

# Arsitektur Mikrolayanan pada Platform Penilaian Risiko Kardiovaskular Menggunakan Vue.js dan Integrasi Vertex AI

Muhammad Anas Kusuma Amir<sup>1</sup>, Prio Kustanto<sup>1,\*</sup>

\* Korespondensi: e-mail: [pkustanto@dsn.ubharajaya.ac.id](mailto:pkustanto@dsn.ubharajaya.ac.id)

<sup>1</sup> Informatika; Fakultas Ilmu Komputer, Universitas Bhayangkara Jakarta Raya; Jalan Raya Perjuangan Bekasi Utara 17142, telp.021-88955882; e-mail: [manaska129@gmail.com](mailto:manaska129@gmail.com), [pkustanto@dsn.ubharajaya.ac.id](mailto:pkustanto@dsn.ubharajaya.ac.id)

Submitted : 14 April 2026  
Revised : 27 April 2026  
Accepted : 12 Mei 2026  
Published : 30 Mei 2026

## Abstract

Heart disease is a leading cause of death in Indonesia, exacerbated by delayed diagnosis and lack of access to healthcare facilities. This study aimed to develop JantungIn, a cardiovascular risk assessment web platform using a microservices architecture. The system separated the user interface layer using Vue.js, server logic with Hapi.js, and a Supabase relational database. The novelty of this research lay in the real-time integration of a TensorFlow.js prediction model and a Vertex AI Dialogflow CX intelligent virtual assistant. System evaluation using the Black Box method covered positive and negative scenarios on core functionalities, including role authentication for doctors and patients, medical parameter validation, and medical history storage. The testing results showed that the software architecture successfully processed data exchange without functional errors. In conclusion, JantungIn was successfully implemented as a reliable medical decision support system to facilitate early detection of heart disease in the community.

**Keywords:** Black Box Testing, Heart Disease, Microservices Architecture, Vertex AI, Vue.js

## Abstrak

Penyakit jantung merupakan penyebab utama kematian di Indonesia yang diperparah oleh keterlambatan diagnosis dan minimnya akses fasilitas kesehatan. Penelitian ini bertujuan mengembangkan JantungIn, platform web penilaian risiko kardiovaskular berarsitektur mikrolayanan. Sistem ini memisahkan lapisan antarmuka menggunakan Vue.js, logika server dengan Hapi.js, dan basis data relasional Supabase. Kebaruan penelitian terletak pada integrasi model prediksi TensorFlow.js dan asisten virtual cerdas Vertex AI Dialogflow CX secara real-time. Pengujian sistem menggunakan metode *black box* mencakup skenario positif dan negatif pada fungsionalitas utama, meliputi autentikasi peran ganda (dokter dan pasien), validasi parameter medis, hingga penyimpanan riwayat rekam medis. Hasil pengujian menunjukkan bahwa arsitektur perangkat lunak mampu memproses pertukaran data tanpa adanya galat fungsional. Kesimpulannya, JantungIn berhasil diimplementasikan sebagai sistem pendukung keputusan medis yang andal untuk memfasilitasi deteksi dini penyakit jantung di masyarakat luas.

**Kata kunci:** Arsitektur Mikrolayanan, Pengujian Black Box, Penyakit Jantung, Vertex AI, Vue.js

## **1. Pendahuluan**

Penyakit jantung merupakan ancaman kesehatan global yang signifikan, termasuk di Indonesia. Berdasarkan data Riset Kesehatan Dasar (Riskesdas), prevalensi penyakit tidak menular seperti penyakit jantung terus menunjukkan tren peningkatan yang menuntut adanya solusi deteksi dini berbasis teknologi (Kemenkes RI, 2018). Masalah utama yang dihadapi adalah ketidakmerataan fasilitas kesehatan, di mana akses layanan di daerah pedesaan jauh lebih terbatas dibandingkan daerah perkotaan. Selain itu, keterlambatan diagnosis menjadi faktor krusial yang memperburuk kondisi pasien, sehingga diperlukan sistem pendukung keputusan medis yang dapat diakses secara mandiri oleh masyarakat luas (Murni et al., 2021).

Dalam konteks rekayasa perangkat lunak, solusi digital untuk kesehatan seringkali terkendala pada skalabilitas dan interaktivitas. Tantangan utama adalah bagaimana membangun platform yang tidak hanya akurat dalam pengolahan data medis, tetapi juga responsif dan memiliki interaksi yang intuitif bagi pengguna awam. Sebagian besar aplikasi kesehatan yang ada saat ini masih bersifat statis dan kurang memberikan panduan interaktif, sehingga menyulitkan pengguna non-medis dalam memahami hasil penilaian risiko kesehatan mereka (Axza et al., 2023). Padahal, penerapan algoritma kecerdasan buatan telah terbukti mampu mengoptimalkan efektivitas dan akurasi sistem pengambilan keputusan di berbagai sektor (Noe'man et al., 2025).

Penelitian mengusulkan pengembangan platform JantungIn dengan fokus pada arsitektur perangkat lunak berbasis mikrolayanan yang memisahkan antara lapisan antarmuka (*frontend*), logika bisnis (*backend*), dan layanan kecerdasan buatan. Penggunaan *framework Vue.js* pada sisi *frontend* dipilih untuk memastikan performa yang cepat melalui mekanisme *virtual DOM*, sementara penggunaan *Hapi.js* pada sisi *backend* bertujuan untuk menjamin keamanan pertukaran data melalui *Representational State Transfer Application Programming Interface (REST API)* (Sutara & Gunawan, 2024).

Nilai kebaruan (*novelty*) dari penelitian ini terletak pada strategi integrasi model prediksi medis ke dalam ekosistem aplikasi web melalui asisten virtual interaktif *Dialogflow CX* di layanan *Vertex AI*. Berbeda dengan penelitian sebelumnya yang mayoritas hanya berfokus pada optimasi akurasi model di lingkungan terbatas, penelitian ini menekankan pada implementasi perangkat lunak yang siap pakai (*production-ready*) dengan arsitektur yang modular (Haryanto & Saefurrahman, 2024). Dengan pendekatan ini, sistem mampu melakukan pengolahan data secara *real-time*, sehingga memberikan aksesibilitas tinggi bagi masyarakat untuk memantau kesehatan jantung secara mandiri dengan memanfaatkan layanan komputasi awan dan analitik API yang skalabel (Kotadiya et al., 2024; Salehi et al., 2026).

## **2. Metode Penelitian**

Penelitian ini menggunakan pendekatan *Rapid Application Development (RAD)* untuk membangun platform JantungIn, sebuah aplikasi penilaian risiko kardiovaskular berbasis *web*. Metode RAD dipilih karena memungkinkan iterasi desain dan pengembangan fungsionalitas

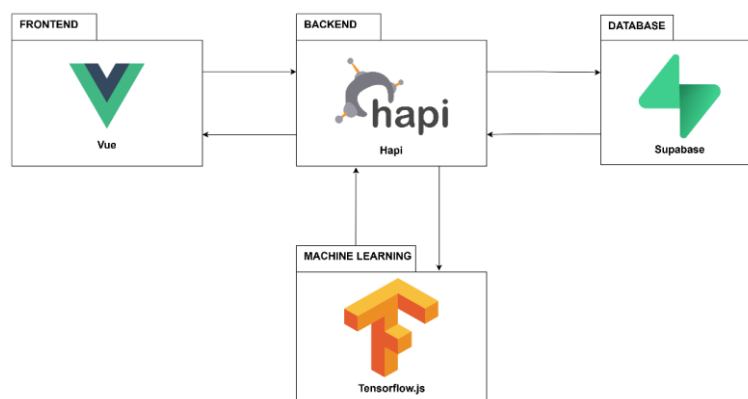
yang cepat. Tahapan pengembangan perangkat lunak dalam penelitian diadaptasi dari metode RAD.

### 2.1. Tahap Komunikasi (*Communication*)

Tahap awal difokuskan pada analisis kebutuhan sistem dan identifikasi peran pengguna. Sistem dirancang untuk memfasilitasi dua peran utama, yaitu Pasien dan Dokter, demi menjaga keamanan dan validitas data medis. Dokter membutuhkan platform yang dapat memproses data klinis secara cepat sebagai alat bantu *skrining*, sementara pasien membutuhkan akses mudah untuk melihat riwayat deteksi dan mendapatkan edukasi kesehatan.

### 2.2. Tahap Perencanaan (*Planning*)

Berdasarkan analisis kebutuhan, arsitektur perangkat lunak direncanakan menggunakan pendekatan mikrolayanan (*microservices*). Pada sisi *front-end*, teknologi yang dipilih adalah *Vue.js* yang dipadukan dengan *Tailwind CSS* untuk membangun antarmuka responsif yang ideal bagi arsitektur mikrolayanan modern (Qiu, 2025; Santos et al., 2024). Pada sisi *back-end*, sistem dibangun menggunakan *runtime* Node.js dengan *framework* Hapi.js untuk mengelola *routing* dan *Representational State Transfer Application Programming Interface (REST API)*. Untuk model prediksinya, diimplementasikan algoritma *Multi-Layer Perceptron (MLP)* menggunakan pustaka *TensorFlow* yang dikonversi ke dalam format *TensorFlow.js* agar dapat dieksekusi secara efisien di sisi klien atau *server*. Arsitektur mikrolayanan yang menghubungkan keseluruhan komponen perangkat lunak ini dapat dilihat pada Gambar 1.



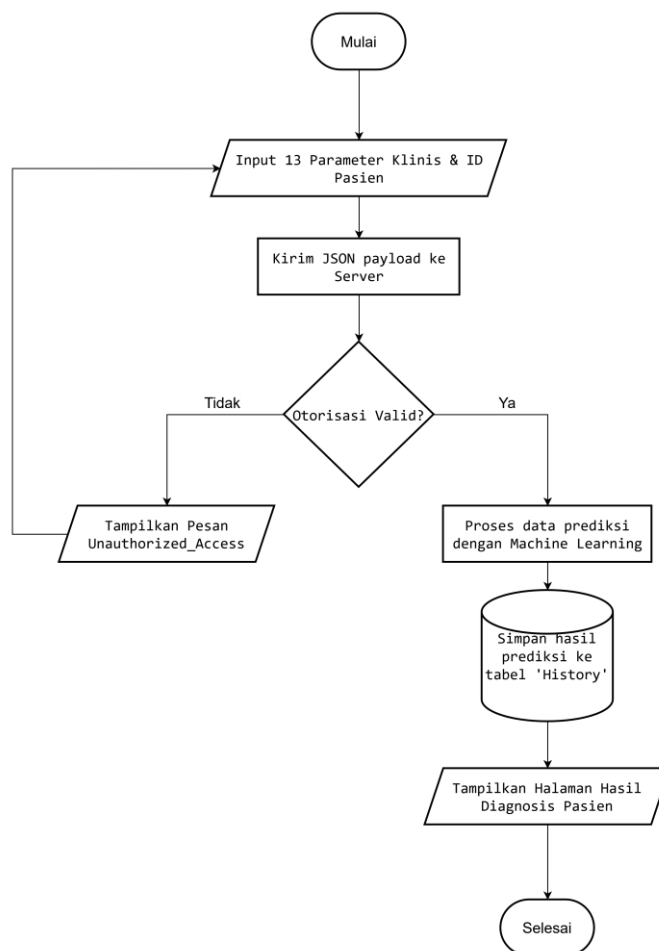
Sumber: Hasil Penelitian (2026)

Gambar 1. Arsitektur Sistem JantungIn

### 2.3. Tahap Pemodelan (*Modeling*)

Tahap pemodelan mencakup perancangan antarmuka pengguna (UI/UX) dan pemodelan data. Perancangan visual awal dibuat menggunakan Figma, yang mencakup halaman autentikasi, dasbor dokter untuk *input* parameter medis (usia, tekanan darah, kolesterol, dll.), dan halaman riwayat pasien. Pendekatan purwarupa (*prototype*) pada perancangan media interaktif ini sangat krusial untuk memastikan kesesuaian antarmuka sistem dengan kebutuhan pengguna sebelum tahap konstruksi kode dilakukan (Kustanto et al., 2024).

Alur data dirancang sedemikian rupa agar *input* dari antarmuka dikirim melalui *REST API* ke *back-end*, diproses oleh model *Machine Learning*, dan hasilnya disimpan secara terstruktur di *database* Supabase sebelum ditampilkan kembali kepada pengguna. Seluruh alur interaksi dan logika pemrosesan data dari tahap *input* hingga menghasilkan diagnosis tersebut diilustrasikan secara lebih rinci melalui bagan alir (*flowchart*) pada Gambar 2.



Sumber: Hasil Penelitian (2026)

Gambar 2. *Flowchart* Alur Prediksi Sistem Jantungln

#### 2.4. Tahap Konstruksi (*Construction*)

Pada fase ini, rancangan diterjemahkan ke dalam kode pemrograman. *Front-end* dikembangkan menggunakan *Vue.js* dan *JavaScript*, sementara *back-end* diimplementasikan dengan *Hapi.js* untuk menangani logika bisnis. Model *MLP* dilatih menggunakan *dataset* penyakit jantung dari *UCI Machine Learning Repository*, dan telah melalui proses prapemrosesan data yang meliputi penanganan nilai hilang (*missing value*), data duplikat, serta *outlier*. Model yang telah dievaluasi ini kemudian diintegrasikan ke dalam sistem perangkat lunak untuk proses inferensi. Seluruh titik akhir (*endpoint*) API diuji menggunakan perangkat lunak *Postman* untuk memastikan pertukaran data berjalan sesuai spesifikasi. Alur logika pemrograman dalam mengeksekusi fungsi prediksi dan otorisasi penyimpanan riwayat diimplementasikan seperti yang ditunjukkan pada Gambar 3.

```
INPUT: Medical_Parameters, User_ID
OUTPUT: Prediction_Result, History_Saved

BEGIN
1. USER melakukan input parameter klinis pada antarmuka Vue.js
2. SISTEM mengirimkan JSON payload ke backend Hapi.js
3. IF Request_Authorized (user_role Valid) THEN
  a. BACKEND memproses data prediksi menggunakan TensorFlow.js (Model MLP)
  b. SISTEM menerima hasil prediksi risiko (0: Sehat, 1: Berisiko)
  c. SISTEM menyimpan data dan hasil ke tabel 'History' di database Supabase
  d. RETURN Response_Data ke Dashboard User
ELSE
  e. RETURN Unauthorized_Access
ENDIF
END
```

Sumber: Hasil Penelitian (2026)

Gambar 3. Pseudocode Alur Logika Sistem Jantungln

## 2.5. Tahap Penyerahan (*Deployment*)

Aplikasi yang telah dibangun kemudian di-*deploy* ke *server* komputasi awan. Sisi *front-end* di-hosting menggunakan layanan Netlify untuk memastikan aksesibilitas publik. Selama fase ini, dilakukan *user testing* internal yang menghasilkan sejumlah perbaikan pada komponen UI agar sistem menjadi lebih intuitif, terutama bagi pengguna non-teknis.

## 3. Hasil dan Pembahasan

Pada bagian ini dijelaskan mengenai implementasi antarmuka perangkat lunak Jantungln berdasarkan rancangan tahap pemodelan, serta hasil pengujian fungsionalitas dan kinerja prediksi dari sistem yang telah dibangun.

### 3.1. Implementasi Antarmuka Pengguna

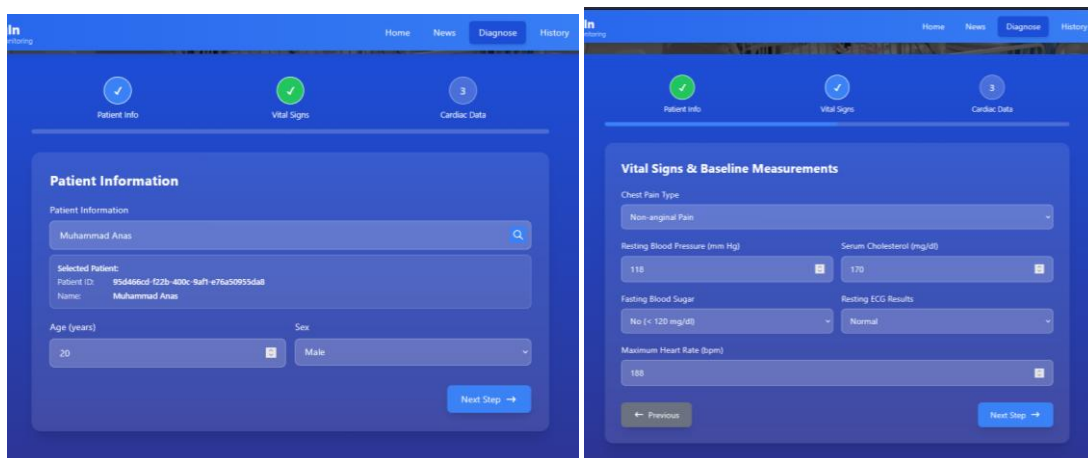
Antarmuka pengguna (UI) dikembangkan menggunakan *framework Vue.js* yang dipadukan dengan *Tailwind CSS* untuk menghasilkan tampilan yang responsif dan interaktif. Berdasarkan rancangan arsitektur, implementasi sistem difokuskan pada tiga alur utama, yaitu *input* data, pemrosesan data, dan *output* diagnosis:

#### a. Implementasi Input Data Diagnosis

Pada halaman *Doctor Dashboard*, sistem menyediakan antarmuka "Cardiovascular Diagnosis" yang interaktif. Untuk meminimalisasi kesalahan entri (kesalahan kognitif) oleh tenaga medis, formulir *input* 13 parameter klinis ini dipecah ke dalam tiga tahapan langkah (*wizard*), yang masing-masing penjelasan dan fungsinya dapat dilihat secara terperinci pada Gambar Tampilan Formulir Input Diagnosis Pasien.

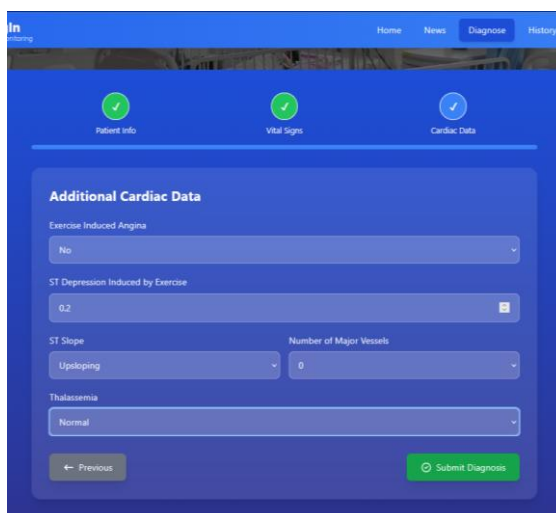
Gambar 4(a) Bagian *Patient Info*: Berfungsi untuk melakukan pencarian dan pendataan demografi pasien, mencakup parameter Nomor Rekam Medis (*Patient ID*), Nama, Usia, dan Jenis Kelamin. Gambar 4(b) Bagian *Vital Signs & Baseline Measurements*: Berfungsi untuk memasukkan parameter tanda-tanda vital dasar, yang meliputi Tipe Nyeri Dada, Tekanan Darah Istirahat, Kolesterol Serum, Gula Darah Puasa, Hasil EKG Istirahat, serta Detak

Jantung Maksimum. Gambar 4(c) Bagian *Additional Cardiac Data*: Digunakan untuk mencatat hasil uji klinis spesifik, yang terdiri atas parameter Angina Akibat Olahraga, Depresi ST, Kemiringan ST, Jumlah Pembuluh Darah Utama, dan status Talasemia.



(a)

(b)



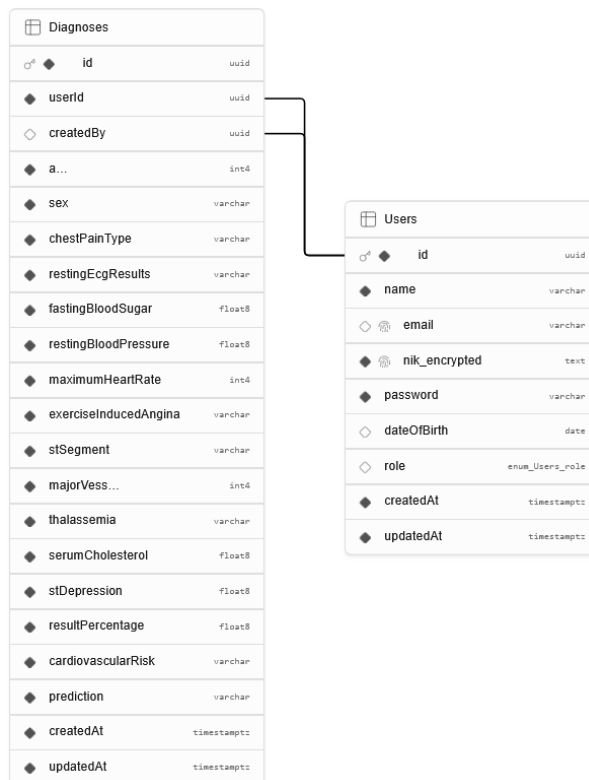
(c)

Sumber: Hasil Penelitian (2026)

Gambar 4. Tampilan Formulir Input Diagnosis Pasien (a) Bagian *Patient Info*, (b) Bagian *Vital Signs & Baseline Measurements*, (c) Bagian *Additional Cardiac Data*

#### b. Proses dan Relasi Data

Data medis yang telah dimasukkan kemudian dienkapsulasi ke dalam format JSON dan dikirimkan melalui *REST API* ke *backend* Hapi.js. Seluruh data otentikasi pengguna (Dokter dan Pasien) serta riwayat rekam medis disimpan secara terpusat di *database* Supabase. Skema basis data relasional dirancang untuk menghubungkan entitas pengguna dengan entitas riwayat rekam medis, memastikan setiap hasil diagnosis terhubung secara aman dengan ID Pasien yang tepat tanpa adanya redundansi data.



Sumber: Hasil Penelitian (2026)

Gambar 4. Relasi Antar Tabel pada Database Supabase

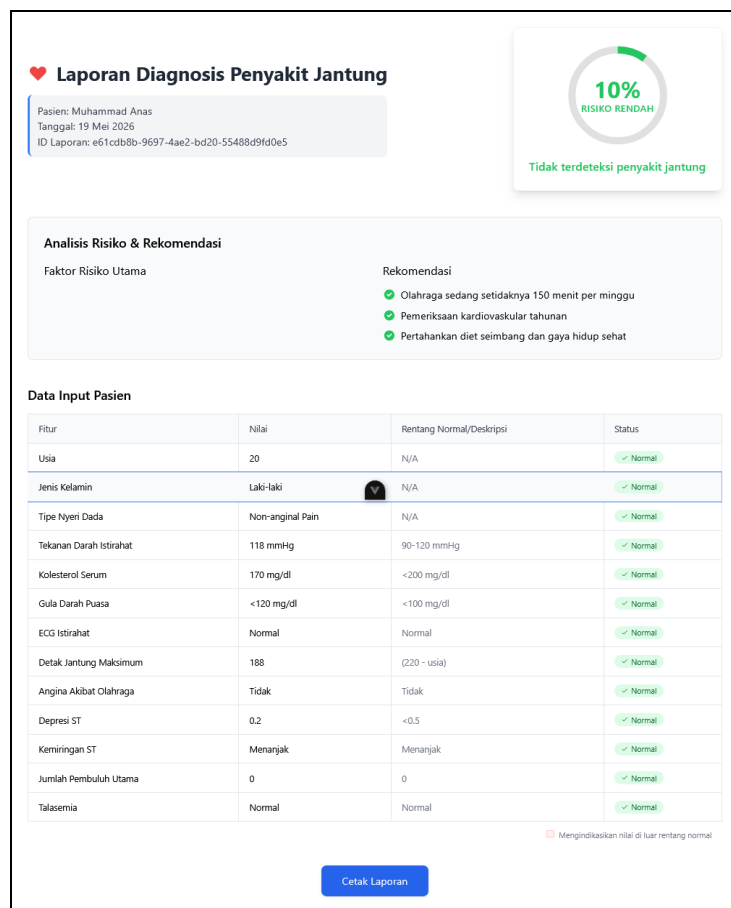
Berdasarkan Gambar 5, terdapat dua tabel utama dalam *database*, yaitu tabel *Users* dan *Diagnoses*. Tabel *Users* digunakan untuk menyimpan data pengguna, seperti identitas, email, kata sandi terenkripsi, serta peran pengguna dalam sistem. Sementara itu, tabel *Diagnoses* menyimpan data hasil pemeriksaan dan diagnosis penyakit jantung, termasuk parameter medis yang digunakan dalam proses prediksi, seperti jenis kelamin, tipe nyeri dada, tekanan darah, kadar kolesterol, hasil elektrokardiogram, serta hasil prediksi risiko kardiovaskular. Relasi antara kedua tabel dihubungkan melalui atribut *userId*, sehingga setiap riwayat diagnosis dapat dikaitkan dengan pengguna yang bersangkutan. Struktur relasi ini mendukung pengelolaan data yang terorganisir, menjaga integritas data, serta memudahkan proses penyimpanan dan penelusuran riwayat diagnosis pasien.

c. Implementasi Output Hasil Diagnosis

Hasil komputasi dari model *Machine Learning* ditampilkan pada antarmuka "Diagnosis Details" atau laporan rekam kesehatan (*Health Records*). Halaman ini menyajikan visualisasi tingkat risiko kardiovaskular dalam bentuk persentase dan indikator status risiko (*Low Risk* atau *High Risk*). Selain itu, *output* ini dilengkapi dengan interpretasi dokter dan rekomendasi tindakan pencegahan konkret terkait pola makan, olahraga, dan manajemen stres.

Gambar 6, halaman laporan hasil diagnosis menampilkan informasi pasien, hasil prediksi sistem, tingkat risiko penyakit jantung dalam bentuk persentase, serta kategori risiko yang dihasilkan oleh model *Machine Learning*. Informasi tersebut disajikan secara terstruktur sehingga memudahkan pengguna dalam memahami hasil diagnosis yang diberikan. Selain

menampilkan hasil prediksi, sistem juga menyediakan interpretasi medis dan rekomendasi tindakan pencegahan yang dapat dilakukan pengguna untuk mengurangi faktor risiko penyakit jantung. Tampilan laporan dirancang agar mudah dibaca dan dapat digunakan sebagai referensi awal dalam proses pemantauan kondisi kesehatan pasien.



Sumber: Hasil Penelitian (2026)

Gambar 5. Tampilan *Output* Laporan Hasil Diagnosis

### 3.2. Hasil Pengujian

Untuk memvalidasi bahwa perangkat lunak yang dibangun telah memenuhi spesifikasi dan analisis kebutuhan, dilakukan pengujian fungsionalitas menggunakan metode *Black Box Testing*. Pengujian ini difokuskan pada interaksi pengguna dengan antarmuka (*frontend*), keamanan akses fungsionalitas, validasi input di sisi peladen (*backend*), dan respons dari integrasi antarmuka pemrograman aplikasi (API). Skenario pengujian dirancang secara komprehensif mencakup skenario positif dan skenario negatif. Hasil pengujian dibagi berdasarkan fungsionalitas utama sistem, yaitu registrasi, autentikasi, akses diagnosis, serta pengelolaan riwayat.

#### a. Pengujian Fitur Registrasi

Pengujian pada modul registrasi bertujuan untuk memastikan sistem mampu memvalidasi data pengguna baru sebelum disimpan ke dalam basis data.

Tabel 1. Hasil Pengujian Fitur Registrasi

No	Fitur	Test Case	Ekspetasi	Hasil
1	Register	name, nik 16 digit, password >= 6	Registrasi sukses, akun dibuat dan token login dikembalikan.	Berhasil
2	Register	email tidak dikirim	Registrasi tetap sukses karena email bersifat opsional.	Berhasil
3	Register	nik 15 digit	Ditolak karena NIK wajib tepat 16 digit angka (validasi).	Berhasil
4	Register	nik ada huruf	Ditolak karena NIK harus hanya angka (validasi).	Berhasil
5	Register	password 5 char	Ditolak karena password minimal 6 karakter.	Berhasil
6	Register	nik existing	Ditolak karena NIK sudah terdaftar (konflik data).	Berhasil
7	Register	email existing	Ditolak karena email sudah terdaftar (konflik data).	Berhasil

Sumber: Hasil Penelitian (2026)

Berdasarkan Tabel 1, pengujian fitur registrasi menunjukkan bahwa sistem berhasil memvalidasi parameter wajib (seperti ketepatan 16 digit NIK dan minimal 6 karakter kata sandi). Sistem juga terbukti mampu menangani skenario negatif dengan menolak duplikasi data (NIK atau *email* yang sudah terdaftar), sehingga integritas data pengguna tetap terjaga secara aman di dalam *database*.

b. Pengujian Fitur Autentikasi (*Login*)

Pengujian ini dilakukan untuk memvalidasi proses masuk pengguna berdasarkan kredensial dan hak akses (pasien atau dokter).

Tabel 2. Hasil Pengujian Fitur Autentikasi (*Login*)

No	Fitur	Test Case	Ekspetasi	Hasil
1	Login	NIK valid + password benar	Login sukses, token autentikasi dikembalikan.	Berhasil
2	Login	NIK valid + password salah	Ditolak karena kredensial tidak valid.	Berhasil
3	Login	NIK belum terdaftar	Ditolak karena pengguna tidak ditemukan / kredensial tidak valid.	Berhasil
4	Login	NIK non-digit / bukan 16	Ditolak karena format NIK tidak sesuai (validasi).	Berhasil
5	Login	password kosong	Ditolak karena password wajib diisi (validasi).	Berhasil
6	Admin Login	email admin/dokter + password benar	Login admin/dokter sukses, token role admin/dokter dikembalikan.	Berhasil
7	Admin Login	email role user	Ditolak karena akun bukan admin/dokter.	Berhasil

Sumber: Hasil Penelitian (2026)

Pada Tabel 2, sistem terbukti mampu memverifikasi kredensial pengguna secara akurat. Sistem dengan cerdas membedakan sesi masuk antara pengguna biasa (pasien) dan tenaga medis (dokter). Percobaan akses yang tidak memenuhi syarat secara otomatis ditolak dengan memunculkan pesan galat kredensial tidak valid.

c. Pengujian Fitur Diagnosis dan Pembatasan Akses Antarmuka

Pengujian ini memastikan bahwa hanya peran berwenang yang dapat mengakses formulir diagnosis dan memproses model prediksi.

Tabel 3. Hasil Pengujian Fitur Diagnosis dan Pembatasan Akses

No	Fitur	Test Case	Ekspetasi	Hasil
1	Diagnosis	token admin/dokter + payload lengkap	Diagnosis sukses, hasil prediksi & persentase risiko dikembalikan.	Berhasil
2	Diagnosis	token admin/dokter + patientId valid	Diagnosis sukses untuk pasien tertentu (userId menjadi patientId).	Berhasil
3	Diagnosis	no Authorization	Ditolak karena tidak ada token autentikasi.	Berhasil
4	Diagnosis	token role user	Ditolak karena role user tidak punya izin melakukan diagnosis.	Berhasil
5	Diagnosis	sex bukan Male/Female	Ditolak karena nilai enum tidak sesuai validasi.	Berhasil

Sumber: Hasil Penelitian (2026)

Berdasarkan Tabel 3, pengujian pada fitur diagnosis menunjukkan bahwa pembatasan akses antarmuka (*authorization*) berfungsi dengan sangat baik. Sistem menolak permintaan diagnosis dari pengguna yang tidak memiliki otorisasi (pasien atau publik) demi menjaga kerahasiaan dan standar prosedur medis. Data diagnosis juga tervalidasi dengan ketat sebelum dikirim ke model *Machine Learning*.

d. Pengujian Fitur Pengelolaan Riwayat Diagnosis

Pengujian ini memvalidasi kemampuan sistem dalam melacak dan menampilkan riwayat rekam medis secara aman.

Tabel 4. Hasil Pengujian Fitur Riwayat dan Hasil Diagnosis

No	Fitur	Test Case	Ekspetasi	Hasil
1	Hasil Diagnosis	id diagnosis existing	Data diagnosis ditemukan dan dikembalikan.	Berhasil
2	Hasil Diagnosis	id invalid	Ditolak karena format id harus UUID (validasi).	Berhasil
3	Hasil Diagnosis	id UUID tapi tidak ada	Ditolak karena diagnosis tidak ditemukan.	Berhasil
4	Hasil Diagnosis	token user + id bukan miliknya	Ditolak/hasil tidak ditemukan karena akses dibatasi hanya diagnosis milik sendiri.	Berhasil
5	Riwayat Diagnosis	token user	Riwayat diagnosis user sendiri dikembalikan (bisa kosong).	Berhasil

No	Fitur	Test Case	Ekspetasi	Hasil
6	Riwayat	token admin/dokter +	Riwayat diagnosis pasien tertentu	Berhasil
	Diagnosis	patientId valid	dikembalikan.	
7	Riwayat	patientId bukan UUID	Ditolak karena format patientId	Berhasil
	Diagnosis		harus UUID.	
8	Riwayat	no Authorization	Ditolak karena tidak ada token	Berhasil
	Diagnosis		autentikasi.	

Sumber: Hasil Penelitian (2026)

Hasil pengujian pada Tabel 4 membuktikan bahwa privasi rekam medis pasien terjaga dengan aman. Sistem berhasil memblokir upaya akses paksa jika pasien mencoba melihat riwayat medis pasien lain, sekaligus memastikan bahwa format pencarian data (*UUID*) divalidasi secara ketat oleh *backend Hapi.js*.

#### 4. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, penelitian ini berhasil mengembangkan platform web JantungIn sebagai sistem perangkat lunak yang andal untuk deteksi dini risiko kardiovaskular. Arsitektur mikrolayanan yang memisahkan lapisan antarmuka (*frontend*) berbasis *Vue.js*, logika peladen (*backend*) menggunakan *Hapi.js*, dan basis data *Supabase* terbukti efektif dalam memfasilitasi integrasi model prediksi *TensorFlow.js* dan asisten virtual edukasi *Vertex AI* secara *real-time*. Hasil pengujian fungsionalitas menggunakan metode *Black Box* menunjukkan bahwa seluruh skenario interaksi pengguna dan hak akses peran ganda (Dokter dan Pasien) mampu merespons instruksi dengan tepat dan aman tanpa adanya galat fungsional. Sebagai prospek pengembangan pada penelitian selanjutnya, sistem ini dapat dioptimalkan lebih lanjut melalui peningkatan enkripsi keamanan data medis pada lapisan *Application Programming Interface (API)*, serta pengintegrasian platform ke dalam Sistem Informasi Manajemen Rumah Sakit (*SIMRS*) agar dapat memperluas cakupan pemantauan rekam medis secara nasional.

#### Ucapan Terima Kasih

Peneliti mengucapkan terima kasih kepada Program Studi Independen Bersertifikat DBS Foundation Coding Camp dan Dicoding Indonesia atas dukungan fasilitas serta sumber daya pembelajaran selama proses pengembangan perangkat lunak ini. Apresiasi juga ditujukan kepada seluruh rekan tim CC25-CF129 tahun 2025 atas dedikasi dan kolaborasinya hingga platform JantungIn berhasil diselesaikan.

#### Daftar Pustaka

- Axza, F., Sofi'ie, F., & Qoiriah, A. (2023). Analisis Perbandingan Framework Front-End Javascript React dan Vue Pada Pengembangan Website. *Journal of Informatics and Computer Science*, 05.
- Haryanto, I. D., & Saefurrahman, S. (2024). Implementasi Chatbot Kesehatan Kucing Melalui

- Dialogflow dan Telegram untuk Pemberian Informasi Penyakit dan Perawatan. *JTIM: Jurnal Teknologi Informasi Dan Multimedia*, 5(4), 365–376. <https://doi.org/10.35746/jtim.v5i4.484>
- Kemendes RI. (2018). *Laporan Riskesdas 2018 Nasional*. [https://repository.badankebijakan.kemkes.go.id/id/eprint/3514/1/Laporan Riskesdas 2018 Nasional.pdf](https://repository.badankebijakan.kemkes.go.id/id/eprint/3514/1/Laporan_Riskesdas_2018_Nasional.pdf)
- Kotadiya, U., Arora, A. S., & Yachamaneni, T. (2024). Intelligent Orchestration of Cloud-Native Applications Using Google Cloud Platform and Microservices-Based Architectures. *International Journal of AI, BigData, Computational and Management Studies*, 5. <https://doi.org/10.63282/3050-9416.ijaibdcms-v5i4p111>
- Kustanto, P., Bram Khalil, R., & Noe'man, A. (2024). Penerapan Metode Prototype dalam Perancangan Media Pembelajaran Interaktif. *Journal of Students' Research in Computer Science*, 5(1), 83–94. <https://doi.org/10.31599/6x0dfz47>
- Murni, I. K., Wirawan, M. T., Patmasari, L., Sativa, E. R., Arafuri, N., Nugroho, S., & Noormanto. (2021). Delayed diagnosis in children with congenital heart disease: a mixed-method study. *BMC Pediatrics*, 21(1). <https://doi.org/10.1186/s12887-021-02667-3>
- Noe'man, A., Hidayat, A., Yogaswara, N., Handayani, D., Kustanto, P., & Hartanti, D. (2025). Implementasi algoritma random forest dalam sistem seleksi karyawan terbaik untuk meningkatkan efektivitas keputusan di PT. XYZ. *Jurnal Manajemen Informatika Jayakarta*, 5(3), 263. <https://doi.org/10.52362/jmijayakarta.v5i3.2081>
- Qiu, M. (2025). *ExtJS-JSP-Vue.js 3 Hybrid Architecture: A Case Study in Enterprise Web Application Development*. 4, 2025. [www.h-tsp.com](http://www.h-tsp.com)
- Salehi, S. S., Saadatfar, H., Oyelere, S. S., Hussain, S., Hassannataj Joloudari, J., Taheri Ledari, M., Arslan, E., & Barzegar, B. (2026). Enhancing healthcare outcome with scalable processing and predictive analytics via cloud healthcare API. *Frontiers in Digital Health*, 7. <https://doi.org/10.3389/fgdth.2025.1687131>
- Santos, L. F. C. Dos, Silva, M. V. S., Santos, S. R. R. Dos, Rocha, F. G., & Silva, E. B. Da. (2024). Microfront-End: Systematic Mapping. *International Conference on Web Information Systems and Technologies, WEBIST - Proceedings*, 119–130. <https://doi.org/10.5220/0013015400003825>
- Sutara, B., & Gunawan, S. S. (2024). Comparative analysis of REST API performance between Express.js framework and Hapi.js using Apache JMeter. *Jurnal Riset Teknik Informatika (JURETI)*, 1(1), 19–26.