID	age	cp	thalach	target
1	63	3	150	1
2	37	2	187	1
3	41	1	172	1
4	56	1	178	1
5	57	0	163	0
6	57	0	148	0

Model GBM dimulai menghitung rata-rata kelas target sebagai prediksi awal.

Karena terdapat 4 data positif dari 6 data total :

$$\bar{y} = \frac{\text{jumlah data dengan label positif}}{\text{jumlah seluruh data}} \dots\dots\dots (3.1)$$

$$\bar{y} = \frac{4}{6} = 0.6667$$

Selanjutnya, hitung **log-odds (logit)** awal :

$$F_0 = \log\left(\frac{\bar{y}}{1-\bar{y}}\right) \dots\dots\dots (3.2)$$

$$F_0 = \log\left(\frac{0.6667}{0.3333}\right) = \log(2) = 0.6931$$

Kemudian, estimasi probabilitas awal untuk semua data menggunakan fungsi sigmoid :

$$\hat{p}_i = \frac{1}{1 + e^{-F_0}} \dots\dots\dots (3.3)$$

$$\hat{p}_i = \frac{1}{1 + e^{-0.6931}} = 0.6667$$

Langkah berikutnya adalah menghitung nilai residual, sebagai berikut :

$$r_i^{(0)} = y_i - \hat{p}_i \dots \dots \dots (3.4)$$

Tabel 3. 3 Tabel Perhitungan Residual

ID	y_i	p_i	$r_i^{(0)}$
1	1	0.6667	0.3333
2	1	0.6667	0.3333
3	1	0.6667	0.3333
4	1	0.6667	0.3333
5	0	0.6667	-0.6667
6	0	0.6667	-0.6667

Untuk mengetahui lebih jelas performa model, dilakukan perhitungan Error untuk kuadrat residual dengan rumus dibawah ini :

$$SSE_i = \sum (r_i^{(0)})^2 \dots \dots \dots (3.5)$$

$$SSE_0 = \sum (r_i^{(0)})^2 = 4 \times (0.3333)^2 + 2 \times (-0.6667)^2$$

$$SSE_0 = 4 \times 0.1111 + 2 \times 0.4444$$

$$SSE_0 = 0.4444 + 0.8888 = 1.3332$$

Pada iterasi 0, model awalnya memprediksi semua data dengan probabilitas sama (66,67% kemungkinan terkena penyakit jantung). Residual menunjukkan bahwa pasien sebenarnya positif (target = 1) masih kurang sekitar 0.3333, sedangkan yang negative (target = 0) overestimate sebesar -0.6667. dengan SSE masih cukup tinggi, di angka 1.33332 , Kesimpulan awal pada iterasi 0 adalah model awal belum mampu membedakan pasien positif dan negatif diperlukan koreksi melalui model *boosting* selanjutnya.

Langkah berikutnya adalah memisahkan data model, berdasarkan fitur *chest pain* (cp), dalam praktiknya, seluruh fitur pada dataset dicoba hingga menemukan prediksi yang tepat dan nilai SSE yang relatif kecil, Pemisahan CP membagi dua data berdasarkan target yang dipisah dengan *region*.

Region 1 : cp = 0 (5,6)

Region 2 : cp > 0 (ID 1,2,3,4)

Variansi dihitung dengan :

$$Variansi_i = \hat{p}_i(1 - \hat{p}_i) \dots\dots\dots(3.6)$$

$$Variansi_1 = 0.6667(1 - 0.6667) = 0.6667 \times 0.3333 = 0.2222$$

Kemudian hitung gamma di setiap region :

Region 1

$$\gamma_1 = \frac{-1.3334}{2 \times 0.2222} = -3.0$$

Region 2

$$\gamma_2 = \frac{1.3332}{4 \times 0.2222} = 1.5$$

Setelah menghitung gamma, dilakukan perhitungan *update log-odds* atau logit baru untuk iterasi berikutnya dengan rumus :

$$F_m(x) = F_{m-1}(x) + v \times \gamma_j \dots\dots\dots(3.7)$$

Keterangan :

$F_m(x)$ = logit model pada iterasi ke-m

$F_{m-1}(x)$ = logit model pada iterasi sebelumnya

v = *learning rate* (biasanya 0.1 atau 0.01)

γ_j = *output* dari pohon pada region j

Dengan menggunakan *learning rate* 0.1, berikut update logit pada iterasi ke 1

Tabel 3. 4 Tabel update logit

ID	Region	Update Logit F1(x)
1	R2	$0.6931 + 0.1 \times 1.5 = 0.8431$
2	R2	0.8431
3	R2	0.8431
4	R2	0.8431
5	R1	$0.6931 + 0.1 \times -3.0 = 0.3931$
6	R1	0.3931

Selanjutnya dihitung probabilitas baru untuk setiap target kelas seperti dibawah ini :

$$\hat{p}_1 = \frac{1}{1 + e^{-F_1}}$$

Tabel 3. 5 Tabel Probabilitas baru pada iterasi ke -1

ID	F1(x)	$\hat{p}_1$
1	0.8431	0.6992
2	0.8431	0.6992
3	0.8431	0.6992
4	0.8431	0.6992
5	0.3931	0.5972
6	0.3931	0.5972

Berikut perhitungan residual yang baru

Tabel 3. 6 Nilai Residual pada iterasi ke-1

ID	ri(1)
1	0.3008
2	0.3008
3	0.3008
4	0.3008
5	-0.5972
6	-0.5972

Berikut perhitungan SSE pada iterasi ke – 1

$$SSE_1 = \sum (r_i^{(1)})^2 = 4 \times (0.3008)^2 + 2 \times (-0.5972)^2$$

$$SSE_1 = 4 \times 0.0905 + 2 \times 0.3566 = 0.3620 + 0.732 = 1.0752$$

SSE pada iterasi 1 berkurang dari 1.332 menjadi 1.0752, dapat dilihat bahwa model sudah mulai mengarahkan probabilitas lebih akurat, namun perlu *boosting* lagi untuk perbaikan lebih lanjut.

3.3.3. Implementasi dengan Python

Implementasi algoritma Gradient Boosting Machine (GBM) dilakukan dengan menggunakan bahasa pemrograman Python. Untuk membangun model prediksi penyakit jantung, penulis menggunakan pustaka scikit-learn yang menyediakan modul GradientBoostingClassifier. Berikut adalah tahapan lengkap implementasi mulai dari membaca data hingga evaluasi model.

1. Membaca dan Menyiapkan Dataset

Langkah awal adalah membaca dataset heart.csv yang digunakan dalam penelitian ini. Dataset tersebut diperoleh dari Kaggle dan berisi 14 kolom yang mencakup berbagai informasi medis seperti usia pasien, jenis nyeri dada, tekanan darah, kolesterol, dan detak jantung maksimum. Penulis memisahkan variabel target (target) sebagai label prediksi dan kolom-kolom lainnya sebagai fitur input. Dataset kemudian diperiksa untuk memastikan tidak terdapat nilai yang hilang atau tidak valid.

2. Membagi Data Menjadi Data Latih dan Uji

Setelah data disiapkan, penulis membagi dataset menjadi dua bagian: data latih dan data uji. Pembagian dilakukan dengan proporsi 80% data latih dan 20% data uji, menggunakan metode pembagian acak dengan pengaturan random_state

agar hasil dapat direproduksi. Tujuan pembagian ini adalah agar model dapat dilatih pada sebagian data, lalu diuji pada data yang belum pernah dilihat untuk mengevaluasi kemampuannya dalam melakukan generalisasi.

3. Melatih Model GBM

Pada tahap ini, penulis membangun model menggunakan kelas `GradientBoostingClassifier` dari `scikit-learn`. Parameter yang digunakan dalam model antara lain `n_estimators` sebanyak 100 (jumlah pohon yang dibangun), `learning_rate` sebesar 0.1 (untuk mengontrol kontribusi setiap pohon), dan `max_depth` sebanyak 3 (untuk membatasi kompleksitas masing-masing pohon). Model GBM dilatih menggunakan data latih dengan tujuan meminimalkan fungsi loss logistik secara bertahap, melalui penambahan pohon-pohon baru yang fokus memperbaiki kesalahan prediksi dari pohon sebelumnya.

Model *Gradient Boosting Machine* diimplementasikan menggunakan `GradientBoostingClassifier` dari pustaka `scikit-learn`, khususnya modul `sklearn.ensemble`. Model dilatih dengan parameter default, dan kemudian disesuaikan secara bertahap untuk meningkatkan akurasi.

4. Melakukan Prediksi dan Evaluasi Model

Setelah model selesai dilatih, penulis menggunakan model tersebut untuk memprediksi nilai target pada data uji. Prediksi ini kemudian dibandingkan dengan nilai target aktual untuk menghitung metrik evaluasi model. Evaluasi dilakukan menggunakan beberapa metrik penting dalam klasifikasi, yaitu akurasi, precision, recall, dan F1-score. Selain itu, confusion matrix juga digunakan untuk melihat jumlah prediksi benar dan salah dari masing-masing kelas (positif dan

negatif). Hasil evaluasi menunjukkan bahwa model GBM mampu memberikan kinerja klasifikasi yang baik pada data penyakit jantung.

5. Menganalisis Tingkat Kepentingan Fitur

Sebagai tambahan, penulis juga melakukan analisis terhadap pentingnya fitur (feature importance) untuk mengetahui fitur mana yang paling berpengaruh dalam proses prediksi. Hasil analisis menunjukkan bahwa fitur-fitur seperti jenis nyeri dada (cp), detak jantung maksimum (thalach), dan angina akibat olahraga (exang) memiliki kontribusi yang tinggi terhadap output model. Analisis ini memberikan wawasan tambahan yang dapat digunakan untuk mendukung pengambilan keputusan oleh tenaga medis.

3.3.4. Evaluasi Model

Evaluasi model merupakan tahap penting dalam pengembangan sistem berbasis *machine learning* karena menentukan seberapa baik model bekerja dalam melakukan klasifikasi. Pada penelitian ini, performa algoritma *Gradient Boosting Machine* (GBM) dievaluasi menggunakan beberapa metrik yang umum digunakan dalam klasifikasi biner, yaitu akurasi, presisi, *recall*, *F1-Score*, dan *Confussion Matrix*. Seluruh metrik ini dihitung menggunakan *scikit-learn*, khususnya melalui fungsi *confussion_matrix()* dan *classification_report()* dari modul *sklearn.metrics*.

1. Akurasi

Akurasi mengukur proporsi jumlah prediksi yang benar dibandingkan dengan total keseluruhan prediksi. Metrik ini memberikan gambaran umum seberapa sering model membuat prediksi yang benar.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Presisi

Presisi mengukur seberapa tepat model dalam memprediksi kelas positif, yaitu dari seluruh prediksi positif, berapa banyak yang benar-benar positif. Presisi tinggi menunjukkan bahwa model jarang memberikan prediksi positif yang salah (false positives rendah). Metrik ini penting ketika konsekuensi salah prediksi positif besar, seperti diagnosis penyakit.

$$Presisi = \frac{TP}{TP + FP}$$

3. Recall

Recall mengukur kemampuan model dalam menemukan semua kasus positif, yaitu dari seluruh data positif, berapa yang berhasil diprediksi dengan benar. Recall tinggi menunjukkan bahwa model jarang melewatkan kasus positif. Dalam konteks medis, recall sangat penting karena lebih baik “terlalu waspada” daripada tidak mendeteksi penyakit sama sekali.

$$Recall = \frac{TP}{TP + FN}$$

4. F1-Score

F1-Score adalah rata-rata harmonik dari presisi dan recall. Metrik ini memberikan penilaian yang seimbang ketika kita perlu mempertimbangkan trade-off antara presisi dan recall.

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

5. *Confussion Matrix*

Confusion matrix adalah tabel dua dimensi yang menunjukkan jumlah prediksi benar dan salah yang dibuat oleh model untuk setiap kelas. Tabel ini digunakan sebagai dasar untuk menghitung keempat metrik di atas. Dalam konteks klasifikasi biner, confusion matrix terdiri dari:

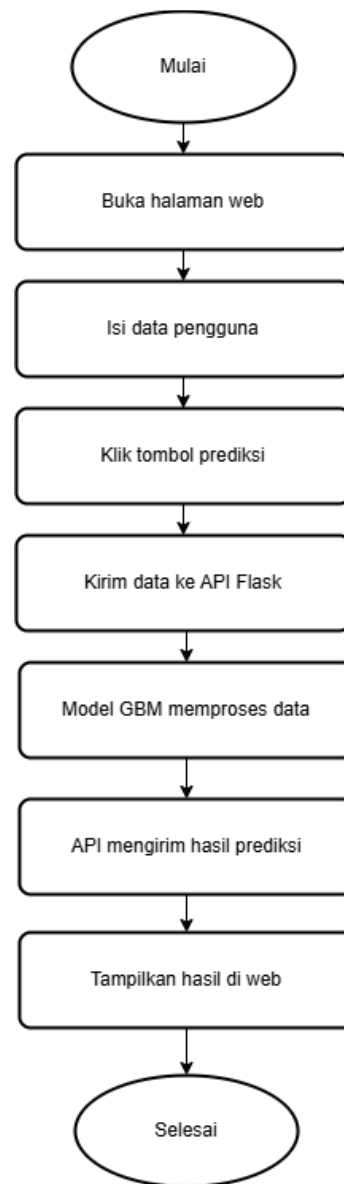
- a. TP (*True Positive*): Kasus positif yang berhasil diprediksi positif
- b. TN (*True Negative*): Kasus negatif yang berhasil diprediksi negatif
- c. FP (*False Positive*): Kasus negatif yang salah diprediksi sebagai positif
- d. FN (*False Negative*): Kasus positif yang salah diprediksi sebagai negatif

3.4 Desain Sistem

Desain sistem yang dikembangkan dalam penelitian ini merupakan aplikasi prediksi penyakit jantung berbasis *web* yang dirancang untuk digunakan secara internal oleh tenaga medis di rumah sakit. Sistem ini menggabungkan komponen frontend sebagai antarmuka pengguna untuk input data dan tampilan hasil, serta backend yang menangani pemrosesan logika dan integrasi dengan model machine learning Gradient Boosting Machine (GBM) yang telah dilatih.

3.4.1. *Flowchart Aplikasi*

Untuk menggambarkan alur kerja sistem prediksi penyakit jantung yang dikembangkan, flowchart berikut digunakan untuk menjelaskan proses interaksi antara pengguna, antarmuka web, dan *backend machine learning*. Sistem terdiri dari dua komponen utama, yaitu frontend (web) dan backend (model GBM + API), yang saling terhubung melalui REST API. Dapat dilihat pada gambar dibawah ini.



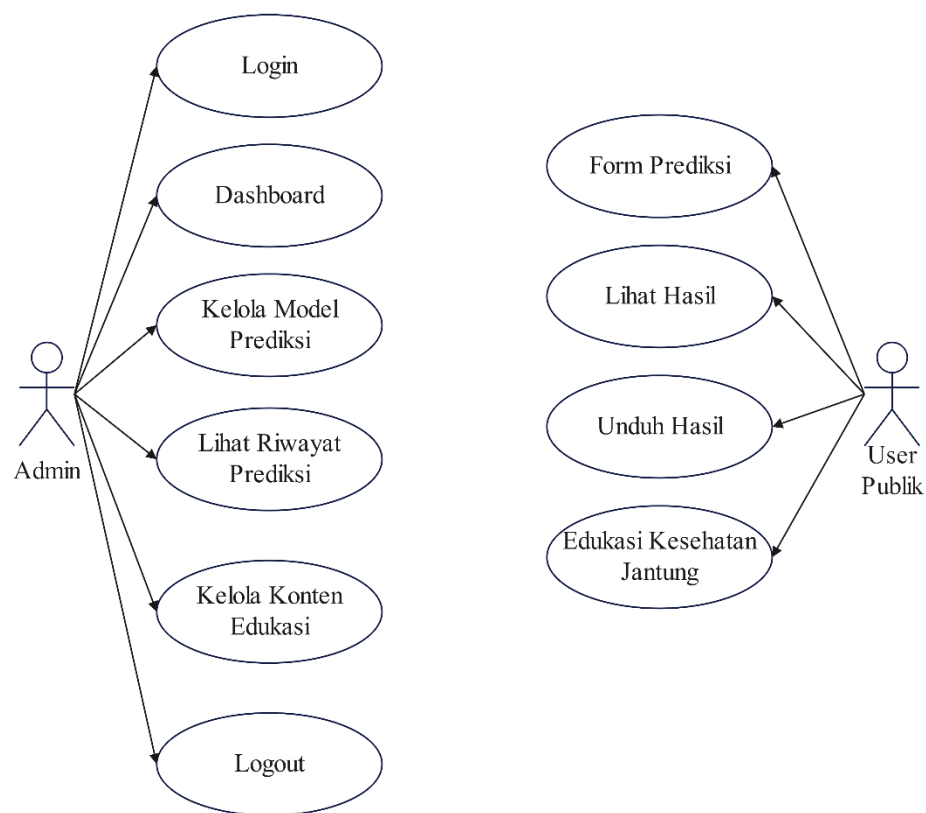
Gambar 3. 2 Flowchart Aplikasi

3.4.2. Use case diagram

Diagram dibawah ini menggambarkan dua aktor utama dalam sistem prediksi penyakit jantung berbasis *web*, yaitu **Admin** dan **User Publik**. Admin memiliki peran untuk melakukan *login*, mengakses *dashboard*, mengelola model prediksi, melihat riwayat hasil prediksi pengguna, mengatur konten edukasi, dan keluar dari sistem (*logout*). Sementara itu, User Publik dapat langsung

menggunakan fitur sistem tanpa *login*, yaitu mengisi form prediksi, melihat hasil estimasi risiko penyakit jantung, mengunduh hasil prediksi sebagai dokumen, serta mengakses konten edukasi kesehatan jantung. Diagram ini menunjukkan perancangan sistem yang terbuka dan ramah pengguna, dengan kontrol pengelolaan penuh di tangan admin.

IMPLEMENTASI GRADIENT BOOSTING MACHINE UNTUK MENDETEKSI PENYAKIT JANTUNG

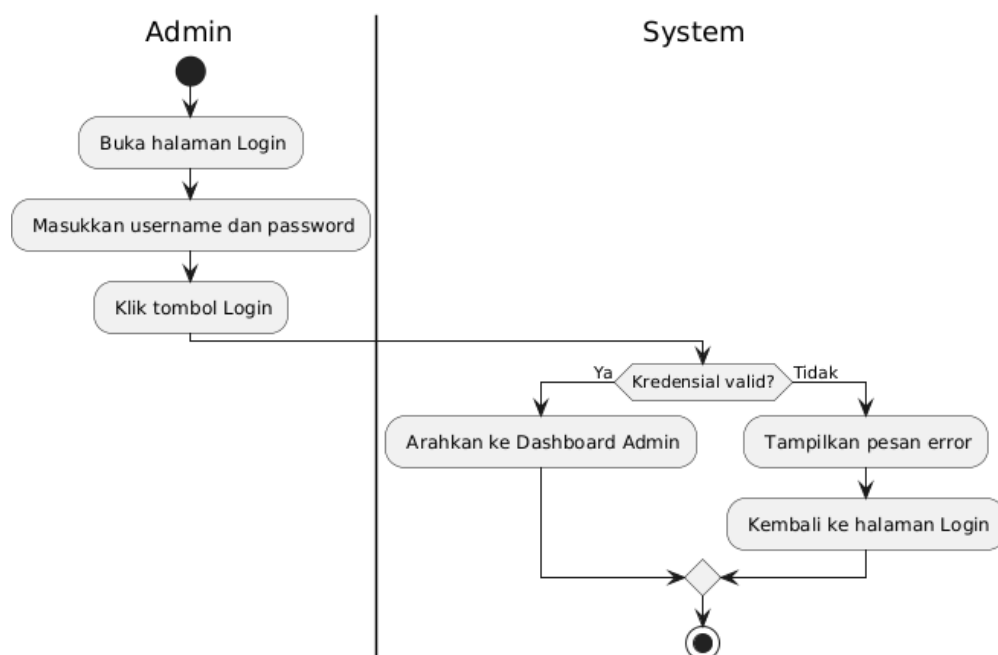


Gambar 3. 3 Use case diagram

3.4.3. Activity diagram

1. Activity diagram Login (Admin)

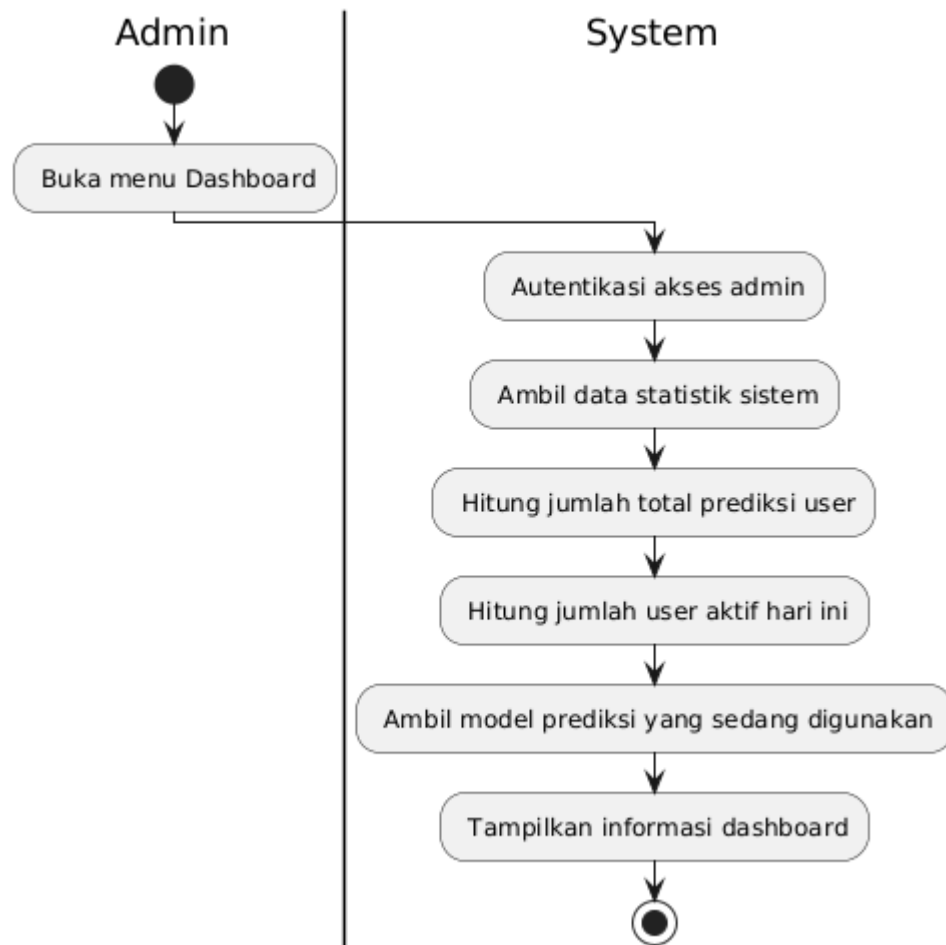
Diagram aktivitas *login* menggambarkan proses autentikasi yang dilakukan oleh Admin sebelum mengakses sistem. Pengguna memasukkan username dan password yang kemudian diverifikasi oleh sistem. Jika kredensial valid, sistem akan mengarahkan pengguna ke halaman *dashboard*; jika tidak valid, sistem akan menampilkan pesan kesalahan dan proses *login* dihentikan.



Gambar 3. 4 Activity diagram Login

2. Activity diagram Dashboard (Admin)

Diagram aktivitas *dashboard* menampilkan proses saat Admin membuka *dashboard* setelah *login*. Sistem akan menampilkan informasi ringkas terkait aktivitas pengguna dan jumlah prediksi yang telah dilakukan.

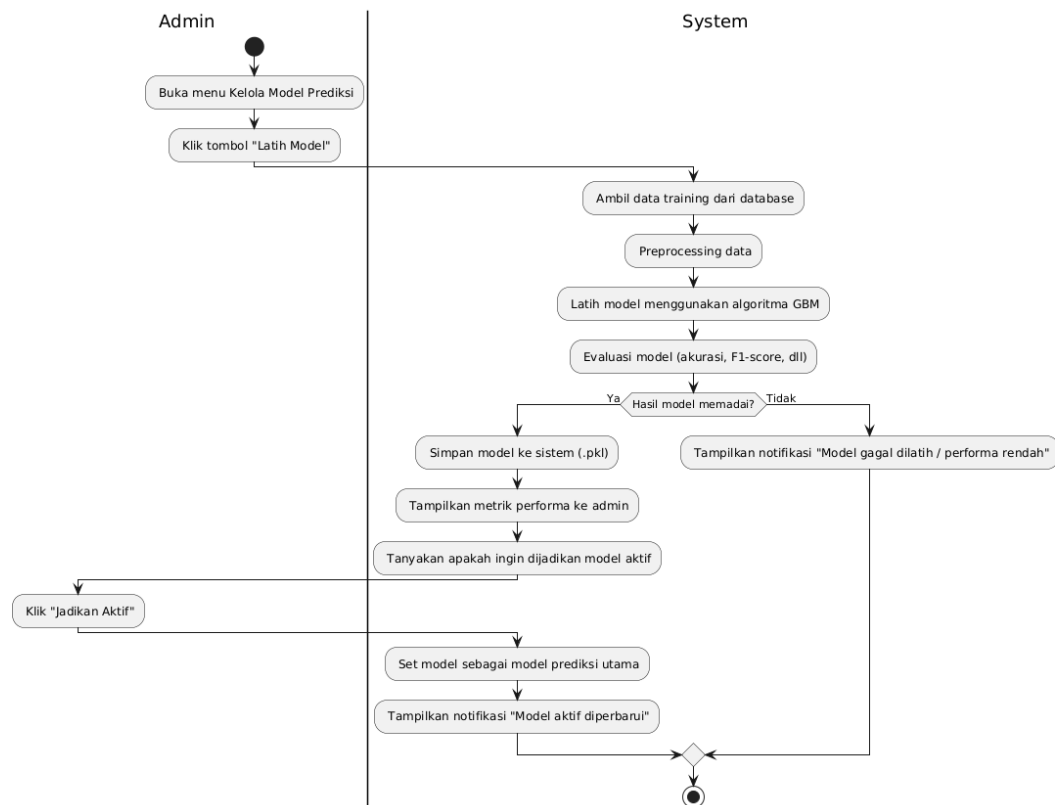


Gambar 3. 5 Activity diagram Dashboard

3. Activity diagram Kelola Model Prediksi (Admin)

Diagram ini menggambarkan dimana sistem akan otomatis melatih model GBM dari dataset yang tersedia begitu admin mengklik tombol "Latih Model". Setelah pelatihan, sistem akan mengevaluasi performa model, lalu memberikan pilihan kepada admin untuk menjadikan model tersebut sebagai model aktif. Jika disetujui, model akan digunakan dalam sistem prediksi user. Ini sangat membantu

untuk mengurangi beban teknis admin non-teknis dan menjamin bahwa sistem selalu diperbarui secara terukur.

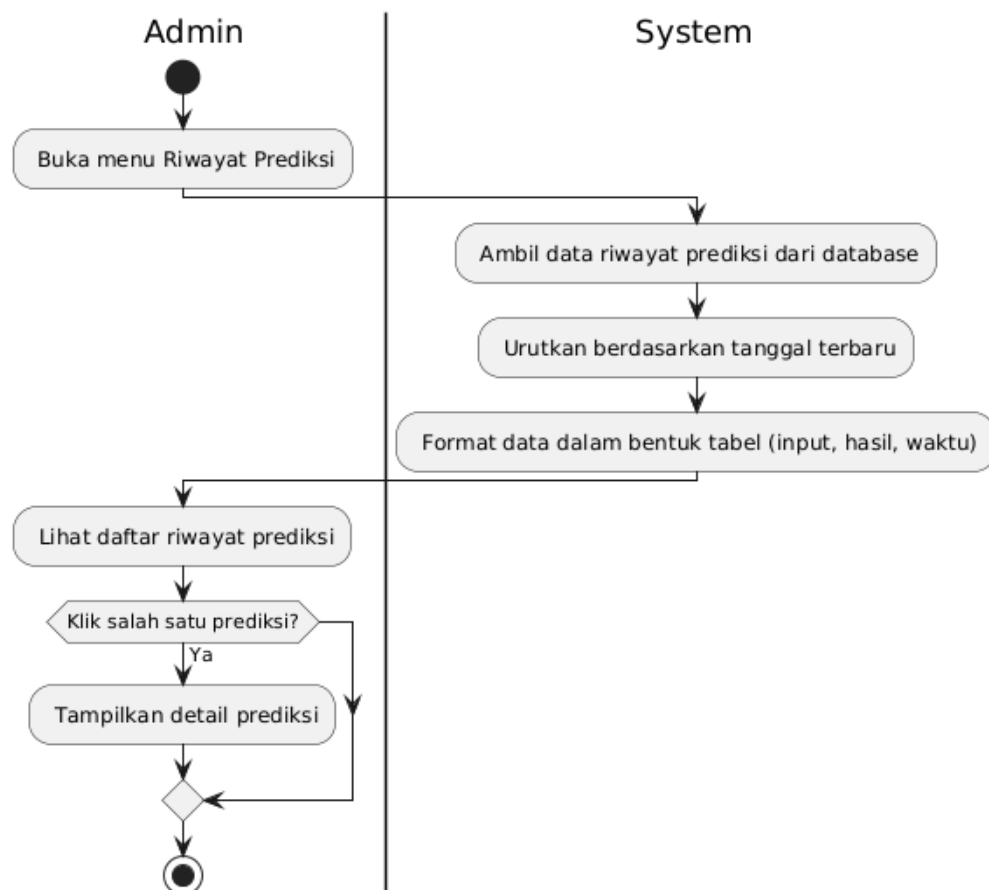


Gambar 3. 6 Activity diagram Kelola Model Prediksi

4. Activity diagram Lihat Riwayat Prediksi (Admin)

Diagram aktivitas ini menjelaskan alur saat admin mengakses menu Setelah membuka menu, sistem secara otomatis mengambil seluruh log prediksi dari database, termasuk input pengguna (usia, tekanan darah, dll), hasil prediksi (tingkat risiko), dan waktu eksekusinya. Data ditampilkan dalam bentuk tabel yang terurut dari prediksi terbaru. Admin juga dapat mengklik salah satu baris untuk melihat detail lengkap dari prediksi tersebut. Fitur ini berfungsi sebagai

monitoring historis aktivitas sistem dan dapat digunakan untuk mengevaluasi pola penggunaan oleh publik.

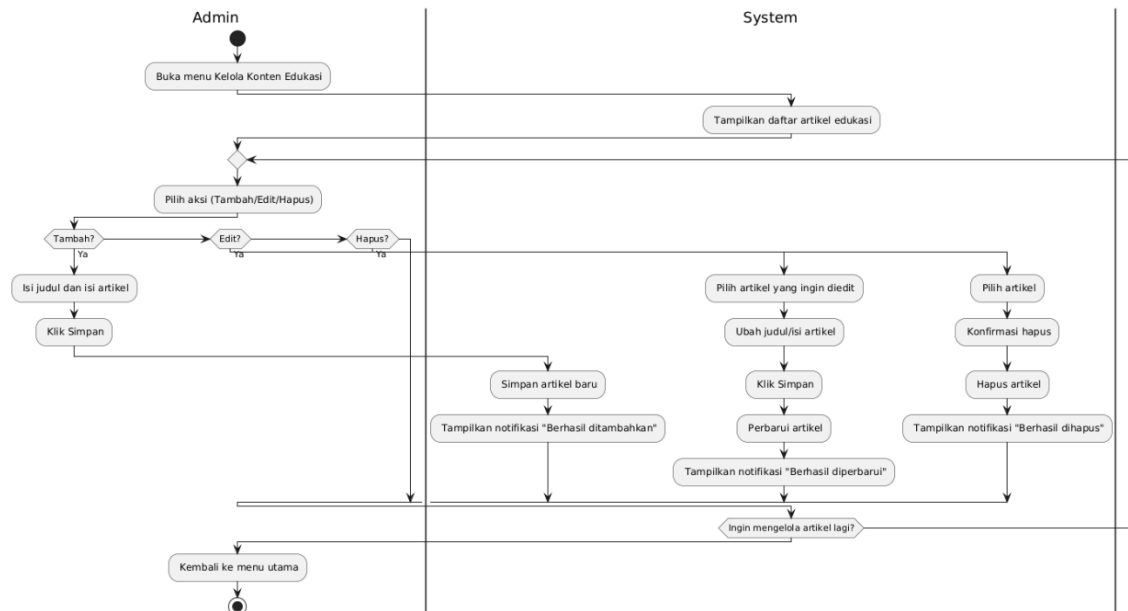


Gambar 3. 7 Activity diagram Lihat Riwayat Prediksi

5. Activity diagram Kelola Data Edukasi

Diagram ini menjelaskan bagaimana admin mengelola konten edukasi yang ditampilkan kepada user publik. Admin dapat membuka menu edukasi untuk melihat daftar artikel yang tersedia. Dari sana, admin memiliki tiga opsi utama: menambahkan artikel baru, mengedit artikel yang ada, atau menghapus artikel. Setiap aksi akan memicu sistem untuk memperbarui database dan memberikan

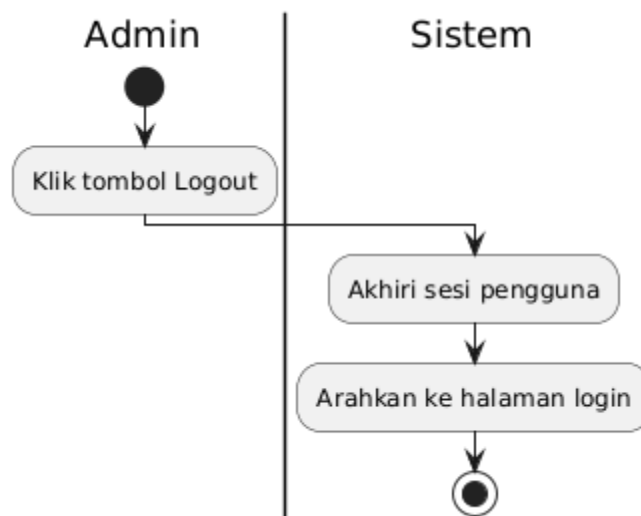
umpan balik berupa notifikasi. Fitur ini penting agar informasi yang tersedia untuk user publik selalu relevan.



Gambar 3. 8 Activity diagram Kelola Data Edukasi

6. Activity diagram Logout (Admin)

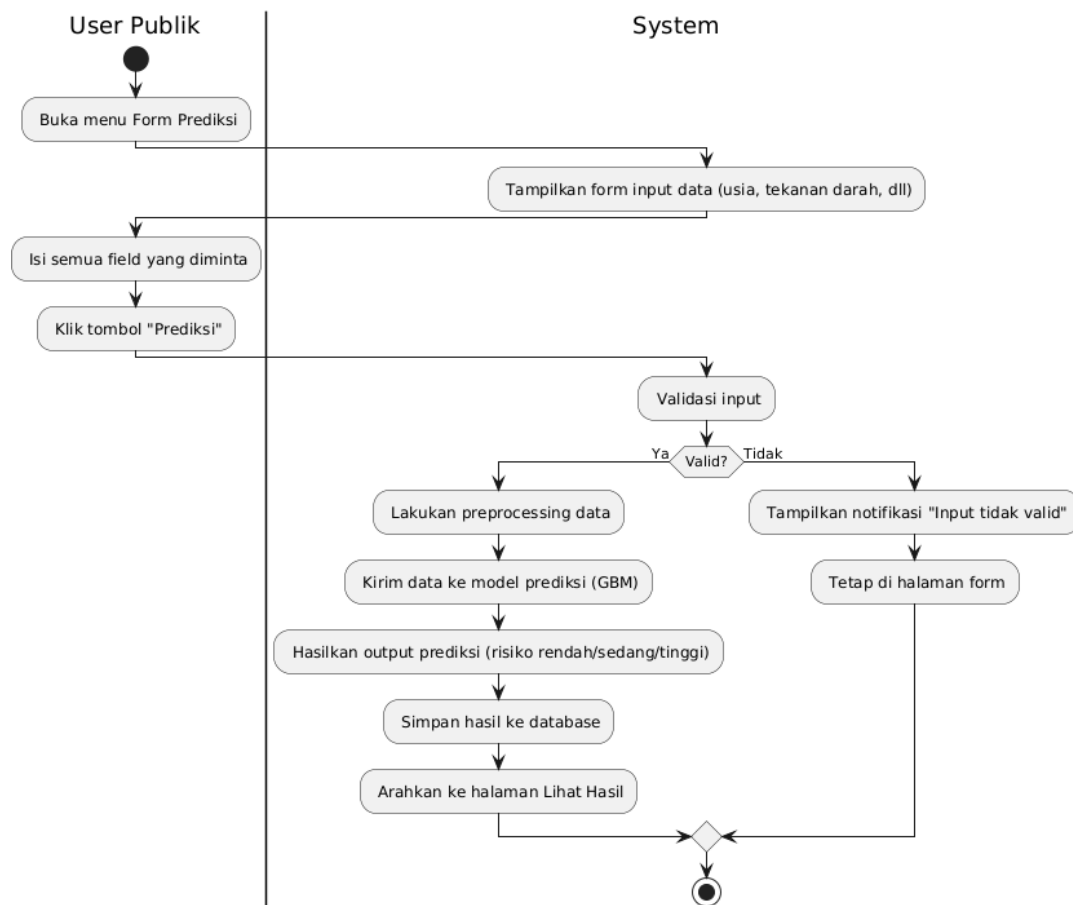
Diagram aktivitas *logout* menggambarkan proses akhir ketika Admin selesai menggunakan sistem. Pengguna memilih menu *logout*, sistem kemudian mengakhiri sesi yang sedang berjalan dan mengarahkan kembali ke halaman *login* sebagai tindakan keamanan.



Gambar 3. 9 Activity diagram Logout

7. Activity Diagram Form Prediksi (User Publik)

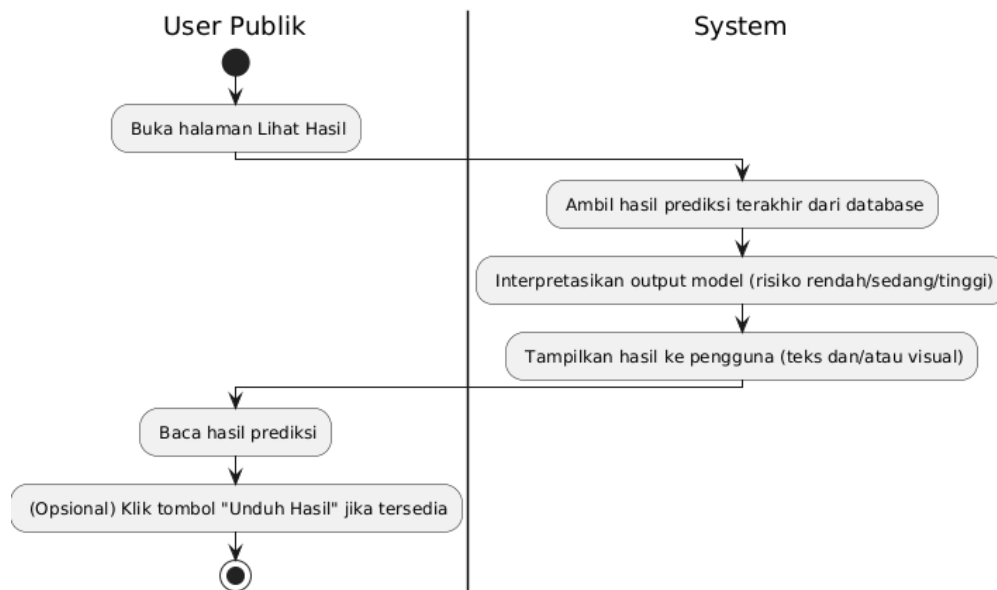
Activity diagram ini menggambarkan alur saat user publik melakukan deteksi risiko penyakit jantung. Pengguna membuka menu *Form Prediksi*, lalu mengisi sejumlah field kesehatan seperti usia, tekanan darah, kolesterol, dan lainnya. Setelah menekan tombol prediksi, sistem memvalidasi input, memproses data, menjalankan model GBM, dan menyimpan hasilnya. Jika semua berjalan lancar, user diarahkan ke halaman hasil prediksi. Jika ada input yang tidak valid, sistem memberi peringatan dan tetap berada di halaman form.



Gambar 3. 10 Activity Diagram Form Prediksi

8. Activity Diagram Lihat Hasil (User Publik)

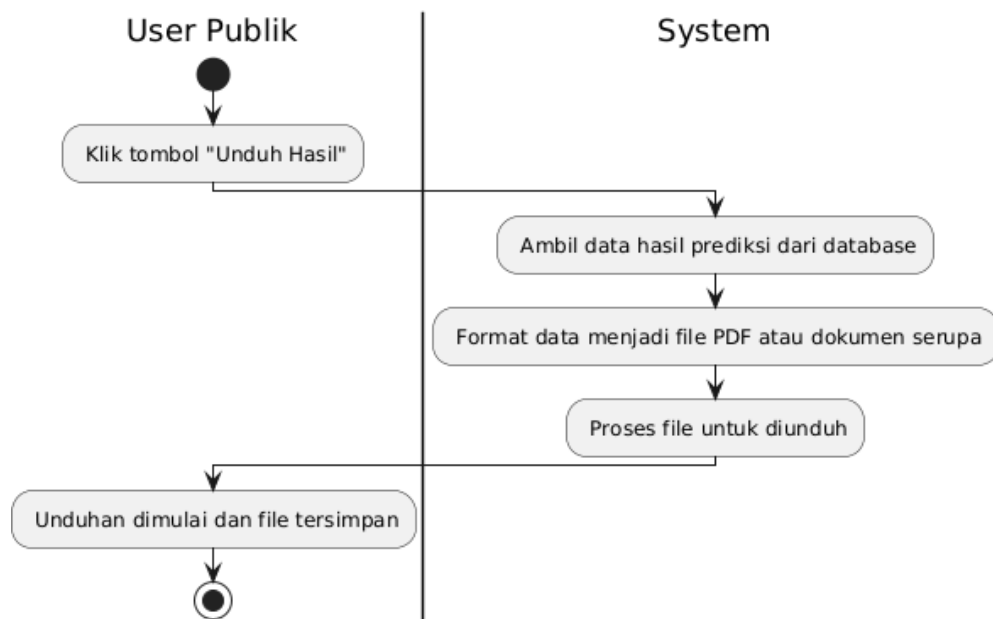
Diagram ini menunjukkan proses ketika user publik melihat hasil prediksi yang telah dilakukan sebelumnya. Sistem secara otomatis mengambil hasil prediksi terakhir dari database dan menyajikannya dalam bentuk yang mudah dipahami—baik berupa teks (misalnya: “Risiko tinggi”) maupun visualisasi sederhana (warna, ikon, dsb.). Pengguna cukup membaca hasil tersebut, dan jika tersedia, dapat melanjutkan ke opsi mengunduh hasil.



Gambar 3. 11 Activity Diagram Lihat Hasil

9. Activity Diagram Unduh Hasil (User Publik)

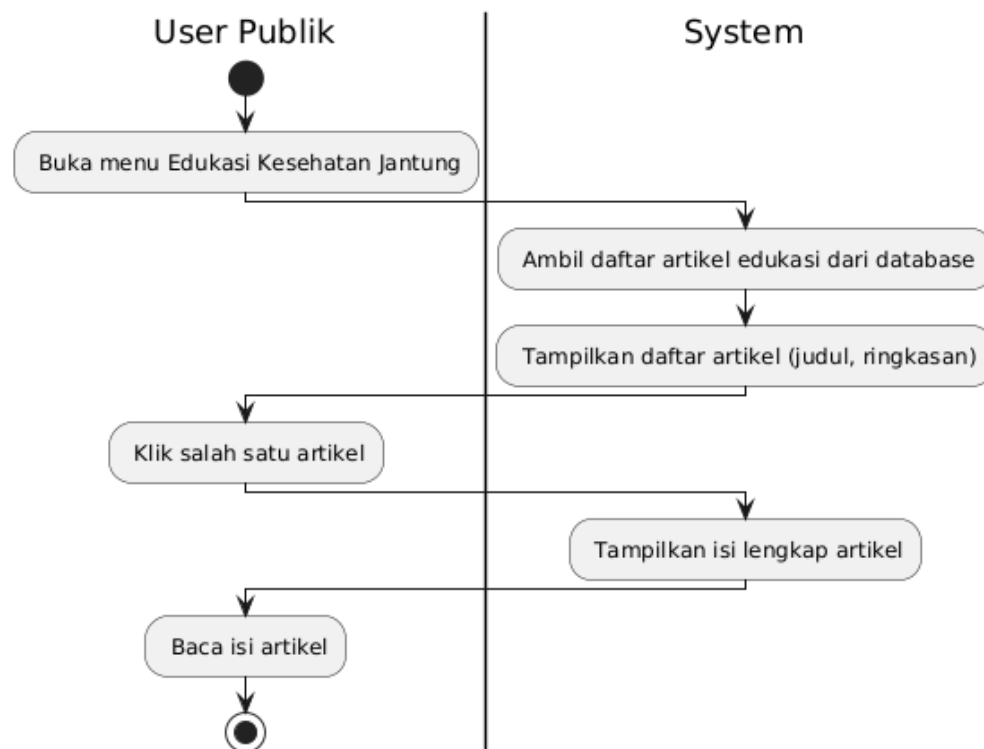
Diagram ini menggambarkan alur sederhana namun penting saat user publik ingin menyimpan hasil prediksi penyakit jantung. Setelah pengguna mengklik tombol *Unduh Hasil*, sistem mengambil data prediksi yang sudah ada, lalu memformatnya ke dalam dokumen seperti PDF. File ini kemudian dikirim ke pengguna untuk diunduh dan dapat digunakan sebagai catatan pribadi atau referensi saat berkonsultasi dengan tenaga medis.



Gambar 3. 12 Activity Diagram Unduh Hasil

10. Activity Diagram Edukasi Kesehatan Jantung (User Publik)

Activity diagram ini menjelaskan proses ketika user publik mengakses konten edukasi terkait penyakit jantung. Setelah membuka menu edukasi, sistem menampilkan daftar artikel kesehatan dari database. Pengguna kemudian dapat memilih satu artikel untuk dibaca. Sistem akan menampilkan isi artikel secara lengkap, memungkinkan pengguna memperoleh informasi yang relevan tentang pencegahan, gejala, dan gaya hidup sehat.

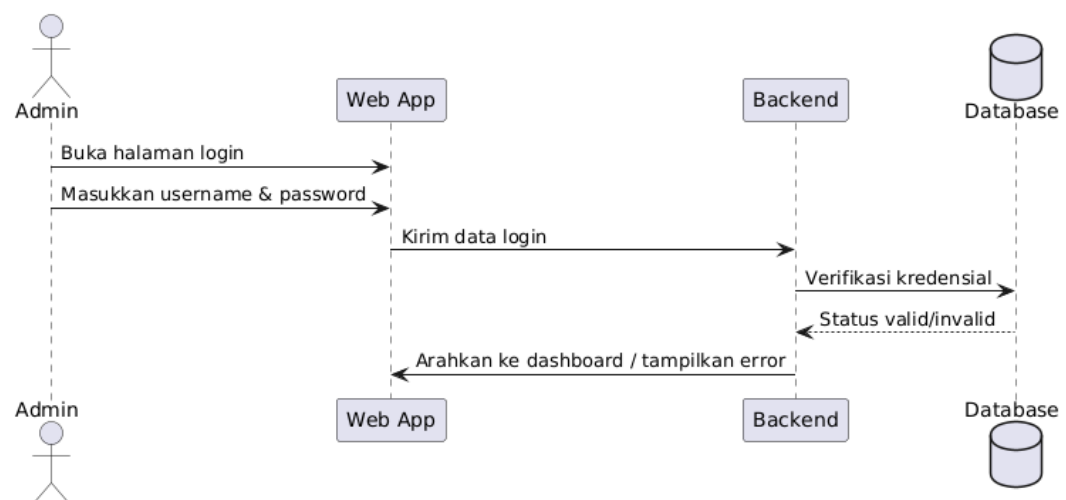


Gambar 3. 13 Activity Diagram Edukasi Kesehatan

3.4.4. Sequence Diagram

1. Sequence Diagram Login

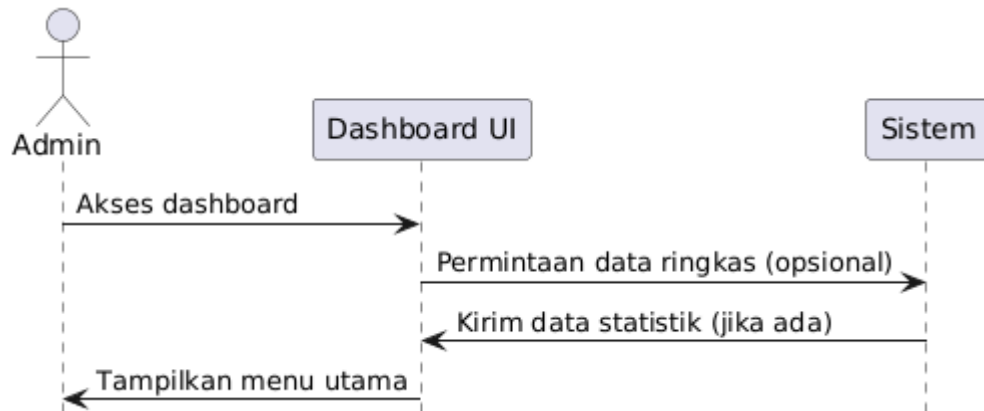
Sequence diagram ini menggambarkan alur proses autentikasi pengguna sebelum dapat mengakses sistem. Dapat dilihat pada gambar dibawah ini



Gambar 3. 14 Sequence Diagram Login

2. Sequence Diagram Dashboard

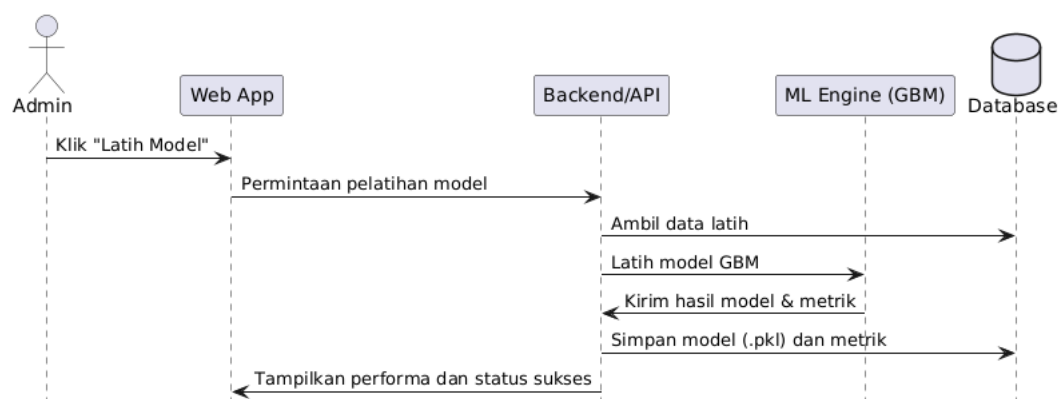
Diagram ini menunjukkan proses saat Admin mengakses *dashboard* setelah berhasil *login*, dapat dilihat pada gambar dibawah ini.



Gambar 3. 15 Sequence Diagram Dashboard

3. Sequence Diagram Kelola Model Prediksi

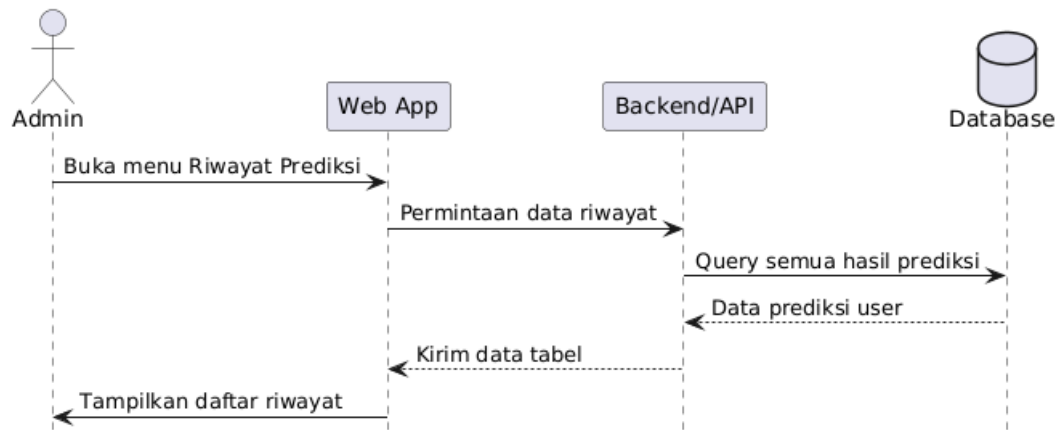
Diagram ini menjelaskan Admin hanya menekan tombol *Latih Model*, lalu sistem melatih model secara otomatis, menyimpan hasilnya, dan menampilkan evaluasi.



Gambar 3. 16 Sequence Diagram Kelola Model Prediksi

4. *Sequence Diagram* Lihat Riwayat prediksi

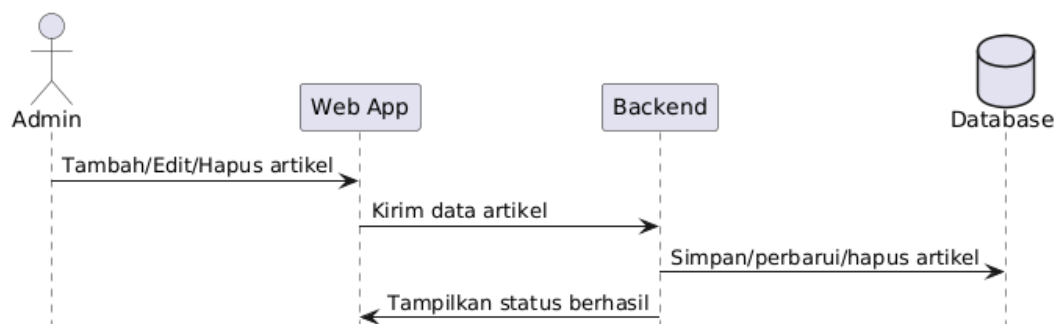
Diagram ini menggambarkan alur proses Admin meminta data log prediksi dan sistem menampilkan semua hasil dari user publik.



Gambar 3. 17 Sequence Diagram Lihat Riwayat Prediksi

5. *Sequence Diagram* Kelola Konten Edukasi

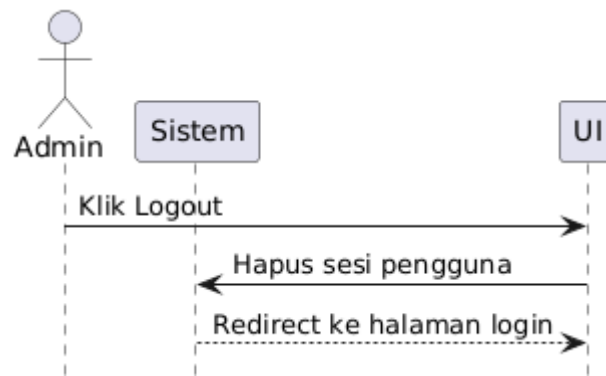
Diagram ini menjelaskan bagaimana admin mengelola konten edukasi Kesehatan jantung yang dapat diproses user publik, dapat dilihat pada gambar dibawah ini.



Gambar 3. 18 Sequence Diagram Kelola Konten Edukasi

6. *Sequence Diagram* Logout

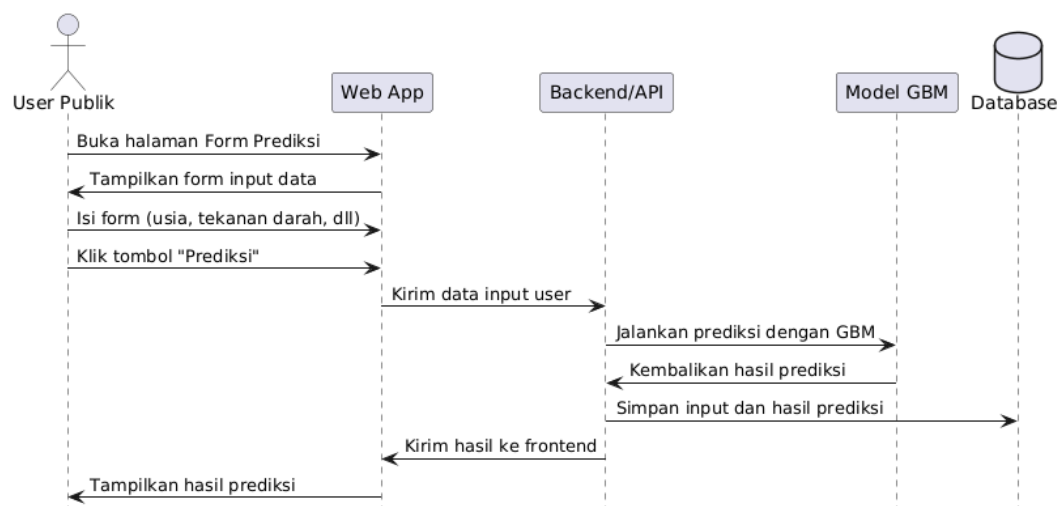
Sequence diagram ini menunjukkan langkah terakhir dalam sesi penggunaan sistem, yaitu *logout*. Dapat dilihat pada gambar dibawah ini.



Gambar 3. 19 Sequence Diagram Logout

7. Sequence Diagram Form Prediksi (User Publik)

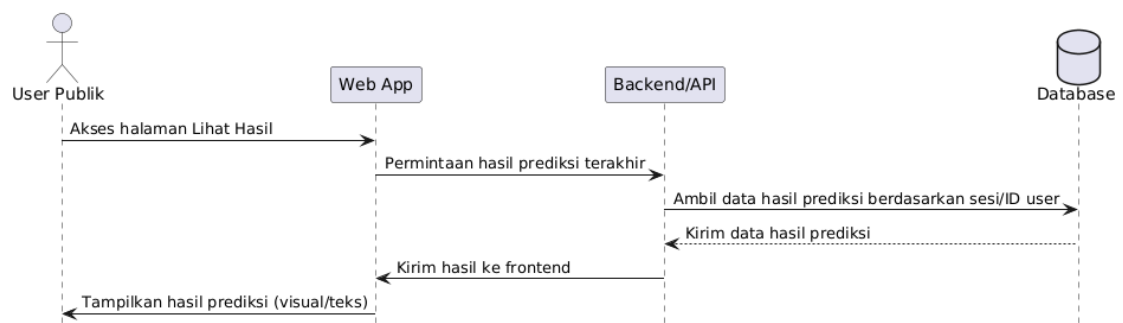
Diagram ini menggambarkan proses lengkap saat *User Publik* menggunakan fitur Form Prediksi. Setelah membuka halaman form, pengguna mengisi data kesehatan dan mengirimkan formulir. *Web App* meneruskan data tersebut ke *Backend*, yang kemudian memproses input melalui model GBM. Hasil prediksi dikembalikan ke *Backend*, disimpan ke database, dan akhirnya ditampilkan kepada user dalam bentuk visualisasi risiko.



Gambar 3. 20 Sequence Diagram Form Prediksi

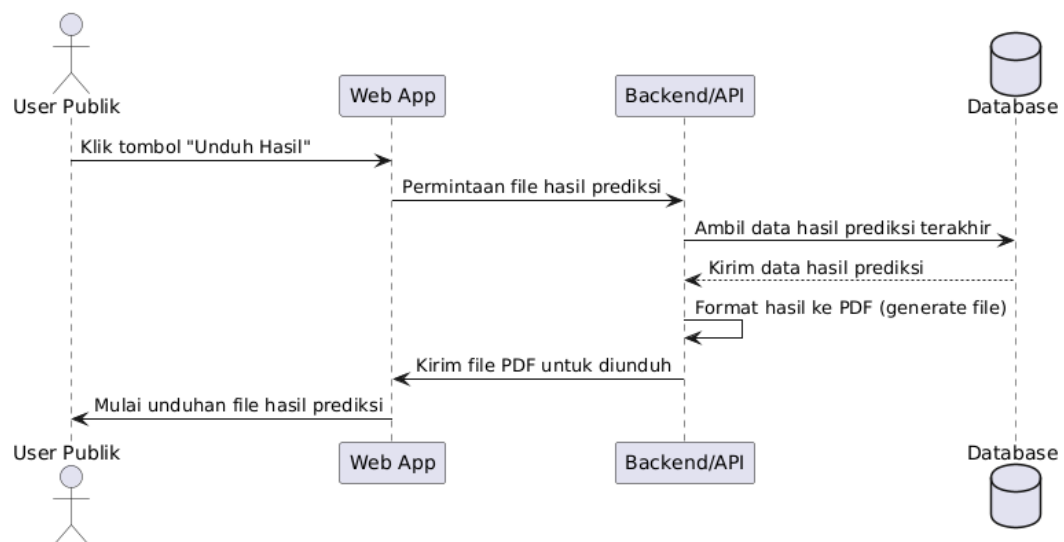
8. Sequence Diagram Lihat Hasil (User Publik)

Diagram ini menunjukkan alur saat *User Publik* membuka halaman hasil prediksi. Sistem web akan mengirim permintaan ke *backend* untuk mengambil hasil prediksi terakhir dari *database* berdasarkan sesi pengguna (tanpa *login*), kemudian menampilkannya dalam bentuk risiko penyakit jantung secara visual atau teks. Diagram ini menekankan pemisahan tanggung jawab antara *frontend*, *backend*, dan *database*.



Gambar 3. 21 Sequence Diagram Lihat Hasil

9. Sequence Diagram Unduh Hasil (User Publik)

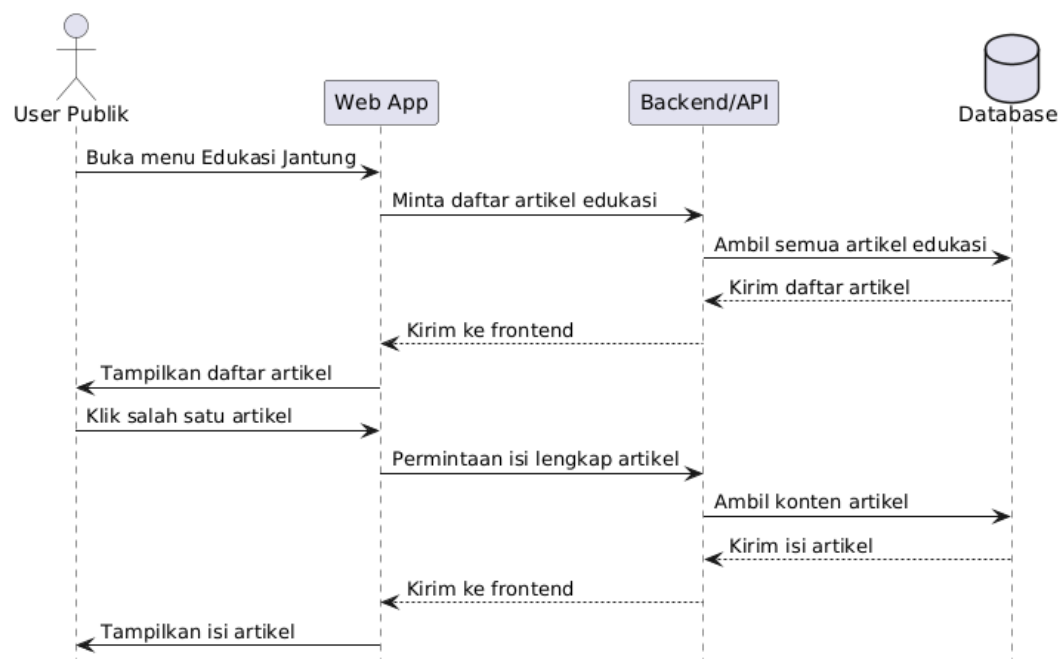


Gambar 3. 22 Sequence Diagram Unduh Hasil

Diagram ini menjelaskan proses saat User Publik mengunduh hasil prediksi penyakit jantung. Ketika tombol "Unduh Hasil" diklik, sistem akan mengambil data hasil prediksi dari database, memformatnya menjadi file PDF, lalu mengirim file tersebut ke browser user untuk diunduh. File ini bisa menjadi bukti dokumentasi atau dibawa ke konsultasi medis.

10. *Sequence Diagram* Edukasi Kesehatan Jantung (User Publik)

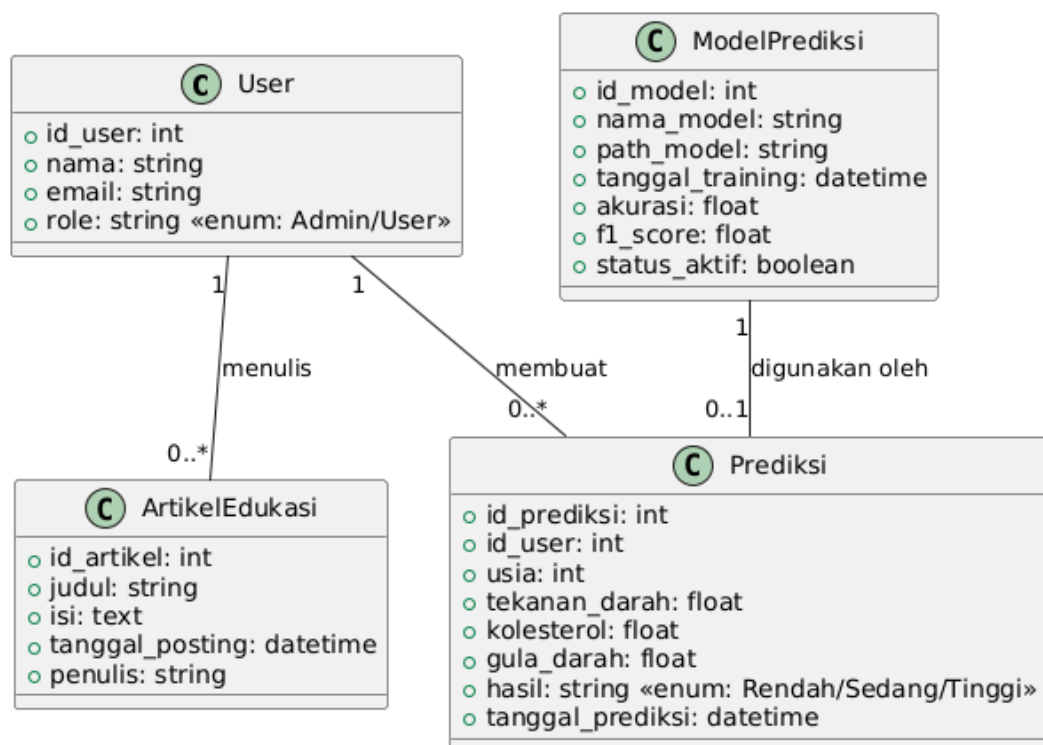
Diagram ini menggambarkan interaksi user publik saat mengakses konten edukasi tentang kesehatan jantung. Proses dimulai ketika pengguna membuka menu edukasi. Sistem menampilkan daftar artikel dari database. Saat pengguna memilih artikel tertentu, backend mengambil isi lengkap artikel dari database dan menampilkannya ke user. Diagram ini memperjelas peran sistem dalam mendukung fitur edukatif berbasis web.



Gambar 3. 23 Sequence Diagram Edukasi Kesehatan Jantung

3.4.5. Class Diagram

Class diagram ini menggambarkan struktur utama dari sistem prediksi penyakit jantung berbasis web yang terdiri dari empat entitas utama: User, Prediksi, ModelPrediksi, dan ArtikelEdukasi. Entitas User mewakili baik admin maupun pengguna publik, yang dapat melakukan prediksi dan menulis artikel edukasi. Setiap Prediksi menyimpan data input seperti usia, tekanan darah, dan kolesterol, serta hasil klasifikasi yang dihasilkan oleh model ModelPrediksi yang sedang aktif. Sementara itu, ArtikelEdukasi berisi informasi edukatif tentang kesehatan jantung yang dikelola oleh admin. Diagram ini juga menunjukkan relasi satu-ke-banyak antara user dengan prediksi dan artikel, serta keterkaitan antara prediksi dengan model yang digunakan, menggambarkan hubungan logis antar komponen dalam sistem secara terstruktur.



Gambar 3. 24 Class Diagram

3.5 Desain Basis Data

Desain basis data sistem ini bertujuan untuk menyimpan informasi pengguna (admin), data pasien, dan hasil prediksi yang dilakukan oleh model GBM. Setiap data disimpan dalam bentuk tabel yang saling terhubung dengan relasi antar kunci utama dan kunci tamu. Sistem menggunakan pendekatan basis data relasional (RDBMS) dengan struktur tabel yang dirancang untuk efisiensi pencarian, keamanan data, dan integrasi dengan sistem prediksi.

1. Rancangan Tabel Admin

Tabel admin menyimpan informasi pengguna sistem yang memiliki hak akses untuk menggunakan fitur-fitur seperti pengelolaan data pasien, pelaksanaan prediksi, dan melihat riwayat hasil prediksi. Setiap admin memiliki username dan password sebagai kredensial *login*.

Tabel 3. 7 Rancangan Tabel Admin

Kolom	Tipe Data	Keterangan
id_user	INT (PK, AI)	ID unik pengguna
nama	VARCHAR(100)	Nama lengkap
email	VARCHAR(100)	Email (bisa nullable untuk user publik anonim)
role	ENUM('Admin','User')	Peran pengguna

2. Rancangan Prediksi

Tabel prediksi merekam seluruh hasil prediksi yang dilakukan user publik. Masing-masing baris menyimpan id_prediksi sebagai primary key, data input kesehatan seperti usia, tekanan_darah, kolesterol, dan gula_darah, serta hasil klasifikasi hasil (Rendah, Sedang, atau Tinggi). Data ini juga mencatat tanggal_prediksi dan model (id_model) yang digunakan, dengan relasi ke tabel users dan model_prediksi.

Tabel 3. 8 Rancangan Tabel Prediksi

Kolom	Type Data	Keterangan
id_prediksi	INT (PK, AI)	ID unik hasil prediksi
id_user	INT (FK)	Relasi ke tabel users
usia	INT	Usia pasien
tekanan_darah	FLOAT	Tekanan darah
kolesterol	FLOAT	Kolesterol
gula_darah	FLOAT	Gula darah
hasil	ENUM('Rendah','Sedang','Tinggi')	Output prediksi
tanggal_prediksi	DATETIME	Waktu prediksi dilakukan
id_model	INT (FK)	Model yang digunakan untuk prediksi

3. Rancangan Tabel Model Prediksi

Tabel model_prediksi menyimpan informasi tentang model machine learning yang digunakan dalam sistem. Setiap model memiliki id_model sebagai identifikasi unik, nama_model dan path_model untuk menunjuk file .pkl, serta metrik performa seperti akurasi dan f1_score. Field status_aktif digunakan untuk menandai apakah model tersebut sedang digunakan oleh sistem atau tidak.

Tabel 3. 9 Rancangan Tabel Model Prediksi

Nama Kolom	Type Data	Keterangan
id_prediksi	INT	Primary key, auto increment
id_pasien	INT	Foreign key dari pasien.id_pasien
hasil	VARCHAR	Hasil klasifikasi: 'Positif' atau 'Negatif'
probabilitas	FLOAT	Probabilitas keyakinan model terhadap prediksi
waktu_prediksi	DATETIME	Waktu dan tanggal ketika prediksi dilakukan

4. Rancangan Tabel Artikel Edukasi

Tabel artikel_edukasi menyimpan konten edukatif tentang kesehatan jantung yang ditulis oleh admin. Setiap artikel memiliki id_artikel sebagai primary key, judul dan isi sebagai konten utama, serta metadata seperti tanggal_posting dan penulis. Tabel ini juga memiliki relasi ke tabel users melalui id_user untuk mengidentifikasi admin penulisnya.

Tabel 3. 10 Rancangan Tabel Artikel Edukasi

Kolom	Tipe Data	Keterangan
id_artikel	INT (PK, AI)	ID artikel
judul	VARCHAR(255)	Judul artikel
isi	TEXT	Isi lengkap artikel
tanggal_posting	DATETIME	Tanggal diterbitkan
penulis	VARCHAR(100)	Nama admin penulis
id_user	INT (FK)	ID admin dari tabel users

BAB IV

HASIL DAN PEMBAHASAN

4.1. Perhitungan *Gradient Boosting Machine*

Pada bagian ini, akan dibahas Langkah-langkah perhitungan manual dari algoritma *Gradient Boosting Machine* (GBM) yang diterapkan pada *dataset* penyakit jantung. Berikut adalah *preview* dataset yang digunakan.

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
52	male	typical angina	125	212	below 120 mg/ml	having ST-T wave abnormality	168	no	1	downsloping	2	reversible defect	no heart disease
53	male	typical angina	140	203	above 120 mg/ml	normal	155	yes	3,1	upsloping	0	reversible defect	no heart disease
70	male	typical angina	145	174	below 120 mg/ml	having ST-T wave abnormality	125	yes	2,6	upsloping	0	reversible defect	no heart disease
61	male	typical angina	148	203	below 120 mg/ml	having ST-T wave abnormality	161	no	0	downsloping	1	reversible defect	no heart disease
62	female	typical angina	138	294	above 120 mg/ml	having ST-T wave abnormality	106	no	1,9	flat	3	fixed defect	no heart disease
58	female	typical angina	100	248	below 120 mg/ml	normal	122	no	1	flat	0	fixed defect	heart disease

58	male	typical angina	114	318	below 120 mg/ml	showing probable or definite left ventricular hypertrophy	140	no	4,4	upsloping	3	normal	no heart disease
55	male	typical angina	160	289	below 120 mg/ml	normal	145	yes	0,8	flat	1	reversible defect	no heart disease
46	male	typical angina	120	249	below 120 mg/ml	normal	144	no	0,8	downsloping	0	reversible defect	no heart disease
.....													

Dataset ini digunakan untuk memprediksi adanya penyakit jantung berdasarkan sejumlah fitur medis dari pasien. Berikut adalah deskripsi lengkap mengenai kolom-kolom dalam dataset ini

1. **age:** Umur pasien (numerik, dalam tahun)
2. **sex:** Jenis kelamin pasien (kategorikal: "male" untuk laki-laki, "female" untuk perempuan)
3. **cp:** Jenis nyeri dada yang dialami pasien (kategorikal: "typical angina" untuk angina tipikal, "atypical angina" untuk angina atipikal, "non-anginal" untuk non-anginal, "asymptomatic" untuk asimtomatik)
4. **trestbps:** Tekanan darah saat istirahat (numerik, dalam mm Hg)
5. **chol:** Kadar kolesterol dalam darah (numerik, dalam mg/dl)
6. **fbs:** Gula darah puasa > 120 mg/dl (kategorikal: "above 120 mg/ml" untuk di atas 120 mg/ml, "below 120 mg/ml" untuk di bawah 120 mg/ml)

7. **restecg**: Hasil elektrokardiogram (kategorikal: "normal" untuk normal, "having ST-T wave abnormality" untuk abnormalitas gelombang ST-T)
8. **thalach**: Detak jantung maksimum yang tercapai (numerik, dalam denyut per menit)
9. **exang**: Angina yang dipicu oleh olahraga (kategorikal: "yes" untuk ya, "no" untuk tidak)
10. **oldpeak**: Depresi ST akibat olahraga dibandingkan dengan saat istirahat (numerik)
11. **slope**: Kemiringan segmen ST saat latihan (kategorikal: "upsloping" untuk naik, "downsloping" untuk turun, "flat" untuk datar)
12. **ca**: Jumlah pembuluh darah utama yang diwarnai (numerik, nilai antara 0 hingga 3)
13. **thal**: Status thalassemia (kategorikal: "normal" untuk normal, "fixed defect" untuk cacat tetap, "reversable defect" untuk cacat yang dapat dibalik)
14. **target**: Diagnosis penyakit jantung (kategorikal: "no heart disease" untuk tidak ada penyakit jantung, "heart disease" untuk ada penyakit jantung).

Dataset ini terdiri dari **1025 baris data**, dengan distribusi kelas sebagai berikut :

- a) Penyakit Jantung (*Heart Disease*) : 526 Sampel
- b) Tidak ada Penyakit Jantung (*No Heart Disease*) : 499 Sampel.

4.1.1. Proses *Label Encoding*

Label Encoding adalah teknik yang digunakan untuk mengubah data kategorikal menjadi data numerik agar dapat diproses oleh model pembelajaran mesin (*machine learning*). Banyak algoritma pembelajaran mesin, termasuk

Gradient Boosting Machine (GBM), memerlukan input dalam bentuk numerik, sehingga fitur kategorikal perlu diubah menjadi nilai numerik yang representatif.

Pada dataset ini, ada beberapa kolom yang mengandung nilai kategorikal, seperti sex, cp, fbs, restecg, exang, slope, thal, dan target. *Label Encoding* mengubah setiap kategori unik menjadi angka yang sesuai, yang memungkinkan model untuk memproses data tersebut dengan lebih efisien.

Berikut adalah penjelasan tentang bagaimana *Label Encoding* diterapkan pada kolom-kolom kategorikal di dataset ini:

Tabel 4. 1 Tabel Pemberian Label Kolom Kategori

Kolom	Kategori Asli	Kategori Encoded
sex	male, female	1, 0
cp	typical angina, atypical angina, non-anginal, asymptomatic	0, 1, 2, 3
fbs	above 120 mg/ml, below 120 mg/ml	1, 0
restecg	normal, having ST-T wave abnormality	1, 0
exang	yes, no	1, 0
slope	upsloping, downsloping, flat	2, 0, 1
thal	normal, fixed defect, reversable defect	0, 1, 2
target	no heart disease, heart disease	0, 1

Sehingga setelah pemberian label encoding dataset berubah menjadi seperti dibawah ini

Tabel 4. 2 Dataset Setelah *Label Encoding*

age	sex	cp	trestbps	cholesterol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
52	1	3	125	212	1	0	168	0	1	0	2	2	1
53	1	3	140	203	0	1	155	1	3,1	2	0	2	1
70	1	3	145	174	1	0	125	1	2,6	2	0	2	1
61	1	3	148	203	1	0	161	0	0	0	1	2	1
62	0	3	138	294	0	0	106	0	1,9	1	3	0	1
58	0	3	100	248	1	1	122	0	1	1	0	0	0
58	1	3	114	318	1	2	140	0	4,4	2	3	1	1
55	1	3	160	289	1	1	145	1	0,8	1	1	2	1
46	1	3	120	249	1	1	144	0	0,8	0	0	2	1
54	1	3	122	286	1	1	116	1	3,2	1	2	0	1
71	0	3	112	149	1	0	125	0	1,6	1	0	0	0

4.1.2. Analisa Korelasi Fitur

Untuk melakukan analisis korelasi antara fitur dalam dataset, kita menggunakan *korelasi Pearson*. *Korelasi Pearson* mengukur sejauh mana dua variabel linier berhubungan, dengan nilai antara -1 dan 1:

- 1 menunjukkan korelasi positif sempurna (ketika satu variabel meningkat, yang lain juga meningkat).
- -1 menunjukkan korelasi negatif sempurna (ketika satu variabel meningkat, yang lain menurun).
- 0 menunjukkan tidak ada korelasi linear.

Rumus *korelasi pearson* adalah sebagai berikut :

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \dots \dots \dots (4.1)$$

Langkah-Langkah:

1. Menghitung korelasi antara fitur-fitur numerik dan target menggunakan rumus *Pearson*.
2. Menggunakan Python untuk menghitung matriks korelasi antar fitur.

Berikut adalah hasil korelasi antara fitur dan **target** dalam dataset, yang dihitung menggunakan korelasi Pearson:

Tabel 4. 3 Hasil Korelasi Fitur Terhadap Target

Fitur	Korelasi terhadap Target
target	1.000000
thal	0.511025
oldpeak	0.438441
exang	0.438029
cp	0.416055
ca	0.382085
slope	0.345512
sex	0.279501
age	0.229324
restecg	0.186374

trestbps	0.138772
chol	0.099966
fbs	-0.041164
thalach	-0.422895

Fitur-fitur yang dipilih, yaitu **thal**, **oldpeak**, **exang**, **cp**, **ca**, dan **slope**, dipilih berdasarkan korelasi yang cukup tinggi terhadap target **penyakit jantung**, dengan nilai korelasi di atas **0.3**. Fitur-fitur ini memiliki hubungan signifikan dengan kemungkinan seseorang menderita penyakit jantung, yang menunjukkan bahwa faktor-faktor ini memainkan peran penting dalam proses prediksi.

1. **thal** (status thalassemia) memiliki korelasi tertinggi (0.511), menunjukkan bahwa thalassemia secara langsung berkaitan dengan risiko penyakit jantung.
2. **oldpeak** (depresi ST akibat olahraga) dan **exang** (angina yang dipicu oleh olahraga), dengan korelasi sekitar 0.44, menunjukkan bahwa gangguan pada respons jantung terhadap olahraga sangat mempengaruhi kemungkinan penyakit jantung.
3. **cp** (jenis nyeri dada), yang memiliki korelasi 0.42, adalah gejala utama yang sering terkait dengan penyakit jantung koroner.
4. **ca** (jumlah pembuluh darah yang diwarnai) dan **slope** (kemiringan segmen ST), dengan korelasi sekitar 0.38 dan 0.35, memberikan informasi tentang keparahan penyakit jantung, dengan semakin banyak pembuluh darah yang terpengaruh, semakin tinggi risiko penyakit jantung.
5. **thalach** (denyut jantung maksimal), yang memiliki korelasi negatif -0.42 dengan risiko penyakit jantung, menunjukkan bahwa denyut jantung maksimal yang lebih rendah berkaitan dengan peningkatan risiko penyakit jantung. Hal ini mungkin disebabkan oleh ketidakmampuan jantung untuk beradaptasi

dengan stres fisik yang ditandai dengan denyut jantung yang rendah, yang bisa menjadi indikator dari masalah jantung atau kinerja jantung yang buruk. Sebaliknya, denyut jantung yang lebih tinggi selama olahraga atau aktivitas fisik menunjukkan kapasitas jantung yang lebih baik dalam merespons stres, yang cenderung mengurangi risiko penyakit jantung.

Fitur-fitur ini dipilih karena masing-masing memberikan informasi penting yang berhubungan langsung dengan kondisi jantung pasien, sehingga sangat relevan untuk dimasukkan dalam model prediksi penyakit jantung.

4.1.3. Persiapan dan Pemisahan Data

Pada tahap ini, dataset yang telah melalui proses **Label Encoding** dan analisis korelasi akan dipersiapkan untuk digunakan dalam pelatihan dan pengujian model. Pemisahan data menjadi **data latih** (training data) dan **data uji** (test data) sangat penting untuk menghindari **data leakage** dan memastikan bahwa model dapat dievaluasi dengan data yang tidak digunakan selama pelatihan.

Proses Pemisahan Data:

1. **Data Latih (Training Data):** Data latih akan digunakan untuk melatih model. Biasanya, data latih terdiri dari 80% dari total dataset.
2. **Data Uji (Test Data):** Data uji digunakan untuk mengevaluasi kinerja model setelah dilatih. Data ini tidak akan terlibat dalam proses pelatihan dan berfungsi untuk mengukur akurasi model pada data yang belum pernah dilihat sebelumnya.

Proses pemisahan data dilakukan dengan menggunakan fungsi **train_test_split** dari pustaka **sklearn.model_selection**. Proses ini akan membagi dataset menjadi dua bagian:

1. **X_train** dan **y_train** untuk data latih dan label latih,
2. **X_test** dan **y_test** untuk data uji dan label uji.

Sebagai hasilnya, kita akan memiliki dua set data: satu untuk melatih model dan satu untuk menguji kinerjanya. Berikut *preview* data latih dan data uji

Tabel 4. 4 Beberapa Baris Data Latih

thal	oldpeak	exang	thalach	cp	ca	slope	target
2	1	0	168	1	0	2	1
0	1,4	1	125	0	1	0	0
2	0,3	0	141	1	0	0	1
0	0	0	184	1	0	0	0
2	0	1	156	3	0	0	1
2	0	0	163	1	0	0	0
2	0,1	0	147	3	3	0	0
0	0	0	174	1	1	1	1
2	0,8	0	150	2	0	1	1
3	1	1	156	3	0	1	1
0	0	0	152	2	0	1	0
0	0	0	160	3	1	0	1
2	1,2	0	170	3	2	1	1
0	1	0	71	3	0	1	1

Tabel 4. 5 Beberapa Baris Data Uji

thal	oldpeak	exang	thalach	cp	ca	slope	target
0	0	0	125	0	0	0	1
2	0,2	0	133	2	0	0	0
1	2,2	0	134	2	1	1	1
0	0	0	179	1	2	0	0
0	1,5	0	142	2	1	0	0
0	0	0	160	2	1	0	0
0	0	0	153	1	0	0	0
0	0	0	156	2	0	0	0
0	0	1	172	2	0	0	0
0	0	0	184	1	0	0	0
0	0	0	163	2	1	0	0

4.1.4. Standardisasi Data

Proses ini mengubah skala data sehingga fitur-fitur dalam dataset memiliki distribusi dengan rata rata 0 dan deviasi standar 1. Proses ini penting untuk algoritma seperti *Gradient Boosting Machine* yang sensitif terhadap skala data. Proses standardisasi dilakukan dengan mengurangi setiap nilai fitur dengan rata-rata fitur tersebut, dan membaginya dengan deviasi standar fitur tersebut. Hal ini memastikan bahwa setiap fitur memiliki distribusi yang seragam dan tidak dipengaruhi oleh satuan pengukuran atau skala fitur yang berbeda. Rumus *Standard Scaler* adalah sebagai berikut :

$$X_{scaled} = \frac{X - \mu}{\sigma} \dots \dots \dots (4.2)$$

Dimana :

X adalah nilai asli fitur

μ adalah rata-rata (*mean*) fitur

σ adalah deviasi standar (*standard deviation*) fitur

Perhitungan standardisasi dilakukan menggunakan Bahasa pemrograman *Python*, dengan memanfaatkan Pustaka *Scikit-Learn*. Pustaka ini menyediakan kelas *StandardScaler*, yang digunakan untuk melakukan penskalaan pada dataset sehingga setiap fitur memiliki distribusi dengan rata-rata 0 dan deviasi standar 1. Proses ini dilakukan pada seluruh data latih untuk memastikan bahwa semua fitur berada dalam skala yang seragam, yang sangat penting untuk pelatihan model. Berikut adalah hasil standardisasi untuk 10 baris pertama dari data latih menggunakan *standard scaler*.

Tabel 4. 6 Hasil standardisasi untuk 10 Data Latih

thal	oldpeak	exang	thalach	cp	ca	slope
1.158	-0.070	-0.712	0.829	-1.232	-0.731	2.203
-0.895	0.269	1.404	-1.021	-2.282	0.233	-1.014
1.158	-0.664	-0.712	-0.333	-1.232	-0.731	-1.014
-0.895	-0.919	-0.712	1.517	-1.232	-0.731	-1.014
1.158	-0.919	1.404	0.313	0.868	-0.731	-1.014
1.158	-0.919	-0.712	0.614	-1.232	-0.731	-1.014
1.158	-0.834	-0.712	-0.074	0.868	2.161	-1.014
-0.895	-0.919	-0.712	1.087	-1.232	0.233	0.594
1.158	-0.240	-0.712	0.055	-0.182	-0.731	0.594
2.185	-0.070	1.404	0.313	0.868	-0.731	0.594

4.1.5. Implementasi *Gradient Boosting Machine* dengan *Python*

Pada bagian ini, dilakukan penerapan algoritma *Gradient Boosting Machine* (GBM) untuk klasifikasi penyakit jantung menggunakan *Python* dan Pustaka *scikit-learn* menggunakan *Gradient Boosting Classifier*. *Gradient Boosting Classifier* adalah implementasi dari *Gradient Boosting Machine* yang digunakan untuk masalah klasifikasi. Metode ini bekerja dengan cara boosting, di mana setiap iterasi (setiap pohon keputusan baru) bertujuan untuk memperbaiki prediksi yang salah dari model sebelumnya. *Gradient Boosting* mengoptimalkan model dengan gradient descent untuk meminimalkan kesalahan (loss function) yang ada, yang memungkinkan model belajar dengan lebih efektif dari data yang diberikan.

Proses ini terdiri dari beberapa tahap, yang mencakup *Grid Search* untuk pemilihan parameter model, tahap *modelling* untuk pelatihan model, serta evaluasi model untuk menilai kinerja model yang telah dilatih.

1. *Grid Search* untuk Pemilihan Parameter Model

Pada tahap pertama dilakukan *Grid Search* untuk memilih parameter terbaik bagi model *Gradient Boosting Machine*. *Grid Search* adalah teknik

pencarian parameter yang memungkinkan untuk mengeksplorasi berbagai kombinasi parameter dan memilih parameter yang memberikan kinerja terbaik. Dalam penelitian ini, yang ingin dioptimalkan adalah ketiga parameter utama dari *Gradient Boosting Classifier*, yaitu :

- n_estimators* : Jumlah pohon Keputusan yang digunakan dalam model
- learning_rate* : kecepatan pembelajaran yang mengontrol pengaruh setiap pohon Keputusan terhadap model.
- Max_depth* : Kedalaman maksimum dari setiap pohon Keputusan, yang mempengaruhi kompleksitas dan kapasitas model dalam menangkap pola data

Grid Search akan mencoba setiap kombinasi parameter dan mengevaluasi kinerja model yang membagi dataset menjadi beberapa bagian untuk menghindari *overfitting* dan memastikan generalisasi yang baik pada data baru.

```
# Definisikan grid parameter untuk diuji
param_grid = {
    'classifier__n_estimators': [50, 100, 200],
    'classifier__learning_rate': [0.01, 0.05, 0.1],
    'classifier__max_depth': [3, 4, 5]
}

# Gunakan GridSearchCV pada pipeline. Scoring='recall' adalah pilihan baik untuk kasus medis
grid_search = GridSearchCV(pipeline, param_grid, cv=5, n_jobs=-1, scoring='recall', verbose=1)

# Latih!
grid_search.fit(X_train, y_train)

print(f"\nParameter terbaik: {grid_search.best_params_}")
print(f"Skor recall terbaik (cross-validation): {grid_search.best_score_:.4f}")
```

Gambar 4. 1 Implementasi *Grid Search*

Berikut merupakan hasil yang ditampilkan pada *Grid Search*, Berdasarkan hasil yang ditampilkan pada *Grid Search*, proses pencarian parameter terbaik telah selesai dilakukan dengan menggunakan *cross-validation* sebanyak 5 kali

untuk setiap kombinasi parameter yang diuji. Dalam hal ini, *Grid Search* menguji 27 kombinasi parameter dan menghasilkan 135 percobaan pelatihan model.

```
Fitting 5 folds for each of 27 candidates, totalling 135 fits

Parameter terbaik: {'classifier__learning_rate': 0.05, 'classifier__max_depth': 5, 'classifier__n_estimators': 200}
Skor recall terbaik (cross-validation): 0.9824
```

Gambar 4. 2 Output Hasil Grid Search

Hasil dari pencarian parameter terbaik dalam *Grid Search* dapat dilihat dibawah ini :

- a. Learning_rate : 0.05
- b. Max_depth : 5
- c. N_estimators : 200

Hasil terbaik ini menunjukkan bahwa *Grid Search* secara efektif telah memilih kombinasi parameter yang optimal, yang kemudian digunakan untuk melatih model *Gradient Boosting Machine* (GBM) dengan kinerja yang lebih baik pada dataset yang ada.

2. Modelling dengan Parameter Terbaik dan Evaluasi Model

Pada bagian ini, dilakukan pelatihan model menggunakan parameter terbaik yang telah ditemukan melalui *Grid Search* dan dilakukan evaluasi model untuk menilai sejauh mana model yang dilatih dapat memprediksi dengan akurat penyakit jantung pada data uji. Evaluasi dilakukan dengan menggunakan *classification report* dan *confusion matrix* untuk memberikan gambaran yang lebih mendalam tentang kinerja model.

Setelah melakukan *Grid Search* untuk memilih kombinasi parameter terbaik, langkah pertama dalam tahap *modelling* adalah mengakses model terbaik yang dihasilkan oleh pencarian parameter tersebut. Model terbaik ini adalah hasil

dari proses pencarian yang telah mengoptimalkan parameter-parameter seperti `learning_rate`, `max_depth`, dan `n_estimators` untuk memberikan kinerja terbaik pada data pelatihan.

Setelah model terbaik dipilih, langkah berikutnya adalah menggunakan model tersebut untuk melakukan prediksi pada data uji. Data uji ini tidak digunakan dalam proses pelatihan, sehingga memberikan gambaran yang objektif tentang kemampuan model dalam menggeneralisasi pada data yang tidak terlihat sebelumnya.

```
# Ambil model terbaik
best_model = grid_search.best_estimator_

# Prediksi pada data uji
y_pred = best_model.predict(X_test)
```

Gambar 4. 3 Kode untuk *Modelling* dan Prediksi

Untuk mengevaluasi kinerja model, digunakan `classification report` yang menyajikan metrik evaluasi penting, seperti *precision*, *recall*, *F1-score*, dan support untuk setiap kelas dalam dataset. Laporan klasifikasi ini memberikan wawasan yang lebih mendalam tentang seberapa baik model dalam mengidentifikasi kedua kelas, yaitu sehat (0) dan sakit jantung (1). Metrik *precision* menunjukkan akurasi prediksi positif, *recall* mengukur seberapa banyak kasus positif yang terdeteksi, dan *F1-score* memberikan keseimbangan antara *precision* dan *recall*.

```
# Tampilkan laporan klasifikasi
print("Laporan Klasifikasi pada Data Uji:")
print(classification_report(y_test, y_pred, target_names=['Sehat (0)', 'Sakit Jantung (1)'] ))
```

Gambar 4. 4 Kode untuk Membuat Laporan Klasifikasi

Berikut merupakan hasil laporan klasifikasi yang dihasilkan setelah pengujian dengan parameter terbaik

```

=====
      Laporan Klasifikasi pada Data Uji
=====

```

	precision	recall	f1-score	support
Sehat (0)	1	0.9429	0.9706	105
Sakit Jantung (1)	0.9434	1	0.9709	100
accuracy	0.9707	0.9707	0.9707	0
macro avg	0.9717	0.9714	0.9707	205
weighted avg	0.9724	0.9707	0.9707	205

```

=====

Metrik Keseluruhan:
Akurasi: 0.9707
Macro Avg - Precision: 0.9717
Macro Avg - Recall: 0.9714
Macro Avg - F1-score: 0.9707
Weighted Avg - Precision: 0.9724
Weighted Avg - Recall: 0.9707
Weighted Avg - F1-score: 0.9707
=====

```

Gambar 4. 5 Laporan Klasifikasi

Berdasarkan hasil laporan klasifikasi diatas, berikut hasil Kesimpulan yang dapat diambil

- Sehat (0): Model menunjukkan *precision* yang sangat tinggi yaitu 1.0000, yang berarti model berhasil memprediksi seluruh sampel sehat dengan benar. *Recall* sebesar 0.9429 menunjukkan bahwa model dapat mendeteksi 94.29% pasien yang benar-benar sehat. *F1-Score* yang mencapai 0.9760 menunjukkan keseimbangan yang sangat baik antara *precision* dan *recall*, yang berarti model tidak hanya tepat dalam memprediksi pasien sehat, tetapi juga cukup baik dalam mendeteksi pasien sehat yang sebenarnya.
- Sakit Jantung (1): Pada kelas sakit jantung, model juga menunjukkan *precision* yang tinggi, yaitu 0.9434, yang menunjukkan bahwa sekitar 94.34%

dari seluruh prediksi positif yang dilakukan oleh model adalah benar. *Recall* sebesar 1.0000 menunjukkan bahwa model berhasil mendeteksi semua kasus sakit jantung dengan sempurna, tanpa ada *false negatives*. *F1-Score* sebesar 0.9709 menunjukkan keseimbangan yang baik antara *precision* dan *recall*, yang penting untuk mengidentifikasi pasien yang benar-benar menderita penyakit jantung.

Selain metrik per kelas, laporan klasifikasi juga memberikan metrik keseluruhan yang menggambarkan kinerja model secara umum :

- a. Akurasi model adalah 0.9707, yang menunjukkan bahwa model memprediksi dengan benar sebanyak 97.07% dari seluruh sampel dalam data uji.
- b. Macro Average: Menghitung rata-rata *precision*, *recall*, dan *F1-score* untuk kedua kelas, tanpa mempertimbangkan jumlah sampel dalam setiap kelas. Nilai *macro average* memberikan gambaran umum tentang kinerja model pada kedua kelas secara rata-rata.
- c. *Weighted Average* : Menghitung rata-rata *precision*, *recall*, dan *F1-score* dengan memperhitungkan jumlah sampel pada setiap kelas. Nilai *weighted average* memberikan gambaran yang lebih realistis tentang kinerja model, mengingat perbedaan jumlah sampel pada masing-masing kelas.

Secara keseluruhan, model Gradient Boosting Machine menunjukkan kinerja yang sangat baik dalam memprediksi kedua kelas target:

- a. Precision yang tinggi pada kedua kelas (Sehat (0) dan Sakit Jantung (1)) menunjukkan bahwa model sangat akurat dalam memprediksi apakah seseorang sehat atau menderita penyakit jantung.

- b. Recall yang tinggi, terutama pada kelas Sakit Jantung (1) dengan nilai 1.0000, menunjukkan bahwa model mampu mendeteksi semua kasus positif (penyakit jantung) tanpa ada yang terlewat.
- c. F1-Score yang mendekati 1.0 pada kedua kelas menunjukkan bahwa model tidak hanya akurat, tetapi juga dapat mendeteksi kasus positif dengan baik, menjaga keseimbangan antara precision dan recall.

3. Penyimpanan Model

Pada tahap akhir pemodelan, setelah model *Gradient Boosting Machine* (GBM) berhasil dilatih dengan parameter terbaik yang diperoleh melalui *Grid Search* dan dievaluasi dengan metrik yang relevan, langkah selanjutnya adalah menyimpan model yang telah dilatih. Penyimpanan model ini sangat penting untuk memastikan bahwa model yang telah dilatih dapat digunakan di masa mendatang tanpa perlu melakukan pelatihan ulang, yang bisa memakan waktu dan sumber daya yang besar, dalam konteks penelitian ini, model ini akan dihubungkan dengan sistem berbasis web.

Pada kode berikut, digunakan *joblib* untuk menyimpan model yang telah dilatih dalam format *.pkl*. *joblib* adalah pustaka yang sering digunakan dalam *Python* untuk serialisasi objek besar seperti model *machine learning*, yang memungkinkan model untuk disimpan ke dalam file dan dimuat kembali untuk digunakan tanpa memerlukan pelatihan ulang.

```
joblib.dump(best_model, 'heart_disease_final_pipeline.pkl')
print("Model pipeline berhasil disimpan sebagai 'heart_disease_final_pipeline.pkl'")
```

Gambar 4. 6 Kode Penyimpanan Model

4.2. Hasil Implementasi Sistem

4.2.1. Tampilan Menu Pengguna

1. Tampilan Menu *Home*

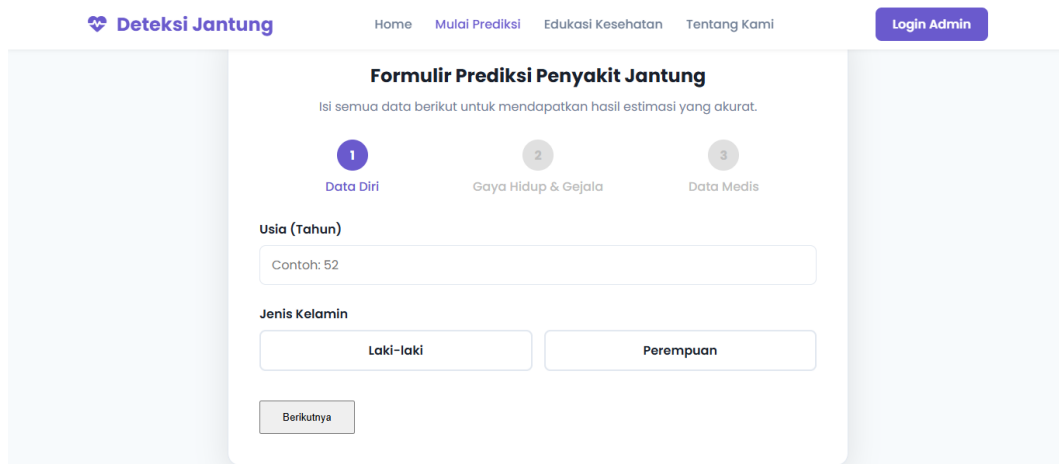
Dalam Menu Home dari sistem prediksi penyakit jantung menampilkan tombol "Mulai Prediksi Sekarang" untuk memulai prediksi risiko kesehatan. Layanan utama mencakup Prediksi Mandiri, Hasil Detail, dan Edukasi Kesehatan untuk meningkatkan pemahaman pengguna tentang risiko dan pencegahan penyakit jantung.



Gambar 4. 7 Tampilan Menu *Home*

2. Tampilan Menu Mulai Prediksi

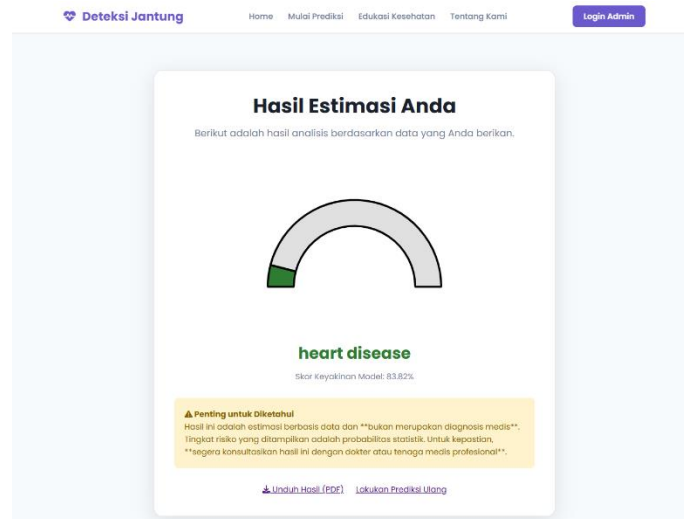
Tampilan Menu Mulai Prediksi ini merupakan tempat Dimana pengguna mengisi data diri, data gaya hidup serta data-data yang berkaitan dengan jantung. Dapat dilihat pada gambar dibawah ini.



Gambar 4. 8 Tampilan Menu Mulai Prediksi

3. Tampilan Hasil Prediksi

Tampilan menu ini memberikan hasil prediksi setelah diproses dengan algoritma *Gradient Boosting Machine*, hasil yang ditampilkan adalah *Heart Disease* (berpotensi memiliki penyakit jantung) dan *No Heart Disease* (berpotensi tidak memiliki penyakit jantung). Tampilan menu terkait dapat dilihat pada gambar dibawah ini.



Gambar 4. 9 Tampilan Hasil Prediksi

4. Tampilan Edukasi Kesehatan

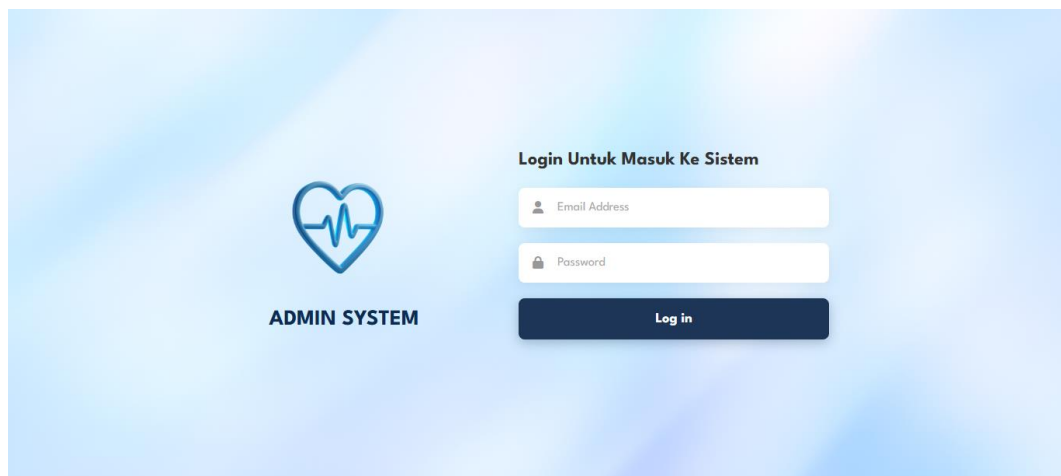
Menu Edukasi Kesehatan Jantung menyediakan informasi, tips, dan panduan untuk membantu pengguna menjaga kesehatan jantung mereka. Pada halaman ini, pengguna dapat membaca artikel terkait Kesehatan jantung. Dapat dilihat pada gambar dibawah ini.



4.2.2. Tampilan Menu Admin

1. Tampilan Menu Login

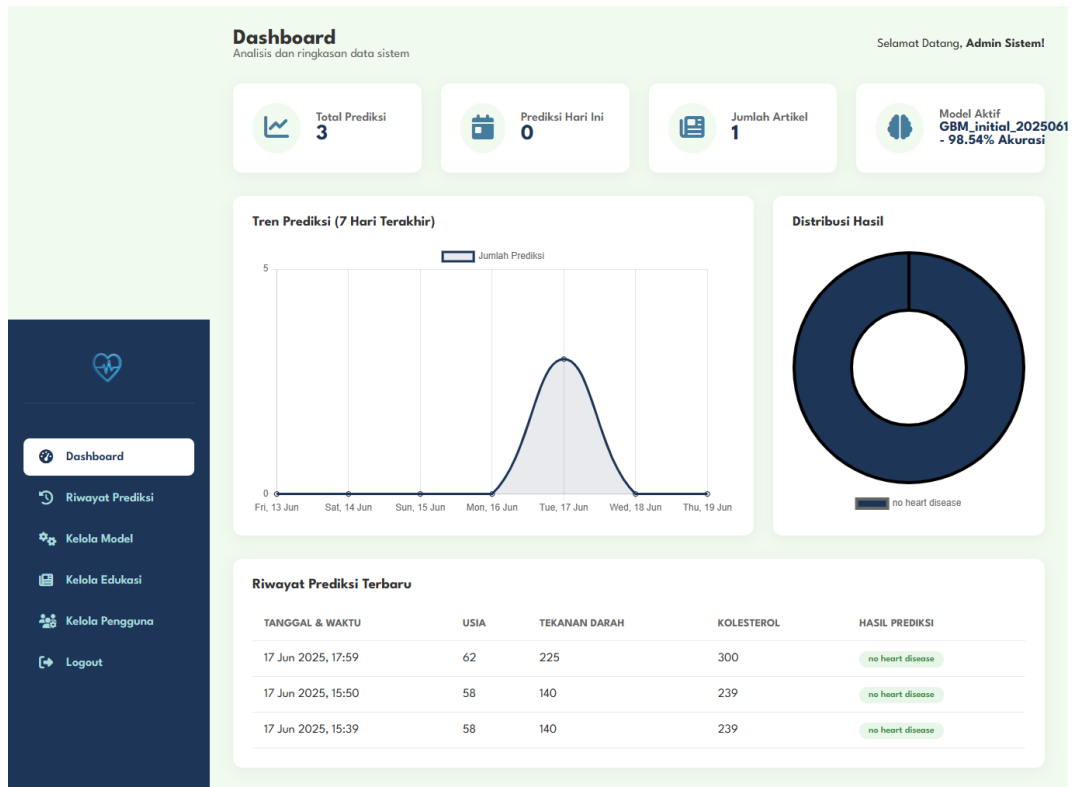
Menu Login Admin memungkinkan admin untuk mengakses sistem dengan memasukkan Email Address dan Password yang terdaftar. Setelah mengisi kolom yang disediakan, admin dapat mengklik tombol Log in untuk masuk ke sistem dan mengelola data serta fungsi administratif lainnya.



Gambar 4. 10 Tampilan Menu Login

2. Tampilan Menu Dashboard

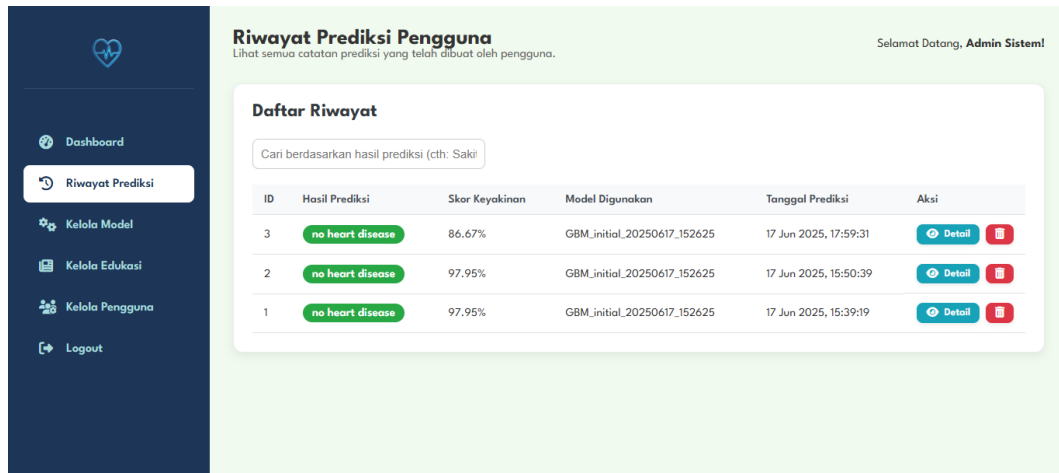
Menu Dashboard memberikan tampilan analisis dan ringkasan data sistem. Di bagian atas, terdapat informasi penting seperti Total Prediksi, Prediksi Hari Ini, dan Jumlah Artikel. Grafik Tren Prediksi (7 Hari Terakhir) menunjukkan jumlah prediksi yang dilakukan setiap hari, membantu admin melihat pola aktivitas. Selain itu, terdapat Distribusi Hasil yang menampilkan persentase hasil prediksi "no heart disease" dan "heart disease". Di bawahnya, terdapat Riwayat Prediksi Terbaru yang menunjukkan data prediksi terakhir, termasuk tanggal, usia, tekanan darah, kolesterol, dan hasil prediksi. Fitur ini memungkinkan admin untuk memantau performa sistem secara keseluruhan.



Gambar 4. 11 Tampilan Menu *Dashboard*

3. Tampilan Menu Kelola Riwayat Prediksi

Menu Riwayat Prediksi memungkinkan admin untuk melihat semua catatan prediksi yang telah dilakukan oleh pengguna. Tabel Daftar Riwayat menunjukkan ID Prediksi, Hasil Prediksi, Skor Keyakinan, Model yang Digunakan, dan Tanggal Prediksi. Admin dapat mencari riwayat prediksi berdasarkan hasil, misalnya dengan memasukkan nama. Setiap entri di tabel memiliki tombol Detail untuk melihat informasi lebih lanjut dan Hapus untuk menghapus riwayat prediksi yang tidak diperlukan. Fitur ini memungkinkan admin untuk memantau dan mengelola riwayat prediksi dengan mudah.



Gambar 4. 12 Tampilan Menu Kelola Riwayat Prediksi

4. Tampilan Menu Kelola Model

Menu Kelola Model memungkinkan admin untuk melatih, mengevaluasi, mengaktifkan dan menghapus model *machine learning* (*Gradient Boosting Machine*). Ini memastikan bahwa model yang digunakan untuk prediksi selalu diperbarui dan optimal. Berikut adalah alur kerja menu terkait :

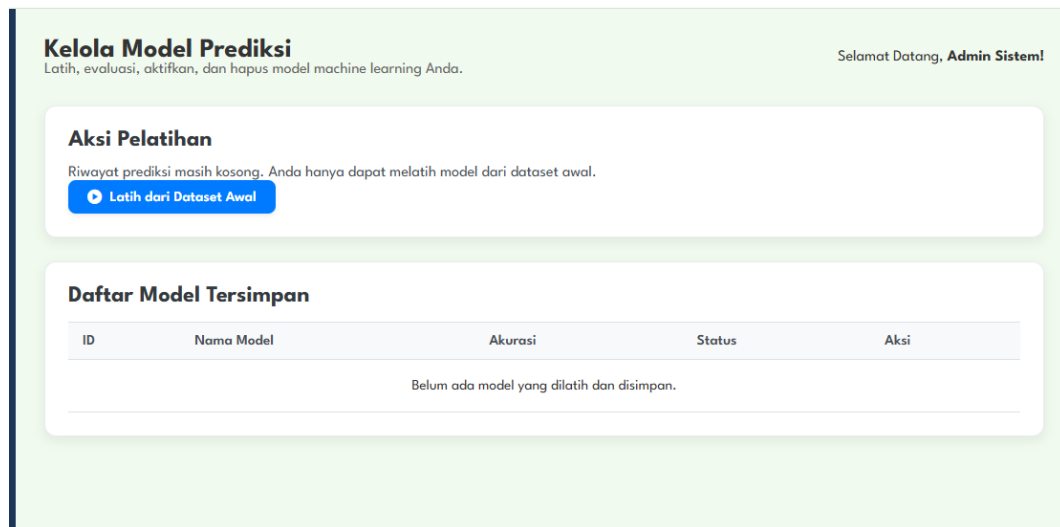
a. Inisialisasi dan Pemuatan Data Model

Ketika halaman Kelola model prediksi dimuat, sistem akan terhubung dengan *database* MySQL, kemudian, sistem akan mengambil semua data model prediksi yang tersimpan dari tabel prediksi model dan mengurutkannya berdasarkan tanggal pelatihan terbaru. Sistem juga memeriksa jumlah total data Riwayat prediksi dari tabel prediksi untuk menentukan opsi pelatihan model yang tersedia.

b. Aksi Pelatihan Model

Jika belum ada Riwayat prediksi dari pengguna, admin hanya memiliki opsi untuk mengklik tombol latih dari dataset awal. Sistem akan melatih model

Gradient Boosting Machine (GBM) menggunakan dataset awal yang telah ditentukan.



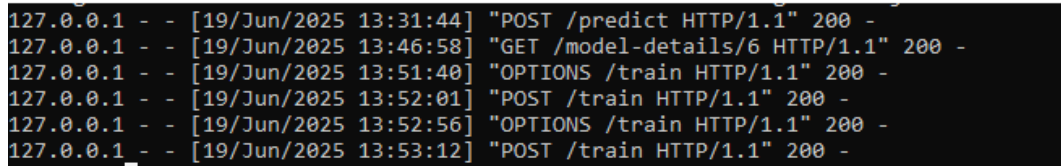
Gambar 4. 13 Tampilan jika belum ada model dan riwayat prediksi

Apabila terdeteksi ada data baru di riwayat prediksi, admin akan diberikan dua opsi, opsi pertama yaitu **Latih Ulang (Gunakan Riwayat)**, Tombol ini akan memicu modal (jendela pop-up) yang meminta admin untuk memilih strategi pelatihan ulang. Pilihan strateginya adalah "Gabungkan dengan Dataset Awal" (menggunakan *dataset* awal ditambah data riwayat baru) atau "Gunakan Hanya Data Riwayat Baru" (hanya menggunakan data dari input pengguna untuk pelatihan). Ini bertujuan untuk memperbarui model agar lebih adaptif terhadap pola data terbaru. Opsi kedua adalah **Latih dari Dataset Awal Saja** : Tombol ini akan melatih ulang model hanya dengan menggunakan *dataset* awal, mengabaikan data riwayat prediksi yang baru.

c. Proses Pelatihan (Melalui API Flask)

Ketika admin memicu pelatihan (baik "Latih dari Dataset Awal" atau "Mulai Pelatihan" dari modal), *frontend* (PHP/HTML) akan mengirimkan permintaan ke *API backend* yang dibangun dengan Python dan Flask. *Backend*

akan melakukan proses pra-pemrosesan data yang diperlukan (seperti *encoding* data kategorikal dan normalisasi fitur). Model Gradient Boosting Machine akan dilatih atau dilatih ulang menggunakan data yang relevan sesuai strategi.



```

127.0.0.1 - - [19/Jun/2025 13:31:44] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [19/Jun/2025 13:46:58] "GET /model-details/6 HTTP/1.1" 200 -
127.0.0.1 - - [19/Jun/2025 13:51:40] "OPTIONS /train HTTP/1.1" 200 -
127.0.0.1 - - [19/Jun/2025 13:52:01] "POST /train HTTP/1.1" 200 -
127.0.0.1 - - [19/Jun/2025 13:52:56] "OPTIONS /train HTTP/1.1" 200 -
127.0.0.1 - - [19/Jun/2025 13:53:12] "POST /train HTTP/1.1" 200 -

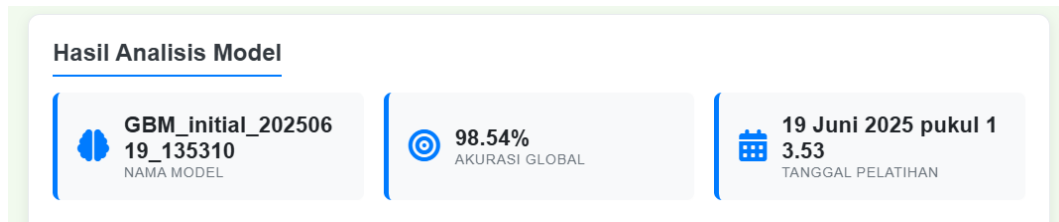
```

Gambar 4. 14 Tampilan Backend saat proses pelatihan

d. Tampilan Hasil Pelatihan & Aksi Model Baru

Hasil analisis dari model yang baru dilatih akan ditampilkan pada bagian "Hasil Analisis Model" di halaman. Ini mencakup nama model, akurasi global, tanggal pelatihan, *confusion matrix* (baik dalam bentuk tabel maupun plot jika tersedia), laporan klasifikasi rinci untuk setiap kelas (sehat dan sakit jantung), parameter *hyperparameter* terbaik yang digunakan (dari *Grid Search*), dan matriks korelasi fitur.

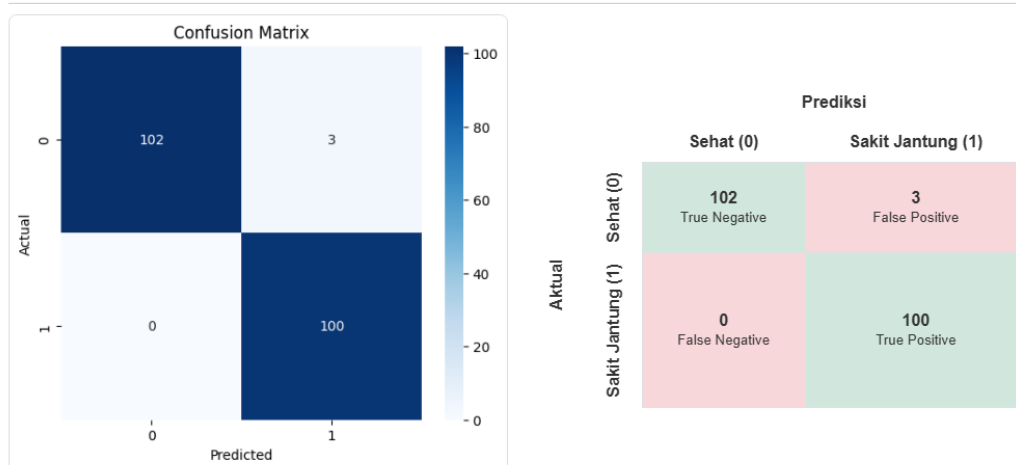
Model yang baru saja dilatih diberi identifikasi sebagai "GBM_initial_20250619_135310", yang menunjukkan bahwa ini adalah model Gradient Boosting Machine (GBM) inisial yang dilatih pada tanggal 19 Juni 2025 pukul 13:53. Model ini menunjukkan performa yang sangat mengesankan dengan mencapai akurasi global sebesar 98.54%. Angka akurasi ini mengindikasikan bahwa sebagian besar prediksi yang dibuat oleh model ini, baik untuk kasus penyakit jantung maupun tidak, adalah benar.



Gambar 4. 15 Hasil Analisis Model

Analisis lebih dalam melalui *Confusion Matrix* mengungkapkan detail kinerja model. Untuk kelas "Sehat (0)", model berhasil mengidentifikasi 102 kasus sebagai True Negative (TN), yang berarti 102 pasien yang sebenarnya sehat diprediksi dengan benar sebagai sehat. Namun, terdapat 3 kasus False Positive (FP), di mana 3 pasien yang sebenarnya sehat justru salah diprediksi mengidap penyakit jantung. Yang sangat krusial, untuk kelas "Sakit Jantung (1)", model mencatat 0 False Negative (FN), artinya tidak ada pasien yang sebenarnya menderita penyakit jantung yang terlewat atau salah diprediksi sehat. Selain itu, model berhasil mengidentifikasi 100 kasus sebagai True Positive (TP), di mana 100 pasien yang benar-benar sakit jantung diprediksi dengan benar sebagai sakit jantung.

Confusion Matrix



Gambar 4. 16 Hasil Confussion Matrix

Laporan Klasifikasi lebih lanjut memperkuat hasil ini. Untuk kelas "Sakit Jantung (1)", model memiliki nilai Precision sebesar 0.9709, yang berarti sekitar 97.09% dari semua prediksi positif penyakit jantung adalah benar. Nilai Recall mencapai 1.0000, menunjukkan kemampuan model untuk mendeteksi 100% dari semua kasus sakit jantung yang sebenarnya, menegaskan pentingnya 0 False Negative dalam konteks medis. F1-Score untuk kelas ini adalah 0.9854, yang menunjukkan keseimbangan sangat baik antara precision dan recall. Sementara itu, untuk kelas "Sehat (0)", Precision model adalah 1.0000, artinya setiap prediksi sehat yang dibuat model adalah benar. Recall-nya adalah 0.9714, menunjukkan bahwa 97.14% dari pasien yang sebenarnya sehat berhasil dideteksi. F1-Score sebesar 0.9854 juga menunjukkan performa yang sangat seimbang untuk kelas ini. Secara keseluruhan, rata-rata makro (Macro Avg) dan rata-rata berbobot (Weighted Avg) untuk Precision, Recall, dan F1-Score masing-masing berada di sekitar 0.9854 hingga 0.9858, menandakan kinerja model yang konsisten dan kuat pada kedua kelas.

Laporan Klasifikasi

Kelas	Precision	Recall	F1-Score	Support
Sakit Jantung	0.9709	1.0000	0.9852	100
Sehat	1.0000	0.9714	0.9855	105
macro avg	0.9854	0.9857	0.9854	205
weighted avg	0.9858	0.9854	0.9854	205

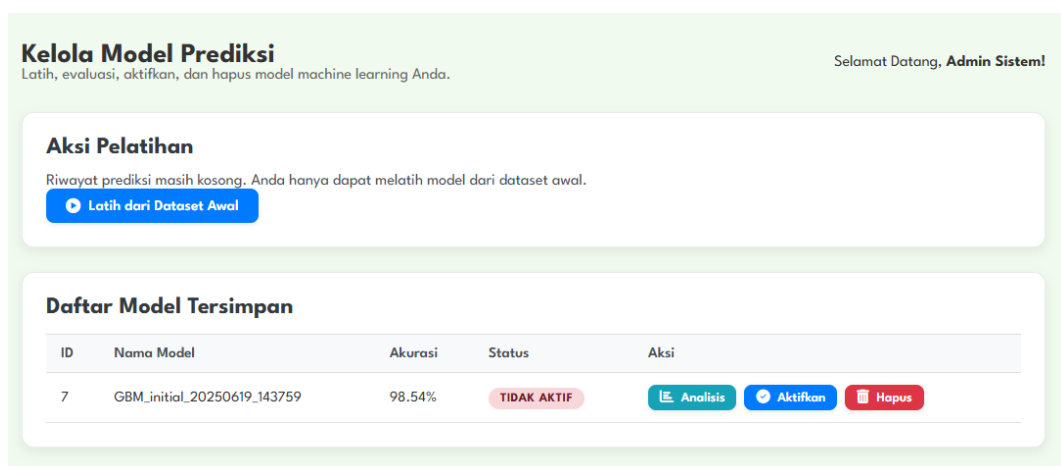
Gambar 4. 17 Laporan Klasifikasi

Admin kemudian memiliki opsi untuk **"Simpan Model Ini ke Database"**: Jika admin puas dengan performa model yang baru dilatih, mereka dapat mengklik tombol ini. Data analisis model (termasuk metrik kinerja dan parameter) akan disimpan ke *database* di tabel *prediction_models*, dan model .pkl

akan disimpan di *backend* agar dapat digunakan nanti atau memiliki menu **"Abaikan & Latih Ulang"**: Jika admin tidak puas, mereka dapat mengabaikan hasil pelatihan ini, dan bagian hasil akan disembunyikan.

e. Daftar Model Tersimpan & Aksi Manajemen

Pada bagian "Daftar Model Tersimpan", sistem menyajikan tabel komprehensif yang memuat semua model yang telah dilatih dan disimpan dalam database, menampilkan ID unik, nama model, tingkat akurasi, dan status saat ini (Aktif atau Tidak Aktif) untuk setiap entri. Dari tabel ini, administrator memiliki beberapa opsi pengelolaan: tombol "Analisis" memungkinkan pembukaan modal yang menyajikan detail analisis model secara menyeluruh, serupa dengan hasil pelatihan awal; tombol "Aktifkan" berfungsi untuk menetapkan model berstatus "Tidak Aktif" sebagai model utama yang akan digunakan untuk prediksi bagi pengguna umum, secara otomatis menonaktifkan model lain yang sebelumnya aktif; sementara itu, tombol "Hapus" memungkinkan penghapusan permanen model yang tidak lagi dibutuhkan dari database dan backend, sebuah tindakan yang tidak dapat dibatalkan.



Kelola Model Prediksi
Latih, evaluasi, aktifkan, dan hapus model machine learning Anda.

Selamat Datang, **Admin Sistem!**

Aksi Pelatihan
Riwayat prediksi masih kosong. Anda hanya dapat melatih model dari dataset awal.
[▶ Latih dari Dataset Awal](#)

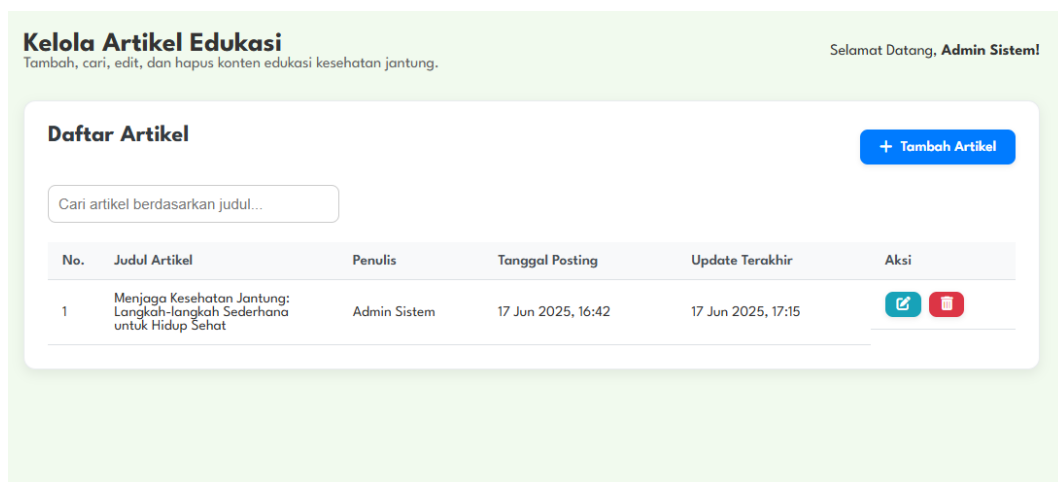
Daftar Model Tersimpan

ID	Nama Model	Akurasi	Status	Aksi
7	GBM_initial_20250619_143759	98.54%	TIDAK AKTIF	Analisis Aktifkan Hapus

Gambar 4. 18 Tampilan Daftar Model

5. Tampilan Menu Kelola Edukasi

Menu "Kelola Artikel Edukasi" berfungsi sebagai pusat administrasi bagi admin untuk mengelola konten edukasi kesehatan jantung, di mana mereka dapat menambah artikel baru, mencari artikel berdasarkan judul, mengedit informasi artikel yang sudah ada, dan menghapus artikel yang tidak lagi relevan, memastikan informasi kesehatan selalu mutakhir dan mudah diakses oleh pengguna.



Gambar 4. 19 Tampilan Menu Kelola Edukasi

6. Tampilan Menu Tambah Artikel Edukasi

Menu Tambah Artikel Edukasi yang memungkinkan admin untuk membuat konten edukasi kesehatan jantung baru. Admin dapat memasukkan judul artikel, mengunggah gambar unggulan (opsional), dan menulis isi artikel menggunakan editor teks yang disediakan, sebelum menyimpannya ke dalam sistem. Dapat dilihat pada gambar dibawah ini.

Tambah Artikel Edukasi
Buat konten baru untuk edukasi kesehatan jantung.

Selamat Datang, Admin Sistem!

Formulir Artikel Baru

Judul Artikel
Masukkan judul yang menarik...

Gambar Unggulan (Opsional)
Pilih File Tidak ada file yang dipilih

Isi Artikel

Rich text editor toolbar: Paragraph, Bold (B), Italic (I), Underline (U), Text Color, Background Color, Bulleted List, Numbered List, Link, Unlink, Code.

0 words Build with [tinyMCE](#)

[Simpan Artikel](#) [Batal](#)

Gambar 4. 20 Tampilan Menu Tambah Artikel Baru

7. Tampilan Menu Edit Artikel Edukasi

Tampilan "Edit Artikel Edukasi" ini memungkinkan admin untuk memperbarui konten artikel edukasi kesehatan jantung yang sudah ada. Admin dapat mengubah judul artikel, mengganti gambar unggulan (opsional), dan mengedit isi artikel menggunakan editor teks yang komprehensif, sebelum menyimpan perubahan.

Edit Artikel Edukasi

Perbarui konten artikel #2

Selamat Datang, Admin Sistem!

Formulir Edit Artikel

Judul Artikel

Menjaga Kesehatan Jantung: Langkah-langkah Sederhana untuk Hidup Sehat

Ganti Gambar Unggulan (Opsional)

Gambar saat ini:



Pilih File Tidak ada file yang dipilih

Kosongkan jika tidak ingin mengubah gambar.

Isi Artikel

Paragraph B I U [List Icons] [Link Icon] [Image Icon] [Table Icon] [Code Icon] [Help Icon]

Penyakit jantung merupakan salah satu penyebab utama kematian di seluruh dunia. Di Indonesia, prevalensi penyakit jantung semakin meningkat, dengan banyak faktor risiko yang berkontribusi terhadap masalah ini. Gaya hidup tidak sehat, pola makan yang buruk, dan kurangnya aktivitas fisik adalah beberapa penyebab yang dapat meningkatkan risiko terkena penyakit jantung. Oleh karena itu, menjaga kesehatan jantung sangat penting untuk mencegah penyakit jantung dan menjaga kualitas hidup.

Berikut adalah langkah-langkah sederhana yang dapat Anda lakukan untuk menjaga kesehatan jantung:

1. Menerapkan Pola Makan Sehat

Pola makan yang sehat sangat berperan dalam menjaga kesehatan jantung. Konsumsilah makanan yang kaya akan serat, seperti buah-buahan, sayuran, dan biji-bijian. Hindari makanan yang mengandung lemak jenuh, kolesterol tinggi, dan garam berlebihan. Sebaiknya, pilihlah lemak sehat, seperti yang terdapat dalam alpukat, ikan, dan kacang-kacangan.

Selain itu, perbanyak konsumsi makanan yang mengandung omega-3, seperti ikan salmon, yang dapat membantu menurunkan kadar trigliserida dan mengurangi peradangan dalam tubuh.

2. Olahraga Secara Teratur

Aktivitas fisik yang teratur membantu menjaga berat badan ideal, mengurangi tekanan darah tinggi, dan meningkatkan kadar

p Press Alt+0 for help 473 words Build with tinyMCE

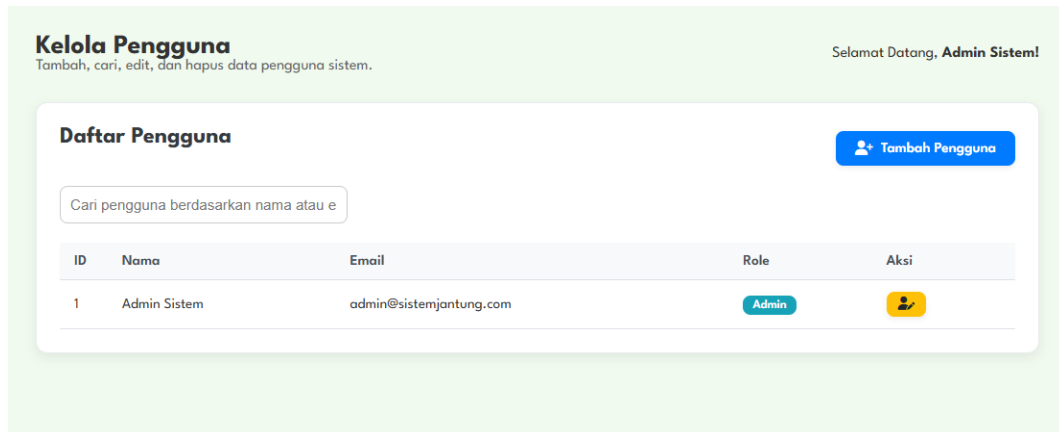
Update Artikel

Batal

Gambar 4. 21 Tampilan Edit Artikel Edukasi

8. Tampilan Menu Kelola Pengguna

Tampilan "Kelola Pengguna" ini memungkinkan Admin Sistem untuk mengelola data pengguna yang terdaftar, termasuk menambah, mencari berdasarkan nama atau email, mengedit detail pengguna, dan menghapus akun pengguna, sehingga memastikan kontrol penuh terhadap akses sistem.



Gambar 4. 22 Tampilan Menu Kelola Pengguna

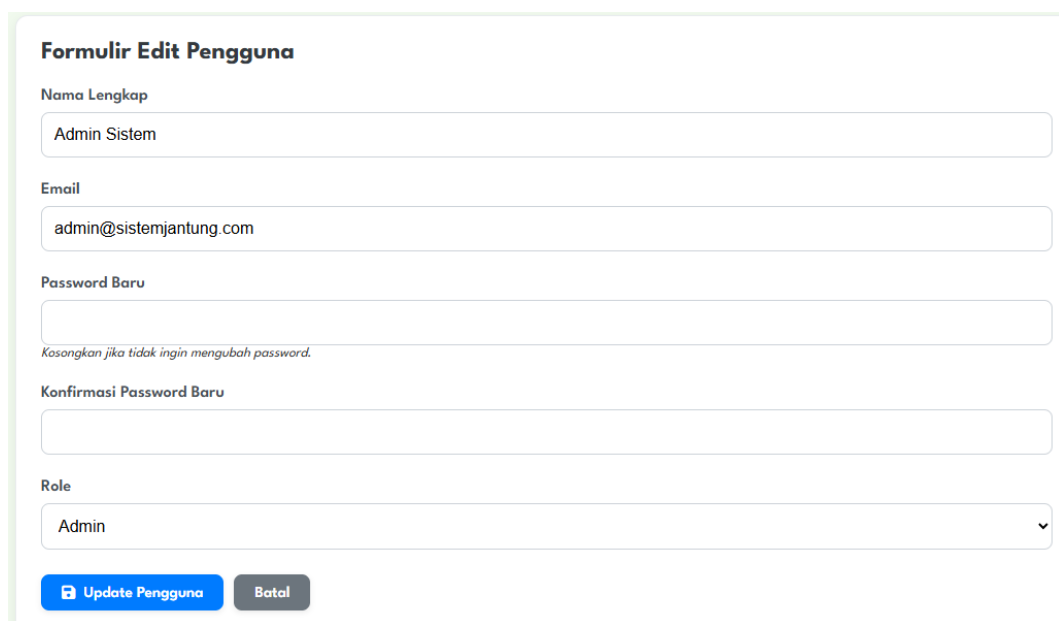
9. Tampilan Menu Tambah Pengguna

Tampilan "Tambah Pengguna Baru" ini adalah formulir yang memungkinkan Admin Sistem untuk membuat akun baru bagi pengguna dengan peran Admin atau User. Admin dapat mengisi nama lengkap, alamat email, membuat dan mengkonfirmasi kata sandi, serta memilih peran pengguna sebelum menyimpan data.

Gambar 4. 23 Tampilan Menu Tambah Pengguna

10. Tampilan Menu Edit Pengguna

Tampilan "Formulir Edit Pengguna" ini memungkinkan admin untuk memperbarui detail akun pengguna yang sudah ada, seperti nama lengkap dan email. Admin juga memiliki opsi untuk mengubah kata sandi pengguna dengan mengisi bidang "Password Baru" dan "Konfirmasi Password Baru", atau dapat membiarkannya kosong jika tidak ingin mengubahnya. Selain itu, peran (Role) pengguna juga dapat diubah melalui formulir ini, sebelum admin menyimpan perubahan dengan mengklik "Update Pengguna".



Formulir Edit Pengguna

Nama Lengkap
Admin Sistem

Email
admin@sistemjantung.com

Password Baru
Kosongkan jika tidak ingin mengubah password.

Konfirmasi Password Baru

Role
Admin

Update Pengguna **Batal**

Gambar 4. 24 Tampilan Menu Edit Pengguna

4.3. Pengujian Program

Pengujian program dilakukan untuk memastikan bahwa setiap fungsi dalam sistem prediksi penyakit jantung berbasis web ini berjalan sesuai dengan yang diharapkan. Metode pengujian yang digunakan adalah *black-box testing*, di mana pengujian difokuskan pada fungsionalitas sistem dari perspektif pengguna tanpa perlu mengetahui struktur kode internalnya. Pengujian mencakup semua fitur utama yang dapat diakses oleh *User Publik* dan *Admin*.

4.3.1. Pengujian Fungsionalitas Menu Publik

Pengujian menu yang bisa diakses publik dapat dilihat pada gambar dibawah ini.

Tabel 4. 7 Tabel Uji Menu Publik

No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Membuka halaman utama dan memulai prediksi.	Pengguna menekan tombol "Mulai Prediksi Sekarang" dan sistem menampilkan formulir prediksi multi-tahap, dimulai dari pengisian data diri.	Sistem berhasil menampilkan halaman formulir prediksi sesuai alur yang dirancang.	Berhasil
2.	Mengisi formulir prediksi dengan data yang valid.	Pengguna mengisi semua kolom yang diperlukan pada formulir prediksi, kemudian sistem memproses data melalui model GBM dan menampilkan halaman hasil estimasi risiko.	Halaman hasil prediksi berhasil ditampilkan dengan skor keyakinan model.	Berhasil
3.	Mengirimkan data yang tidak valid atau kosong pada formulir prediksi.	Sistem akan menampilkan notifikasi "Input tidak valid" dan pengguna tetap berada di halaman formulir untuk memperbaiki isian data.	Sistem berhasil melakukan validasi input dan mencegah pengguna melanjutkan proses tanpa mengisi data dengan benar.	Berhasil
4.	Mengunduh hasil prediksi.	Setelah melihat hasil, pengguna menekan tombol "Unduh Hasil (PDF)". Sistem akan menghasilkan file PDF yang berisi ringkasan hasil prediksi dan memulai proses unduh.	Fungsionalitas unduh berjalan dengan baik, dan file PDF berhasil tersimpan di perangkat pengguna.	Berhasil

5.	Mengakses dan membaca konten edukasi kesehatan.	Pengguna membuka menu "Edukasi Kesehatan", memilih salah satu artikel yang tersedia, dan sistem menampilkan isi lengkap artikel tersebut.	Sistem berhasil menampilkan daftar artikel dan kontennya secara penuh ketika dipilih.	Berhasil
----	---	---	---	----------

4.3.2. Pengujian Fungsionalitas Menu Admin

Pengujian menu yang bisa diakses admin dapat dilihat pada gambar dibawah ini.

Tabel 4. 8 Tabel Pengujian Menu Admin

No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Login dengan kredensial yang valid.	Admin memasukkan email dan password yang benar, sistem memvalidasi dan mengarahkan ke halaman Dashboard.	Admin berhasil masuk dan halaman Dashboard dengan ringkasan data sistem ditampilkan.	Berhasil
2.	Login dengan kredensial yang tidak valid.	Admin memasukkan email atau password yang salah, sistem menampilkan pesan error dan tetap berada di halaman login.	Sistem menolak akses dan menampilkan pesan kesalahan sesuai harapan.	Berhasil
3.	Mengelola Riwayat Prediksi Pengguna.	Admin mengakses menu "Riwayat Prediksi" dan sistem menampilkan seluruh catatan prediksi pengguna dalam bentuk tabel. Admin dapat melihat detail atau menghapus data.	Halaman riwayat prediksi berhasil menampilkan data dari database dengan fungsionalitas detail dan hapus yang berjalan baik.	Berhasil
4.	Melatih model baru dari dataset awal.	Admin membuka menu "Kelola Model" dan menekan tombol "Latih dari Dataset Awal". Sistem melalui API akan melatih model GBM	Proses pelatihan berhasil dipicu, dan hasil analisis kinerja model baru ditampilkan secara lengkap di antarmuka.	Berhasil

		dan menampilkan hasil analisis kinerja model (akurasi, confusion matrix, dll.).		
5.	Menyimpan dan Mengaktifkan model baru.	Setelah model dilatih, admin menyimpannya ke database. Kemudian, admin menekan tombol "Aktifkan" pada model yang baru disimpan untuk menjadikannya model prediksi utama.	Model berhasil disimpan dan statusnya berubah menjadi "TIDAK AKTIF". Setelah menekan tombol "Aktifkan", status model berhasil diperbarui menjadi "Aktif".	Berhasil
6.	Mengelola Konten Edukasi (Tambah, Edit, Hapus).	Admin dapat menambah artikel baru, mengubah artikel yang ada, dan menghapus artikel melalui menu "Kelola Edukasi".	Semua fungsi CRUD (Create, Read, Update, Delete) pada menu kelola artikel edukasi berjalan sesuai dengan yang diharapkan.	Berhasil
7.	Mengelola Pengguna (Tambah, Edit, Hapus).	Admin dapat menambah pengguna baru (admin/user), mengedit data pengguna, dan menghapusnya melalui menu "Kelola Pengguna".	Semua fungsi CRUD pada menu kelola pengguna berjalan sesuai dengan yang diharapkan.	Berhasil
8.	Logout dari sistem.	Admin menekan tombol Logout, sistem mengakhiri sesi dan mengarahkan kembali ke halaman login.	Admin berhasil keluar dari sistem dan kembali ke halaman login dengan aman.	Berhasil

4.4. Kelebihan dan Kekurangan Program

Berdasarkan hasil analisa yang telah dilakukan, berikut merupakan kelebihan dan kelemahan program terkait.

4.4.1. Kelebihan Program

Adapun kelebihan dari program yang dibuat adalah sebagai berikut :

1. Akurasi Model yang Tinggi

Model *Gradient Boosting Machine* (GBM) yang dikembangkan menunjukkan kinerja yang sangat baik dan andal untuk klasifikasi penyakit jantung. Dalam pengujian, model ini berhasil mencapai akurasi global sebesar **98,54%**. Salah satu keunggulan utamanya adalah kemampuan untuk mendeteksi semua kasus positif (pasien dengan penyakit jantung) tanpa ada yang terlewat, yang dibuktikan dengan nilai *Recall* 1.0000 untuk kelas "Sakit Jantung". Hal ini sangat krusial dalam konteks medis untuk meminimalkan risiko *false negative*.

2. Sistem Terintegrasi dan Praktis

Penelitian ini tidak hanya menghasilkan model teoretis, tetapi berhasil mengimplementasikannya ke dalam sebuah sistem berbasis web yang fungsional. Sistem ini mengintegrasikan *backend* yang dibuat dengan Python dan Flask untuk pemrosesan *machine learning* dengan *frontend* berbasis HTML dan PHP, yang terhubung melalui REST API. Pendekatan ini menjadikan model dapat diakses dan digunakan secara praktis oleh pengguna akhir.

3. Antarmuka yang ramah pengguna (Admin dan Publik)

Sistem dirancang dengan dua jenis pengguna, yaitu *Admin* dan *User Publik*. Untuk publik Antarmuka dirancang agar sederhana dan mudah digunakan oleh masyarakat umum sebagai alat bantu deteksi dini non-diagnostik. Pengguna dapat dengan mudah memasukkan data dan mendapatkan hasil estimasi risiko yang disajikan secara visual. Untuk Admin, Terdapat *dashboard* komprehensif yang memungkinkan admin mengelola seluruh aspek sistem, mulai dari

memantau riwayat prediksi, mengelola konten edukasi, hingga mengelola model *machine learning*.

4. Fitur Manajemen Model

Salah satu keunggulan utama sistem ini adalah fitur "Kelola Model" yang memungkinkan admin untuk melatih ulang, mengevaluasi, dan mengaktifkan model secara mandiri hanya dengan beberapa klik. Admin bahkan dapat memilih untuk melatih ulang model menggunakan data riwayat prediksi baru yang dikumpulkan dari pengguna, sehingga sistem memiliki kapabilitas untuk terus berkembang dan beradaptasi.

5. Fokus pada Edukasi dan Deteksi Dini

Selain sebagai alat prediksi, sistem ini juga berfungsi sebagai media promotif dan preventif kesehatan. Terdapat menu "Edukasi Kesehatan" yang menyediakan artikel-artikel informatif mengenai kesehatan jantung, pencegahan, dan gaya hidup sehat. Hal ini sejalan dengan tujuan sistem sebagai alat bantu deteksi dini yang murah, cepat, dan mudah diakses.

4.4.2. Kelemahan Program

Adapun kelemahan dari program yang dibuat adalah sebagai berikut :

1. Ketergantungan pada Dataset Sekunder

Model dilatih dan dievaluasi menggunakan *dataset* sekunder yang bersumber dari *platform* Kaggle. Kinerja model yang tinggi pada *dataset* ini belum tentu dapat digeneralisasi secara langsung pada data populasi pasien di dunia nyata, misalnya di Indonesia, yang mungkin memiliki karakteristik demografis dan klinis yang berbeda.

2. Evaluasi Model yang terbatas

Evaluasi model hanya berfokus pada metrik teknis seperti akurasi, presisi, *recall*, dan F1-score. Aspek-aspek penting lainnya dalam konteks medis, seperti keadilan (*fairness*), interpretabilitas (kemampuan model untuk menjelaskan hasil prediksinya), dan keamanan data tidak dibahas secara mendalam.

3. Bersifat Non-Diagnostik

Secara tegas dinyatakan bahwa sistem ini adalah alat bantu deteksi dini dan bersifat non-diagnostik. Ini berarti sistem tidak dapat dan tidak dimaksudkan untuk menggantikan peran serta keputusan tenaga medis profesional. Meskipun merupakan pendekatan yang etis, hal ini juga menjadi batasan fungsional dari program.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi, pengujian, dan analisis yang telah dilakukan dalam penelitian mengenai implementasi *Gradient Boosting Machine* (GBM) untuk deteksi penyakit jantung berbasis web, dapat ditarik beberapa kesimpulan yang menjawab rumusan masalah sebagai berikut :

1. Implementasi algoritma *Gradient Boosting Machine* (GBM) berhasil dilakukan untuk melakukan klasifikasi risiko penyakit jantung menggunakan data sekunder dari *platform* Kaggle. Proses implementasi ini melibatkan beberapa tahapan, mulai dari pra-pemrosesan data seperti *label encoding* dan standardisasi , hingga pelatihan model menggunakan pustaka *scikit-learn* di Python. Model GBM yang dibangun mampu membedakan antara kelas "Sakit Jantung" dan "Sehat" dengan sangat baik.
2. Performa model GBM yang dievaluasi menggunakan metrik klasifikasi menunjukkan hasil yang sangat tinggi dan andal. Setelah melalui proses optimasi parameter dengan *Grid Search* , model terbaik mencapai akurasi global sebesar 98.54%. Secara spesifik, model ini memiliki nilai Recall 1.0000 untuk kelas "Sakit Jantung" , yang berarti model berhasil mengidentifikasi semua kasus positif tanpa ada yang terlewat (*false negative*), sebuah capaian krusial untuk aplikasi di bidang medis. Nilai *Precision* untuk kelas "Sakit Jantung" adalah 0.9709 dan untuk kelas "Sehat" adalah 1.0000, yang menunjukkan tingkat keakuratan prediksi yang tinggi.

3. Sebuah sistem prediksi penyakit jantung berbasis web yang terintegrasi dengan model GBM telah berhasil dirancang dan dibangun. Sistem ini memiliki dua hak akses, yaitu User Publik dan Admin. *User Publik* dapat melakukan prediksi secara mandiri, melihat hasilnya, mengunduh, serta mengakses konten edukasi kesehatan tanpa perlu login. Sementara itu, Admin memiliki *dashboard* komprehensif untuk memantau aktivitas sistem, mengelola riwayat prediksi, mengelola konten edukasi, serta fitur inovatif untuk melatih ulang dan mengaktifkan model *machine learning* secara mandiri. Berdasarkan pengujian *black-box*, seluruh fungsionalitas sistem berjalan dengan baik dan sesuai dengan hasil yang diharapkan.

5.2 Saran

Meskipun penelitian ini telah mencapai tujuannya, terdapat beberapa aspek yang dapat dikembangkan lebih lanjut untuk meningkatkan kualitas dan jangkauan sistem. Berikut adalah saran untuk penelitian di masa mendatang:

1. Perbandingan dengan Algoritma Lain: Penelitian ini berfokus pada algoritma GBM. Untuk pengembangan selanjutnya, disarankan untuk melakukan studi komparatif dengan algoritma *machine learning* lain seperti *XGBoost*, *LightGBM*, *Random Forest*, atau bahkan metode *deep learning* untuk mengetahui model mana yang memberikan performa paling optimal pada *dataset* penyakit jantung.
2. Integrasi dengan Sistem Informasi Kesehatan: Sistem ini masih bersifat prototipe dan mandiri. Untuk penerapan yang lebih luas, disarankan agar sistem ini diintegrasikan dengan Sistem Informasi Rumah Sakit (SIRS) atau

Rekam Medis Elektronik (RME). Integrasi ini akan mempermudah alur kerja tenaga medis dan memungkinkan pemanfaatan data pasien secara *real-time*.

3. Peningkatan Keamanan dan Skalabilitas: Seiring dengan potensi pengumpulan data riwayat prediksi pengguna untuk melatih ulang model, aspek keamanan data perlu menjadi prioritas. Disarankan untuk mengimplementasikan enkripsi data dan protokol keamanan yang lebih kuat. Selain itu, arsitektur sistem dapat dioptimalkan untuk skalabilitas yang lebih baik guna menangani jumlah pengguna yang lebih besar.

DAFTAR PUSTAKA

- Binus Student Corner. (2022). *Teknik pre-processing dan classification dalam data science – Master of Industrial Engineering*. Binus University. <https://mie.binus.ac.id/2022/08/26/teknik-pre-processing-dan-classification-dalam-data-science/>
- Bisa AI Academy. (2023). *Bisa AI Academy - Course: Machine Learning Dengan Scikit Learn Python*. <https://bisa.ai/course/detail/MzU3/1>
- Budianto, A. J., Ocsa, P., & Saian, N. (2023). Pengembangan Modul Inventory Management pada Aplikasi Master Distribution Centre System Menggunakan Framework Flask di PT XYZ. *Jurnal Teknologi Informasi Dan Komunikasi*, 7(2), 2023. <https://doi.org/10.35870/jti>
- Devia In Baliani, M., Rudolf Huizen, R., & Angga Pradipta, G. (n.d.). *Perbandingan Performa Data Penyakit Jantung Menggunakan Pendekatan Klasifikasi Boosting Methods*.
- Dinas Kesehatan Kalbar. (2024). *Hari Jantung Sedunia 2024: Ayo Bergerak untuk Sehatkan Jantungmu | Dinas Kesehatan Provinsi Kalimantan Barat*. https://dinkes.kalbarprov.go.id/artikel/hari-jantung-sedunia-2024-ayo-bergerak-untuk-sehatkan-jantungmu/?utm_source=chatgpt.com
- H.M., A. (2019). *Penerapan Metode C45 Dalam Memprediksi Pola Pembelian Bahan Campuran Olahan Karet (Studi Kasus : PT.Anugrah Sibolga Lestari)*. <http://repository.potensi-utama.ac.id/jspui/jspui/handle/123456789/3504>
- IBM. (n.d.). *Apa itu Machine Learning (ML)? | IBM*. Retrieved April 2, 2025, from https://www.ibm.com/id-id/topics/machine-learning?utm_source=chatgpt.com
- Kemenkes. (2022). *Mengenal Faktor Risiko Penyakit Jantung Koroner*. https://ayosehat.kemkes.go.id/mengenal-faktor-risiko--penyakit-jantung-koroner?utm_source=chatgpt.com
- Kemenkes. (2024). *Cara Menjaga Kesehatan Jantung dengan Aktivitas Fisik*. https://ayosehat.kemkes.go.id/cara-menjaga-kesehatan-jantung?utm_source=chatgpt.com
- Lina, N., & Saraswati, D. (2020). Deteksi Dini Penyakit Jantung Koroner di Desa Kalimanggis dan Madiasari Kabupaten Tasikmalaya Article Info. *Jurnal Warta LPM*, 23(1), 45–53. <http://journals.ums.ac.id/index.php/warta>
- Mulyo, H., & Khanif Zyen, A. (2025). Pengaruh Hyperparameter Tuning Gradient Boosting Terhadap Prediksi Pemilihan Program Studi Mahasiswa Baru. *Media Online*, 5(2), 131–137. <https://doi.org/10.47065/bulletincsr.v5i2.454>
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. In *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 5, Issue 2).

- Pratama, A. (2020). *HTML Uncover - Panduan Belajar HTML Untuk Pemula*. www.duniailkom.com
- Rashad, I., Isnanto, R. R., & Widodo, C. E. (2023). Klasifikasi Penyakit Jantung Menggunakan Algoritma Analisis Diskriminan Linier. *J. Sistem Info. Bisnis*, 13(1), 29–36. <https://doi.org/10.21456/vol13iss1pp29-36>
- Rumbaugh, James., Jacobson, Ivar., & Booch, Grady. (2021). *The unified modeling language reference manual*. Addison-Wesley.
- Setiawan, R. (2021). *Memahami Class Diagram Lebih Baik - Dicoding Blog*. <https://www.dicoding.com/blog/memahami-class-diagram-lebih-baik/>
- Setiawan, S. (2020). *Membicarakan Precision, Recall, dan F1-Score*. Medium.Com. <https://stevkarta.medium.com/membicarakan-precision-recall-dan-f1-score-e96d81910354>
- Yıldız, A. Y., & Kalayci, A. (2024). *Gradient Boosting Decision Trees on Medical Diagnosis over Tabular Data*. <http://arxiv.org/abs/2410.03705>