

DIGITAL COCKPIT MULTIPLATFORM MODULAR DARI HUMAN DIGITAL TWIN

Herman^{*1}, Wahid Alfaridsi Achmad Zein², Abdul Fadlil³

^{1,2,3}Universitas Ahmad Dahlan, Yogyakarta

Email: ¹hermankaha@mti.uad.ac.id, ²walfaridsi@gmail.com, ³fadlil@mti.uad.ac.id

*Penulis Korespondensi

(Naskah masuk: 13 Desember 2025, diterima untuk diterbitkan: 14 Januari 2026)

Abstrak

Teknologi *Digital Twin* (DT) telah diadaptasi menjadi *Human Digital Twin* (HDT) untuk pemantauan kesehatan secara *real-time*. Namun, sistem pemantauan kesehatan berbasis HDT saat ini terkendala masalah skalabilitas, fleksibilitas antarmuka, dan ketergantungan pada koneksi internet. Arsitektur monolitik konvensional menghambat integrasi fitur dan adaptasi platform, sementara kebutuhan akan layanan berkelanjutan seringkali tidak terpenuhi. Untuk mengatasi kesenjangan penelitian ini, studi ini mengusulkan solusi dengan merancang dan mengimplementasikan *Digital Cockpit* (DC) *Multiplatform Modular* untuk HDT kardiovaskular. Penelitian ini menerapkan arsitektur modular untuk memisahkan lapisan antarmuka depan dan antarmuka belakang. Struktur ini mempermudah integrasi layanan masa depan seperti AI dan LLM. Antarmuka depan diimplementasikan sebagai *Progressive Web Application* (PWA) untuk memastikan kompatibilitas lintas platform dan memberikan kemampuan *offline* penting menggunakan *Service Worker* untuk *caching* dan pengambilan data. Kami memvalidasi sistem melalui *Blackbox Testing* dan *Compatibility Testing* di lingkungan Laptop, iOS, dan &roid. Pengujian *Blackbox* menunjukkan keberhasilan 100% pada seluruh fungsi inti, termasuk sinkronisasi data *real-time* dan *caching offline*. Penggunaan basis kode tunggal pada PWA secara signifikan mengurangi kompleksitas pengembangan antar platform dengan tetap mempertahankan skor kompatibilitas sebesar 89,58% di berbagai perangkat. Sistem DC berbasis PWA ini memberikan solusi yang terukur dengan menggabungkan fleksibilitas web dan kinerja aplikasi native bagi sistem pemantauan kesehatan digital.

Kata kunci: kardiovaskular, digital cockpit, human digital twin, arsitektur modular, progressive web application.

MODULAR MULTIPLATFORM DIGITAL COCKPIT OF HUMAN DIGITAL TWIN

Abstract

Digital Twin (DT) technology has been adapted into *Human Digital Twin* (HDT) for real-time health monitoring. However, current HDT-based health monitoring systems are hampered by issues of scalability, interface flexibility, & dependence on internet connection. Conventional monolithic architecture hinders feature integration & platform adaptation, while the need for continuous service is often unmet. To address this research gap, this study proposes a solution by designing & implementing a *Modular Multiplatform Digital Cockpit* (DC) for cardiovascular HDT. This study applies a modular architecture to separate the front-end & back-end layers. This structure facilitates the integration of future services such as AI & LLM. The front-end is implemented as a *Progressive Web Application* (PWA) to ensure cross-platform compatibility & provides essential offline capabilities using *Service Workers* for data caching & retrieval. The system was validated through *Blackbox Testing* & *Compatibility Testing* on Laptop, iOS, & &roid environments. *Blackbox testing* demonstrated 100% success across all core functions, including real-time data synchronization & offline caching. The use of a single codebase in PWAs significantly reduces cross-platform development complexity while maintaining a compatibility score of 89.58% across devices. This PWA-based DC system provides a scalable solution for digital health monitoring systems, combining the flexibility of the web with the performance of native apps.

Keywords: cardiovascular health, digital cockpit, human digital twin, modular architecture, progressive web application.

1. PENDAHULUAN

1.1. Latar Belakang Masalah

Digital Twin (DT) adalah representasi virtual yang berjalan paralel dengan entitas fisiknya untuk memantau kondisi, memprediksi perilaku, dan mendukung pengambilan keputusan berbasis data

(Armél et al., 2025). Konsep DT awalnya dikembangkan di bidang industri sebelum diadaptasi ke konteks biologis melalui *Human Digital Twin* (HDT) yang secara dinamis memodelkan tubuh manusia, organ, atau sistem fisiologis (Iliu, 2024) (Elgammal, Albrijawi & Alhadj, 2025) (Sprint,

Schmitter-edgecombe & Cook, n.d.). HDT memungkinkan pemantauan kesehatan pribadi secara *real-time* dan integrasi data multimodal (Sun et al., 2022).

Banyak penelitian telah dilakukan di bidang HDT untuk sistem pemantauan kesehatan. Penelitian tersebut berfokus pada berbagai sektor, di antaranya adalah sektor *user interface* untuk sistem tersebut. Pada kebanyakan *user interface* yang ada pada saat ini memiliki keterbatasan dalam hal skalabilitas, fleksibilitas, dan dukungan untuk perangkat *online* maupun *offline*.

Seiring perkembangan *user interface*, sistem pemantauan kesehatan digital berbasis HDT semakin menarik perhatian penelitian, namun masih menghadapi keterbatasan yang signifikan terkait skalabilitas, fleksibilitas antarmuka pengguna, dan dukungan untuk operasi *offline* (Masison et al., 2021), sehingga diperlukan antarmuka yang dapat secara efektif menjembatani kompleksitas ini.

Digital Cockpit (DC) berfungsi sebagai *user interface* yang menghubungkan pengguna dengan sistem DT termasuk HTD, dengan menyediakan visualisasi data *real-time*, integrasi data multimodal, dan interaksi langsung dengan model digital yang mewakili entitas fisik (Pellegrino et al., 2025). Sebagai jendela operasional bagi para tenaga kesehatan dan peneliti, antarmuka ini membutuhkan fleksibilitas, responsifitas, dan kemampuan *cross-device* (Wahlstr, 2017). Namun, kebutuhan ini seringkali terhambat oleh pendekatan pengembangan tradisional, di mana arsitektur monolitik menyulitkan integrasi fitur baru dan adaptasi terhadap platform, sementara antarmuka depan berbasis *native* memerlukan pembuatan versi terpisah untuk setiap sistem operasi, sehingga meningkatkan biaya dan waktu pengembangan (Fauzan et al., 2022).

Meskipun kebutuhan akan akses data kesehatan secara *real-time* dan pemantauan pasien secara terus-menerus sangat penting (Adibi, Rajabifard & Shojaei, 2024), Masih sedikit sistem yang secara optimal mengintegrasikan kemampuan *offline-first* untuk memastikan keberlanjutan layanan dalam kondisi jaringan yang tidak stabil, hal ini menjadi sebuah masalah yang sering ditemui dalam konteks digitalisasi. Oleh karena itu, muncul kesenjangan penelitian untuk menciptakan sistem DC yang mengadopsi arsitektur antarmuka belakang *modular* dan menggunakan antarmuka depan *Progressive Web Application* (PWA) untuk memastikan kompatibilitas *cross-platform* baik dalam kondisi *online* maupun *offline*.

Penelitian ini bertujuan untuk mendesain PWA, mengimplementasikan prototipe sebagai bukti konsep, dan menguji sistem DC berbasis PWA yang terintegrasi dengan arsitektur antarmuka belakang modular untuk mendukung pemantauan kesehatan kardiovaskular secara *real-time*. Lebih spesifiknya, tujuan penelitian ini adalah untuk mengimplementasikan arsitektur sistem DC modular

untuk memisahkan lapisan antarmuka depan dan antarmuka belakang, sehingga memungkinkan skalabilitas, kemudahan pemeliharaan, dan integrasi yang lebih baik dengan layanan Artificial Intelligence (AI) dan *Large Language Model* (LLM) (Elgammal, Albrijawi & Alhajj, 2025) (Masison et al., 2021).

Penelitian ini juga bertujuan untuk menguji fungsionalitas, kompatibilitas, dan kinerja sistem yang dikembangkan pada berbagai lingkungan operasional seperti Desktop, &roid, dan iOS untuk memvalidasi keandalan dan kesiapan sistem dalam penggunaan langsung. Dengan mencapai tujuan-tujuan ini, diharapkan tantangan yang diidentifikasi dalam kesenjangan penelitian dapat diatasi. Lebih jauh, penelitian ini diharapkan dapat memberikan solusi yang mudah diakses pengguna untuk pemantauan kesehatan secara digital. Hal ini sejalan dengan program pemerintah untuk lebih memanfaatkan teknologi digital dalam meningkatkan pelayanan kesehatan pada masyarakat Indonesia secara lebih luas dan berkualitas.

1.2. Perbandingan dengan Penelitian Terdahulu

Pengembangan DC dalam ekosistem DT telah banyak dilakukan untuk mendukung pemantauan jarak jauh di berbagai sektor. Meskipun demikian, penerapan pada HDT masih menghadapi kendala skalabilitas antarmuka dan stabilitas akses data pada kondisi jaringan yang buruk. Tabel 1 merangkum posisi penelitian ini dibandingkan dengan beberapa penelitian sejenis untuk mengidentifikasi celah ilmiah yang ada.

Berdasarkan perbandingan pada Tabel 1, terlihat bahwa gap penelitian yang diisi oleh studi ini adalah ketiadaan antarmuka pemantauan kesehatan yang mampu menyeimbangkan fleksibilitas lintas platform dengan kelebihan aksesibilitas pada jaringan tidak stabil. Keunikan implementasi ini dalam konteks kesehatan kardiovaskular sangat penting karena sifat data fisiologis jantung yang dinamis, di mana deteksi anomali *real-time* seperti detak jantung dan tekanan darah bersifat kritis. Dengan mengintegrasikan teknologi PWA yang mendukung kemampuan *offline-first* (Fern&o, 2022).

1.3. Landasan Teori

DT adalah representasi virtual yang hidup dan dinamis dari suatu objek, proses, maupun sistem fisik (Hergesheimer et al., 2026). DT diperbarui dengan data *real-time* dari sensor untuk memantau kinerja, menganalisis perilaku, dan mensimulasikan skenario masa depan untuk mengoptimalkan efisiensi dan membantu pengambilan keputusan (Bae, Choi & Lee, 2025) (Trivedi, 2024). DT awalnya dikembangkan oleh NASA sekitar tahun 2010 untuk mereplikasi sistem yang kompleks. NASA mengadopsi arsitektur tiga dimensi DT seperti yang diusulkan oleh Grieves (Tudor et al., 2025).

Tabel 1. Tabel perbandingan dengan penelitian sebelumnya

Penelitian	Hasil	Perbedaan
Developing a digital twin framework for remotely monitoring nuclear reactor facilities Hergesheimer, N (2026)	Mengembangkan kerangka kerja DT yang berfungsi untuk memantau performa dan kondisi keamanan fasilitas reaktor nuklir dari jarak jauh melalui representasi virtual yang dinamis.	Mengalihkan fokus implementasi dari aset industri berskala besar ke ranah biologis dengan membangun DC yang memvisualisasikan data tubuh manusia secara personal melalui konsep HDT.
Digital twin for Electronic Centralized Aircraft Monitoring by machine learning algorithms Kilic, U (2023)	Mengimplementasikan sistem pemantauan parameter mesin pesawat berbasis platform MATLAB yang menggunakan algoritma <i>machine learning</i> untuk mendeteksi anomali performa secara otomatis.	Mentransformasi penggunaan sistem pemantauan dari mesin mekanik pesawat menjadi DC berbasis web untuk memantau indikator fisiologis jantung pasien secara <i>real-time</i> dan berkelanjutan.
A modular computational framework for medical digital twins Masison, J (2021)	Mengusulkan <i>hub-&-spoke</i> sebagai kerangka kerja antarmuka belakang untuk mengintegrasikan berbagai model matematika dan biologis yang berbeda	Mentransformasi arsitektur monolitik menjadi modular untuk meningkatkan efisiensi dan independensi pengembangan fitur.
Building a Human Digital Twin (HDTwin) Using Large Language Models for Cognitive Diagnosis: Algorithm Development & Validation Sprint, G (2024)	Membangun HDT menggunakan <i>Large Language Models</i> (LLM) untuk diagnosis kognitif.	Menggunakan LLM secara modular sebagai layanan pendukung rekomendasi teks pada DC, bukan sebagai inti pemrosesan utama diagnosis.

Arsitektur lima dimensi, yang kemudian didefinisikan oleh Tao, terdiri dari Entitas Fisik, Entitas Virtual, Transmisi Data, Koneksi, dan layanan (Annafii & Biddinika, 2025). Konsep ini kemudian diadaptasi menjadi HDT, yang mereplikasi individu manusia, baik seluruh tubuh, satu organ, atau sistem biologis (Tudu, Sharma & Kumar, 2025). Perbedaan mendasar adalah bahwa DT pada mesin mereplikasi aset yang perubahannya relatif dijelaskan oleh fisika dan matematika, sementara HDT mereplikasi biologi dan kesadaran manusia, yang jauh lebih kompleks dan dipengaruhi oleh ribuan variabel genetik, lingkungan, dan perilaku yang belum sepenuhnya dipahami.

DC didefinisikan sebagai *interface* perangkat lunak yang berfungsi untuk mengintegrasikan, memvisualisasikan, dan menganalisis semua data multivariat yang dikumpulkan dari HDT, mulai dari data genomik statis hingga aliran data fisiologis *real-time* (Roopa & Venugopal, 2025). Fungsi utamanya adalah sebagai jendela interaksi bagi pengguna seperti petugas kesehatan, peneliti, atau individu itu sendiri dengan kembaran digital mereka (Mokhtari, 2025). Hubungan antara *interface* visual dan inti pemrosesan. DC adalah User Interface (UI) yang dirancang untuk memungkinkan dan menerjemahkan fungsi inti dari model DT yang kompleks, menjembatani model virtual dengan pengguna.

Desain pusat data yang efektif membutuhkan arsitektur perangkat lunak yang mudah dipelihara untuk menangani aliran data multimodal berkelanjutan yang melekat pada sistem HDT. Mengingat kebutuhan akan pemantauan *real-time* dan integrasi yang mudah dengan komponen yang luas seperti layanan AI dan LLM, struktur sistem yang mendasarinya sangat penting.

Pemisahan lapisan antarmuka depan dan antarmuka belakang, yang biasa disebut arsitektur

decoupled, secara khusus meningkatkan skalabilitas dan pemeliharaan karena memungkinkan spesialisasi fungsional dan kemiripan pengembangan (Collins et al., 2021), hal ini memungkinkan kedua lapisan untuk menangani berbagai jenis beban kerja dan dapat diskalakan secara terpisah sesuai dengan kebutuhan lalu lintas atau pemrosesannya masing-masing.

Perkembangan teknologi seluler menyebabkan munculnya aplikasi berbasis internet, inovasi infrastruktur jaringan yang massif mengabaikan meningkatnya aksesibilitas internet dan mengubah model distribusi perangkat lunak (Agrawal, 2023). Meskipun aplikasi *native* menawarkan kinerja dan akses perangkat keras yang lebih unggul, akan tetapi pengembangannya menghadapi tantangan meliputi kebutuhan akan pembuatan kode g&a, siklus penyebaran yang kompleks melalui *app store*, biaya dan sumber daya manusia yang tinggi, dan masalah fragmentasi perangkat, khususnya pada &roid (Mahmud et al., 2025).

Aplikasi *native*, yang dibangun khusus untuk *Operating System* (OS) tertentu, seperti iOS atau &roid, memberikan kinerja yang unggul, akses perangkat keras, dan pengalaman pengguna yang unggul. Namun, aplikasi *native* membutuhkan biaya pengembangan dan pemeliharaan yang sangat tinggi karena kebutuhan akan pembuatan kode g&a (Souha et al., 2024). Sebaliknya, Aplikasi Web adalah situs web yang diakses melalui *browser*, menggunakan basis satu kode web universal, sehingga biayanya lebih rendah, tetapi memiliki keterbatasan dalam akses ke perangkat keras dan sangat bergantung pada koneksi internet (Xie et al., 2023).

Untuk menjembatani kesenjangan ini, Aplikasi Hibrida muncul, yang membungkus kode web dalam wadah untuk memungkinkan distribusi melalui *App Store* dan akses ke fitur *native* (Wu et al., 2022). Meskipun menawarkan basis kode tunggal dan universal.

Tabel 2. Perbedaan utama dalam aplikasi seluler (BrowserStack, 2025)

Criteria	Native App	Hybrid App	Web App	PWA
<i>Platform</i>	Berjalan pada platform spesifik (iOS, &roid).	Berjalan di banyak platform dengan satu <i>codebase</i> .	Berjalan di <i>web browser</i> pada berbagai perangkat.	Berjalan di <i>web browser</i> dengan pengalaman menyerupai aplikasi.
<i>Development</i>	Menggunakan bahasa dan alat khusus platform.	Menggunakan teknologi <i>web</i> dengan <i>native container</i> .	Dibangun dengan HTML, CSS, dan JavaScript.	Teknologi <i>web</i> dengan <i>service workers</i> .
<i>Performance</i>	Performa tinggi dan pengalaman mulus.	Performa cukup baik, di bawah <i>native app</i> .	Bergantung pada browser dan koneksi internet.	Performa baik dengan <i>offline dan caching</i> .
<i>User Experience</i>	Desain dan interaksi disesuaikan per platform.	Desain seragam lintas platform dengan kompromi.	Bergantung pada browser rendering.	Pengalaman mirip aplikasi, <i>load</i> cepat dan <i>offline</i> .
<i>Offline Functionality</i>	Terbatas tanpa konfigurasi tambahan.	<i>Offline</i> bergantung implementasi.	Mayoritas <i>online</i> , <i>offline</i> minimal.	<i>Offline access</i> dengan <i>caching</i> dan <i>service workers</i> .

Jenis keempat adalah PWA, yang merupakan aplikasi web yang ditingkatkan menggunakan teknologi *Service Worker* dan *Web App Manifest* untuk memberikan pengalaman seperti aplikasi native (Carmen Martín Gaité, 2021). PWA unggul karena menawarkan kemampuan *offline-first*, distribusi yang sangat mudah seperti aplikasi web melalui tautan dan tanpa *App Store*, dan pembaruan otomatis yang cepat (Leshchuk et al., 2022). Tabel 2 memberikan perbandingan rinci tentang perbedaan teknis dan pengalaman utama antara keempat platform aplikasi utama ini.

Tabel 2 membandingkan empat platform pengembangan aplikasi antarmuka depan utama: Aplikasi *Native*, Aplikasi *Hybrid*, Aplikasi Web, dan PWA, berdasarkan kriteria seperti Kinerja, Pengembangan, dan Fungsionalitas *Offline*. Secara tradisional, Aplikasi *Native* menawarkan kinerja yang unggul tetapi memiliki biaya pengembangan yang tinggi karena basis kode *g&a*, sementara Aplikasi Web memiliki biaya rendah tetapi fungsionalitas *offline* yang sangat terbatas. Aplikasi *Hybrid* mencoba menjembatani kesenjangan ini tetapi seringkali menghasilkan kinerja yang lebih rendah dibandingkan dengan Aplikasi *Native*. PWA muncul sebagai solusi penyeimbang yang ideal karena memanfaatkan teknologi web ditambah *Service Worker* untuk fitur yang lebih baik. Kemampuan PWA ini secara langsung mengatasi keterbatasan fungsionalitas *offline* di Aplikasi Web dan kompleksitas distribusi di Aplikasi *Native*, memvalidasi PWA sebagai pilihan yang tepat untuk DC HDT.

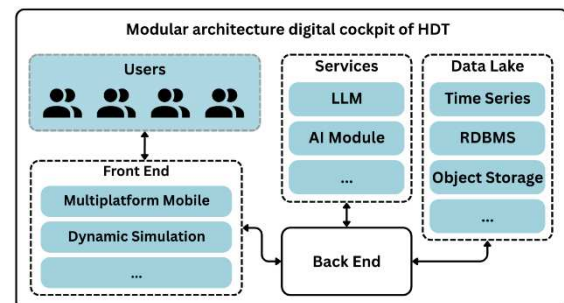
PWA adalah aplikasi web yang dibangun menggunakan teknologi web modern (HTML, CSS, JavaScript) dan dilengkapi dengan kemampuan seperti *Service Worker* dan *Web App Manifest* untuk meniru pengalaman aplikasi *native*. PWA didukung oleh tiga pilar teknologi: *Service Worker*, yang bertindak sebagai proxy jaringan untuk *caching* dan kemampuan *offline-first* (Haryanto & Saputra Elsi, 2021), tugas *Web App Manifest* adalah untuk menentukan tampilan pada layar *ber&a*, dan HTTPS

untuk memastikan keamanan (Leshchuk et al., 2022). PWA merupakan pilihan ideal untuk DC HDT karena menggabungkan aksesibilitas web dengan kinerja seperti aplikasi *native*. PWA mengatasi keterbatasan aplikasi *native* dalam hal distribusi dan pembaruan, memungkinkan distribusi yang mudah pembaruan yang cepat, otomatis.

2. METODE PENELITIAN

2.1. Arsitektur Sistem Digital Cockpit

Berfokus pada implementasi PWA untuk antarmuka depan, penelitian ini mengadopsi desain arsitektur modular untuk mendukung persyaratan skalabilitas dalam konteks HDT. Desain ini secara fundamental memisahkan lapisan antarmuka depan dari lapisan antarmuka belakang, untuk mewujudkan tujuan utama penelitian ini dan memastikan pembaruan, pengembangan, atau penambahan layanan dapat dilakukan secara independen di seluruh sistem tanpa mengganggu seluruh aplikasi,. Arsitektur modular yang diusulkan memastikan bahwa sistem DC dapat mengakomodasi berbagai antarmuka pengguna, sambil tetap terpusat pada satu sumber data.



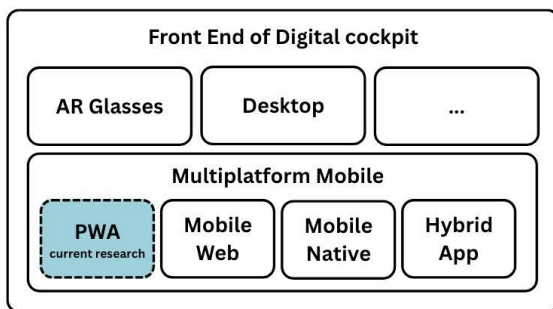
Gambar 1. Arsitektur sistem DC modular untuk HDT

Seperti yang diilustrasikan pada Gambar 1, arsitektur sistem DC modular untuk HDT terdiri dari empat komponen utama yang saling terhubung, yaitu: antarmuka depan, antarmuka belakang, *Services*, dan *Data Lake*. Pada lapisan *interface*, pengguna berinteraksi dengan antarmuka depan, yang berfungsi

sebagai antarmuka pengguna yang mencakup *Mobile Multiplatform* dan *Dynamic Simulation*. antarmuka depan ini kemudian berkomunikasi dengan antarmuka belakang melalui *Application Programming Interface* (API). Antarmuka belakang utama bertindak sebagai pusat kendali yang bertanggung jawab untuk menangani logika bisnis, mengelola sesi pengguna, dan mengarahkan permintaan data *real-time* ke *services* yang dibutuhkan. Selanjutnya, terdapat *Data Lake*, yang berfungsi sebagai repositori data skala besar untuk menampung berbagai jenis data. Terakhir, sistem ini dilengkapi dengan Layanan, seperti LLM dan Modul AI. Layanan-layanan ini bersifat modular atau dapat dipasang dan dilepas.

2.2. Fokus Implementasi Antarmuka Depan

Implementasi DC dapat dilakukan pada berbagai jenis aplikasi seluler, yaitu *Mobile Native*, *Mobile Web Application*, PWA, dan *Hybrid Application*, yang masing-masing menawarkan keunggulan. Untuk penelitian ini, fokus akan diarahkan pada pengoptimalan dan pengujian implementasi DC pada satu jenis platform, yaitu PWA.



Gambar 2. Antarmuka pengguna yang dapat diimplementasikan DC

Seperti yang diilustrasikan pada Gambar 2, DC berpotensi digunakan pada berbagai perangkat seperti *AR Glasses*, *Desktop*, *Mobile Multiplatform*, dan *interface* potensial lainnya. Dalam ruang lingkup *Mobile Multiplatform*, terdapat empat jenis platform aplikasi yang berbeda: *Mobile Native*, *Mobile Web Application*, PWA, dan *Hybrid Application*, yang dipilih sebagai fokus penelitian ini. Pemilihan PWA didasarkan pada keunggulannya dalam menjembatani kesenjangan antara fleksibilitas web dan fungsionalitas native. Implementasi DC saat ini bertujuan untuk mengeksplorasi bagaimana PWA dapat memanfaatkan arsitektur antarmuka belakang modular yang telah dirancang untuk menyediakan pemantauan kesehatan kardiovaskular secara *real-time* dengan kinerja optimal.

2.3. Prototipe Pembuktian Konsep

Kelayakan penelitian ini ditunjukkan melalui pengembangan prototipe bukti konsep DC yang dirancang sebagai PWA. Implementasi tersebut

menggunakan *framework* Laravel versi 11 yang dikonfigurasi memanfaatkan beberapa API untuk menghubungkan antarmuka depan dengan antarmuka belakang. Implementasi ini juga mengadopsi arsitektur modular untuk memastikan skalabilitas dan independensi fungsional. Arsitektur sistem DC secara keseluruhan dalam konteks HDT terdiri dari empat komponen utama yang saling terhubung: Pengguna, antarmuka depan, antarmuka belakang, *Services*, dan *Datalake*. Desain terpisah hal ini memungkinkan pembaruan, pengembangan, atau penambahan layanan dilakukan secara independen tanpa mengganggu aplikasi secara keseluruhan.

Desain UI dimulai dengan *wireframe* yang sepenuhnya mematuhi prinsip desain responsif dan pendekatan *Mobile-First* untuk memastikan konsistensi data di seluruh perangkat. *Wireframe* tersebut menguraikan dasbor utama yang mengelola biodata pasien, analisis kesehatan, dan blok *vital sign* yang penting untuk visualisasi data secara *real-time*. Semua data dan informasi berkaitan pasien yang ditampilkan dalam prototipe ini merupakan data dan informasi tiruan (*dummy*) untuk menjaga privasi dan memenuhi aspek etika dalam bidang medis.

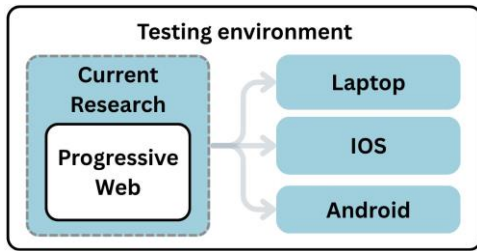
Antarmuka depan diimplementasikan menggunakan teknologi PWA karena kemampuannya untuk memberikan pengalaman seperti aplikasi *native*, kemudahan distribusi, dan kemampuan *offline*. Fungsionalitas *offline* yang penting ini diaktifkan oleh *Service Worker*, yang bertindak sebagai *proxy* jaringan yang berjalan di latar belakang browser. Saat pengguna *offline*, *Service Worker* mencegat permintaan dan mengambil aset penting yang di-*cache* dan data sinkronisasi terbaru dari *Cache Offline*.

Antarmuka belakang diimplementasikan untuk bertindak sebagai pusat kendali utama untuk semua logika bisnis, sekaligus mengelola akuisisi data dan integrasi layanan. Antarmuka belakang ini menangani penyerapan berbagai aliran data perawatan kesehatan dari *Datalake*, termasuk data deret waktu *real-time* dari sensor biometrik dan rekam medis. Yang terpenting, antarmuka belakang menyediakan antarmuka untuk mengintegrasikan layanan modular seperti Modul AI untuk prediksi atau pelatihan dan LLM untuk menghasilkan rekomendasi tekstual, serta mengarahkan data yang diproses ke dan dari komponen yang dapat dihubungkan ini. Struktur ini memastikan tata kelola data terpusat dan fleksibilitas dalam mengintegrasikan berbagai model, sambil menjaga agar antarmuka depan PWA tetap ringan.

2.4. Lingkungan Pengujian

Implementasi PWA DC akan diuji untuk memvalidasi fungsionalitas, kompatibilitas, dan responsivitasnya di berbagai lingkungan, khususnya yang terkait dengan aplikasi seluler dan web. Evaluasi ini akan menggunakan berbagai perangkat,

termasuk laptop, perangkat iOS, dan &roid, untuk memastikan kinerja lintas platform yang optimal.



Gambar 3. Lingkungan Pengujian

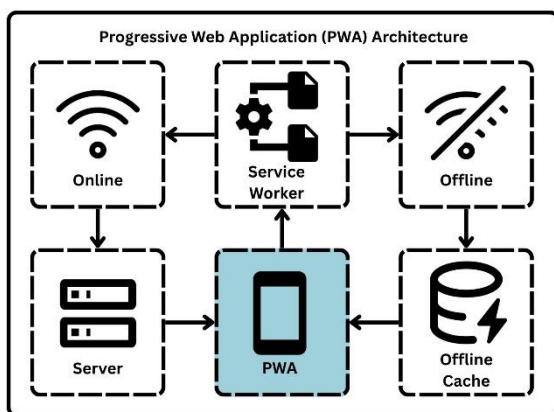
Lingkungan pengujian yang ditunjukkan pada Gambar 3 mencakup tiga lingkungan utama: Laptop, &roid, dan iOS. Pengujian di ketiga lingkungan ini bertujuan untuk memastikan DC PWA mempertahankan konsistensi antarmuka pengguna dan kinerja *real-time* di seluruh sistem operasi seluler, serta memverifikasi aksesibilitas dan fungsionalitas penuhnya saat diakses melalui browser di perangkat desktop dan seluler. Detail teknis dan hasil pengujian ini akan dijelaskan lebih lanjut di bagian Hasil dan Diskusi.

3. HASIL DAN DISKUSI

3.1. Hasil Implementasi Sistem

3.1.1. Arsitektur PWA

Penelitian ini berfokus pada implementasi DC pada arsitektur PWA. Arsitektur PWA DC pada dasarnya dirancang untuk mendukung kemampuan *offline-first*, memisahkan lapisan antarmuka depan yang ringan dari logika bisnis yang kompleks di antarmuka belakang utama. Mekanisme inti dari PWA ini bergantung pada *Service Worker*, sebuah skrip yang berjalan di latar belakang *browser* dan bertindak sebagai *proxy* jaringan, seperti yang digambarkan pada Gambar 4.



Gambar 4. Arsitektur PWA

Gambar 4. Menunjukkan ketika pengguna memulai permintaan dari perangkat mereka, *Service Worker* segera mencegat permintaan tersebut. Jika perangkat *online* atau terhubung ke jaringan, permintaan diprioritaskan dan diteruskan langsung ke

antarmuka belakang utama. Sebaliknya, jika perangkat *offline*, *Service Worker* segera mengambil aset penting seperti shell dan *interface* aplikasi, serta data yang paling baru disinkronkan dari *Cache RAM Offline* atau Penyimpanan Lokal. Mekanisme *caching* dan *routing* ini memastikan respons cepat kepada pengguna di berbagai kondisi jaringan, sekaligus mengoptimalkan komunikasi dengan arsitektur antarmuka belakang modular, memvalidasi efisiensi PWA sebagai jalur antarmuka depan ideal untuk infrastruktur yang dapat diskalkan.

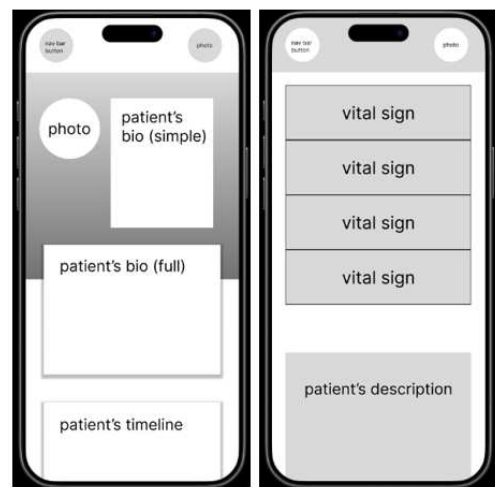
3.1.2. Wireframe

Desain UI PWA DC dibuat melalui proses *wireframing*, yang berfokus pada desain responsif untuk *Desktop* dan *Mobile*, serta penyajian informasi kesehatan kardiovaskular yang intuitif.

Pada gambar 5 menyajikan kerangka kerja versi *Desktop*. Sisi atas menunjukkan halaman utama yang berfungsi sebagai dasbor ringkasan, dilengkapi dengan *top bar* untuk akses cepat ke fitur-fitur seperti layar penuh dan pengalih tema, serta *side bar* navigasi yang konsisten. konten utama terdiri dari sapaan, foto profil, dan analisis kesehatan, yang direncanakan untuk menyajikan ringkasan atau rekomendasi yang dihasilkan oleh LLM.



Gambar 5. Wireframe PWA versi desktop



Gambar 6. Wireframe PWA versi mobile

Pada Gambar 5 menunjukkan proses pembuatan *wireframe* aplikasi versi *desktop*. Bagian utama adalah blok *vital sign* yang menampilkan data sensor *real-time* dari *wearable device*, diikuti oleh *Examination History Table* dan *Daily Task Table* untuk p&uan aktivitas harian yang dipersonalisasi. Sisi bawah menunjukkan kerangka kerja halaman detail pasien yang menyediakan informasi pasien yang komprehensif, termasuk biodata sederhana pasien dan detail pasien, t&a vital pasien, garis waktu pasien yang mendokumentasikan riwayat perjalanan pasien, tujuan dan aktivitas, dan tindakan untuk tugas medis harian tertentu.

Selanjutnya, Gambar 6 menunjukkan versi *mobile* dari *wireframe*, yang dirancang responsif untuk mengakomodasi layar vertikal. Sisi kiri *wireframe mobile* menampilkan informasi utama: Top Bar menampilkan tombol navigasi untuk membuka *Sidebar* dan ikon profil, diikuti oleh presentasi data yang tersusun vertikal, seperti biodata pasien (sederhana), biodata pasien (lengkap), dan pasien's time line. Saat pengguna menggulir ke bawah, tampilan berubah (sisi kanan), menampilkan blok *vital sign* yang tersusun vertikal dan menonjol, seperti tekanan darah, detak jantung, SpO2, dan durasi tidur, diikuti oleh deskripsi pasien yang merinci kondisi klinis pasien. Desain *wireframe* adaptif ini memastikan bahwa semua data HDT, baik untuk konsumsi *desktop* maupun *mobile*, tetap dapat diakses dengan cepat dan terorganisir dengan baik, yang merupakan hal mendasar untuk pengalaman pengguna yang sukses.

3.1.3. Hasil Antarmuka Pengguna

Setelah fase implementasi, PWA DC divalidasi melalui pengujian di berbagai lingkungan operasional untuk memastikan kesesuaiannya. Pengujian ini sangat penting untuk menunjukkan bahwa PWA, sebagai antarmuka depan yang dipilih, mampu berkomunikasi secara optimal dengan antarmuka belakang modular yang dirancang dan memberikan pengalaman pengguna yang konsisten di berbagai perangkat.

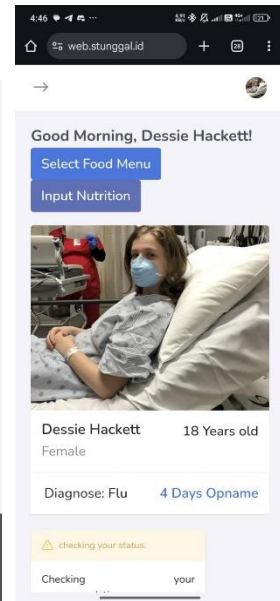
Seperti yang dijelaskan di bagian pengujian, implementasi ini secara khusus diuji di tiga lingkungan utama: Desktop, iOS, dan &roid. Visualisasi hasil pengujian ini menunjukkan keberhasilan PWA dalam mempertahankan konsistensi antarmuka pengguna di berbagai sistem operasi.

Tampilan pada Gambar 7 menunjukkan bagaimana PWA ditampilkan pada sistem iOS. Responsivitas PWA terlihat jelas melalui susunan elemen dasbor, yang secara otomatis tersusun dalam satu kolom untuk mengakomodasi layar vertikal yang sempit. Meskipun diakses melalui browser Safari pada perangkat seluler, antarmuka dengan jelas menyajikan data pasien untuk "Anna binti Anna," 18 Tahun, Diagnosis: Flu, diikuti oleh blok data *vital sign* seperti HEART RATE (71,6 Bpm). Format

tampilan ini memvalidasi keberhasilan PWA dalam mempertahankan fungsionalitas dan keterbacaan data medis penting di lingkungan seluler.

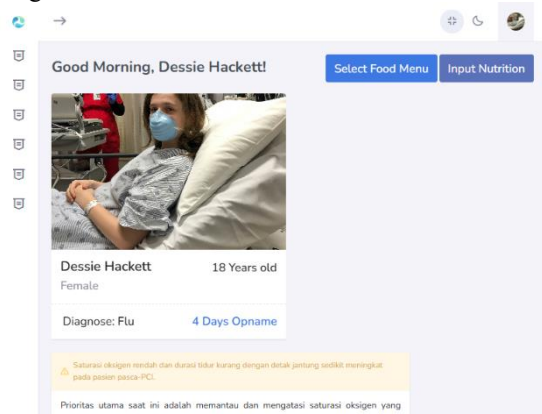


Gambar 7. Tampilan iOS



Gambar 8. Tampilan &roid

Gambar 8 menyajikan tampilan layar penuh PWA di lingkungan desktop. Tampilan ini memanfaatkan lebar layar yang lebih besar dengan mengatur elemen secara horizontal, mencerminkan desain dasbor ikhtisar. *Vital sign data card*, seperti HEART RATE (71,6 Bpm) dan BLOOD PRESSURE (92/75 mmHg), disusun berdampingan, memberikan tata letak yang efisien untuk kebutuhan data para profesional perawatan kesehatan. Rendering yang sukses ini memvalidasi bahwa antarmuka depan PWA mampu merespons permintaan dan menampilkan data yang diambil dari antarmuka belakang modular secara &al pada layar yang lebih lebar.



Gambar 9. Tampilan desktop

Gambar 9 menyajikan hasil pengembangan antarmuka aplikasi versi desktop yang mengadopsi elemen kerangka halaman utama secara konsisten.

Tabel 3. Tabel *pengujian blackbox*

No	Tested Component	Test Scenario	Status
1	Navigation Menu	Click the "Patient" menu	(√) Pass
		Click the "Doctor" menu	(√) Pass
		Click the "Patient Detail" menu	(√) Pass
		Click the "Patient List" menu	(√) Pass
		Click the "Model" menu	(√) Pass
2	Organ Model Filter	Select the "Utuh" option	(√) Pass
		Select the "urat" option	(√) Pass
		Select the "dalam" option	(√) Pass
		Select the "setengah" option	(√) Pass
3	Patient Profile	Verify patient data display (name, age, gender)	(√) Pass
		Verify diagnosis info & length of stay	(√) Pass
4	Health Status	Verify Heart Rate display (0 Bpm)	(√) Pass
		Verify Blood Pressure display (120/74 mmHg)	(√) Pass
		Verify SpO2 display (94%)	(√) Pass
		Verify Sleep Duration display (7.8 H)	(√) Pass
5	Medical Analysis	Verify patient condition analysis display	(√) Pass
		Verify displayed medical recommendations	(√) Pass
6	Examination History	Scroll through the examination history list	(√) Pass
		Verify columns (Event, Performed At, By, Notes)	(√) Pass
		Verify date & time of examination	(√) Pass
7	Daily Tasks	Verify list of 10 daily tasks	(√) Pass
		Verify status "10 of 10 remaining"	(√) Pass
		Verify time & description of each task	(√) Pass
		Simulate checking completed tasks	(√) Pass
8	Layout Responsiveness	Test on mobile screen size	(√) Pass
		Test on tablet screen size	(√) Pass
		Test on desktop screen size	(√) Pass
		Test scrolling in each section	(√) Pass
9	Back Navigation	Click the back/return button from each page	(√) Pass
		Click navigation to another menu	(√) Pass
10	Data Refresh	Pull-to-refresh on health data	(√) Pass
		Pull-to-refresh on examination history	(√) Pass
		Pull-to-refresh on daily tasks	(√) Pass

Visualisasi data kesehatan ditampilkan secara *real-time* melalui *dasboard* interaktif, yang mencakup parameter krusial seperti HEART RATE (71,6 Bpm), BLOOD PRESSURE (92/75 mmHg), SpO2 (93%), dan Sleep Time (7,8 H). Informasi tersebut disajikan dalam format kartu berwarna untuk mempermudah keterbacaan, yang kemudian dilengkapi dengan tabel EXAMINATION HISTORY dan DAILY TASK sebagai instrumen pemantauan aktivitas medis serta p&uan tugas harian pengguna.

3.2. Pengujian

3.2.1. Blackbox

Pengujian *blackbox* pada DC HDT jantung dilakukan untuk memverifikasi fungsionalitas sistem tanpa mengetahui struktur kode internal. Pendekatan ini berfokus pada perspektif pengguna akhir, dengan memastikan seluruh komponen antarmuka berinteraksi secara benar sesuai dengan kebutuhan klinis. DT jantung, sebagai representasi virtual dari organ pasien, memerlukan pengujian yang menyeluruh untuk menjamin keakuratan data medis serta kelengkapan fitur pemantauan.

Pengujian mencakup seluruh alur kerja sistem, mulai dari navigasi menu dan visualisasi model jantung, penampilan parameter fisiologis, hingga pengelolaan tugas harian. Setiap komponen diuji berdasarkan skenario nyata yang digunakan oleh tenaga kesehatan, dengan penekanan khusus pada akurasi data kesehatan kardiovaskular dan konsistensi informasi pada berbagai tampilan model jantung.

Hasil *blackbox testing* yang dipaparkan pada Tabel 3 menunjukkan kinerja optimal dari seluruh komponen DC dengan tingkat keberhasilan seratus persen. Sistem mampu menampilkan data *real-time* parameter jantung secara akurat, seperti heart rate, blood pressure, dan SpO2, serta menyediakan analisis medis yang komprehensif untuk mendukung pengambilan keputusan klinis. Navigasi antar berbagai tampilan model jantung (*whole, vein, deep, dan half*) berjalan mulus tanpa gangguan teknis.

Fitur daily task management dan examination history terbukti &al dalam mendukung alur kerja perawatan pasien. Responsivitas tata letak pada berbagai ukuran layar memastikan pengalaman pengguna yang konsisten baik pada perangkat mobile maupun desktop. Hasil ini menegaskan kesiapan sistem untuk diterapkan pada lingkungan klinis yang menuntut tingkat ke&alan tinggi.

3.2.2. Kompatibilitas

Pengujian kompatibilitas dilakukan untuk memastikan DC dari HDT jantung dapat beroperasi secara optimal pada berbagai perangkat dan *platform* yang digunakan oleh tenaga kesehatan. Parameter yang diukur meliputi stabilitas tata letak, responsivitas antarmuka, kecepatan pemuatan data vital jantung, serta kelancaran navigasi antar tampilan

model jantung. Pendekatan pengujian menggunakan skala Likert empat poin untuk menilai kualitas pengalaman pengguna pada setiap perangkat, dengan mempertimbangkan kebutuhan visualisasi *real-time* dari *heart* DT. Perangkat yang diuji merepresentasikan ekosistem teknologi yang beragam, mulai dari perangkat *flagship* hingga *mid-range smartphone* yang umum digunakan dalam konteks layanan kesehatan.

Tabel 4. Tabel pengujian kompatibilitas

#	C1	C2	C3	C4	C5	C6	C7
D1	True	4	4	4	4	16/16	100%
D2	True	3	3	4	3	13/16	81.25%
D3	True	4	3	4	3	14/16	87.50%
D4	True	3	3	3	2	11/16	68.75%
D5	True	4	4	4	4	16/16	100%
D6	True	4	4	4	4	16/16	100%

Pada Tabel 4, digunakan sistem pengkodean untuk menyederhanakan penyajian hasil pengujian kompatibilitas. Kode D1 hingga D6 merepresentasikan perangkat uji, yaitu D1: iPhone 11 Pro Max, D2: Infinix Hot 50, D3: Poco X6, D4: Oppo, D5: Windows Desktop, dan D6: MacBook Pro. Kode C1 menunjukkan status implementasi, di mana nilai *True* menandakan aplikasi DC dapat berjalan dengan baik tanpa kendala fungsional pada perangkat tersebut. Kode C2 merepresentasikan *layout stability*, C3 menunjukkan *interface responsiveness*, C4 merepresentasikan *data loading speed*, dan C5 menunjukkan *navigation smoothness*, yang seluruhnya dinilai menggunakan skala *Likert* empat poin, yaitu nilai 4 (*Very Good*), 3 (*Good*), 2 (*Fair*), dan 1 (*Poor*). Kode C6 menunjukkan total *Likert score* yang diperoleh dari akumulasi nilai C2 hingga C5 terhadap skor maksimum 16, sedangkan C7 merepresentasikan tingkat *compatibility* dalam bentuk persentase.

Hasil pengujian *compatibility* menunjukkan kinerja yang sangat baik dengan nilai rata-rata sebesar 89,58%. Perangkat *flagship* seperti iPhone 11 Pro Max, Windows Desktop, dan MacBook Pro mencapai skor sempurna 100% pada seluruh parameter, yang menjamin pengalaman terbaik untuk visualisasi *digital twin* jantung secara detail. Perangkat *mid-range* seperti Poco X6 dan Infinix Hot 50 juga menunjukkan kinerja yang sangat memuaskan dengan nilai di atas 80%, yang mengindikasikan aksesibilitas sistem yang luas.

Meskipun perangkat Oppo menunjukkan performa yang lebih rendah pada aspek navigasi, secara keseluruhan sistem tetap terbukti sangat kompatibel di berbagai platform.

3.3. Diskusi

Implementasi PWA sebagai DC yang terintegrasi dengan arsitektur modular HDT telah berhasil memvalidasi solusi terhadap keterbatasan yang diidentifikasi dalam penelitian sebelumnya, khususnya mengenai skalabilitas, fleksibilitas

antarmuka, dan dukungan operasional offline. Mengadopsi arsitektur terpisah yang memisahkan antara antarmuka depan dan antarmuka belakang telah terbukti menjadi pilihan yang lebih baik daripada arsitektur monolitik. Pemisahan ini membuat setiap fungsi sistem bekerja lebih fokus pada tugasnya dan memungkinkan pengembangan fitur dilakukan secara *miri* tanpa saling mengganggu. Secara spesifik, arsitektur modular ini mengurangi beban kerja pengembangan karena hanya menggunakan satu basis kode untuk semua platform. Hal ini menjawab permasalahan yang ada di penelitian sebelumnya yang menggunakan kode *g&a* dan membutuhkan biaya yang lebih besar.

Temuan utama dari penelitian ini adalah keberhasilan PWA dalam menjembatani kesenjangan antara fleksibilitas web dan keandalan native, sekaligus secara langsung menjawab masalah keterlambatan pembaruan konten dan fragmentasi perangkat. Meskipun penelitian lain mengusulkan *hybrid application* untuk mengatasi kendala lintas platform, pendekatan tersebut masih memiliki kelemahan pada performa dan ketergantungan pada proses distribusi di *app store*. Sebaliknya, PWA yang dikembangkan menggunakan *Service Worker* memastikan fungsionalitas *offline-first* dengan tingkat keberhasilan 100% pada fitur *caching offline*, hal ini memastikan bahwa selama fase kritis pemantauan kardiovaskular, kehilangan koneksi internet tidak menyebabkan hilangnya akses data pasien yang sudah tersimpan. Mekanisme ini memastikan keberlanjutan layanan dalam kondisi jaringan yang tidak stabil.

Selain fungsionalitas, aspek kompatibilitas telah divalidasi secara ekstensif dengan skor rata-rata 89,58%. Angka ini menunjukkan peningkatan aksesibilitas yang inklusif, berbeda dengan aplikasi *hibrid* yang sering mengalami penurunan pengalaman pengguna pada perangkat spesifikasi menengah, hasil pengujian ini menunjukkan bahwa perangkat *mid-range* seperti Poco X6 dan Infinix Hot 50 tetap mampu mencapai kinerja stabil dengan skor masing-masing sebesar 87,50% dan 81,25%. Hal ini membuktikan bahwa DC yang dirancang tidak eksklusif bagi perangkat *flagship* seperti iPhone 11 Pro Max dan MacBook Pro yang mencapai 100%, melainkan mampu menjangkau infrastruktur teknologi yang beragam di lingkungan kesehatan. Dengan validasi ini, PWA terbukti sebagai pilihan ideal untuk HDT DC yang dapat diadopsi di berbagai tingkatan perangkat keras rumah sakit. Penelitian selanjutnya dapat mengeksplorasi pengujian skala penuh di lingkungan klinis nyata dan integrasi data yang lebih kompleks dengan model simulasi HDT dinamis.

4. KESIMPULAN

Penelitian ini berhasil menyajikan dan memvalidasi sistem DC inovatif untuk pemantauan HDT kardiovaskular. Dengan mengadopsi arsitektur

modular yang memisahkan antarmuka depan dan antarmuka belakang, sistem ini secara fundamental telah mencapai skalabilitas dan pemeliharaan yang unggul, membuka jalan bagi integrasi yang lebih fleksibel dengan layanan cerdas seperti AI dan LLM. Implementasi kunci dari sistem ini adalah penggunaan PWA sebagai antarmuka depan, yang telah terbukti efektif dalam memberikan pengalaman seperti aplikasi native dengan manfaat tambahan berupa fungsionalitas *offline-first* yang didukung oleh *Service Worker*. Keandalan fungsional dan kompatibilitas yang tinggi di berbagai lingkungan operasional seperti Desktop, Android, dan iOS menegaskan bahwa DC berbasis PWA ini merupakan solusi yang andal, mudah diakses, dan siap diadopsi untuk mengatasi tantangan kontinuitas layanan dalam sistem pemantauan kesehatan digital. Namun, penelitian ini memiliki beberapa keterbatasan yang dapat dikembangkan pada masa mendatang, di antaranya adalah penerapan pada *sampel* data dan *use cases* yang lebih luas serta *deployment* pada lingkungan jaringan rumah sakit yang lebih besar agar level generalisasi solusi yang ditawarkan dapat teruji lebih lanjut.

Hasil penelitian ini memiliki implikasi praktis yang signifikan jika diterapkan pada rumah sakit dan fasilitas kesehatan di Indonesia, mengingat beragamnya kapabilitas peralatan *mobile* yang digunakan masyarakat dan bervariasinya kapasitas jaringan yang tersedia, terutama jaringan seluler.

DAFTAR PUSTAKA

- ADIBI, S., RAJABIFARD, A. & SHOJAEI, D., 2024. Enhancing Healthcare through Sensor-Enabled Digital Twins in Smart Environments : A Comprehensive Analysis.
- AGRAWAL, A., 2023. The convergence of edge computing & 5G. *Control Engineering*, 70(7), pp.32–34.
- ANNAFIL, M.N. & BIDDINIKA, M.K., 2025. Human Digital Twin Modeling for Cardiovascular System. 12(1), pp.11–20. <https://doi.org/10.15294/sji.v12i1.16012>.
- ARMEL, F., SOSSO, E., COUSINEAU, M. & PAR, G., 2025. Advancing Health Care With Digital Twins : Meta-Review of Applications & Implementation Challenges Corresponding Author : 27, pp.1–12. <https://doi.org/10.2196/69544>.
- BAE, C., CHOI, E. & LEE, S., 2025. Technologies, Applications, & Challenges of Digital Twin Across Industries: A Systematic Review of the State-of-the-Art Literature. *IEEE Access*, 13(July), pp.152843–152869. <https://doi.org/10.1109/ACCESS.2025.3601615>.
- BROWSERSTACK, 2025. *How to test Native App vs Hybrid App vs Web App vs Progressive Web App (PWA)*. [online] Available at: <<https://www.browserstack.com/guide/test-native-vs-hybrid-vs-web-vs-progressive-web-app>>.
- CARMEN MARTÍN GAITE, R.M., 2021. Building Progressive Web Apps (PWA) with Angular: Enhancing Offline & Mobile Experiences. *International Journal of Trend in Scientific Research & Development (IJTSRD)*, [online] 5(Enhancing Offline & Mobile Experiences), p.9. Available at: <<https://www.ijtsrd.com/other-scientific-research-area/other/43686/building-progressive-web-apps-pwa-with-angular-enhancing-offline-&-mobile-experiences/carmen-martin-gaite>>.
- COLLINS, S.P., STORROW, A., LIU, D., JENKINS, C.A., MILLER, K.F., KAMPE, C. & BUTLER, J., 2021. A Comparative Analysis of Java Frameworks & the MERN Stack Firoz. pp.167–186.
- ELGAMMAL, Z., ALBRIJAWI, M.T. & ALHAJJ, R., 2025. Digital twins in healthcare : a review of AI - powered practical applications across health domains.
- FAUZAN, R., KRISNAHATI, I., DINDA, B., DELLA, N. & WIBOWO, A., 2022. A SYSTEMATIC LITERATURE REVIEW ON PROGRESSIVE. 33(1). <https://doi.org/10.12962/j20882033.v33i1.13904>.
- FERN&O, K., 2022. Development of Smart Healthcare Web System using PWA with AI: A Review of Related Literature.
- HARYANTO, D. & SAPUTRA ELSI, Z.R., 2021. Analisis Performance Progressive Web Apps Pada Aplikasi Shopee. *Jurnal Ilmiah Informatika Global*, 12(2), pp.106–111. <https://doi.org/10.36982/jiig.v12i2.1944>.
- HERGESHEIMER, N., RYAN, E., HEAPS, K., CAVALUZZI, J., SHIELDS, A., BROWNING, L., RITTER, C. & STEWART, R., 2026. Developing a digital twin framework for remotely monitoring nuclear reactor facilities. *Nuclear Engineering & Design*, [online] 446(PB), p.114585. <https://doi.org/10.1016/j.nucengdes.2025.114585>.
- ILIU, M., 2024. applied sciences Digital Twin — A Review of the Evolution from Concept to Technology & Its Analytical Perspectives on Applications in Various Fields.
- LESHCHUK, S.O., RAMSKYI, Y.S., KOTYK, A. V. & KUTSIY, S. V., 2022. Design a progressive web application to support student learning. *CEUR Workshop Proceedings*, 3077, pp.83–96.
- MAHMUD, T., CHE, M., NGU, A. & YANG, G., 2025. Why Android app testing falls short: empirical insights from open-source projects & a practitioner survey. *Empirical Software Engineering*, 30(6), pp.1–32.

- <https://doi.org/10.1007/s10664-025-10726-x>.
- MASISON, J., BEEZLEY, J., MEI, Y., RIBEIRO, H., KNAPP, A.C., SORDO VIEIRA, L., ADHIKARI, B., SCINDIA, Y., GRAUER, M., HELBA, B., SCHROEDER, W., MEHRAD, B. & LAUBENBACHER, R., 2021. A modular computational framework for medical digital twins. *Proceedings of the National Academy of Sciences*, [online] 118(20).
<https://doi.org/10.1073/pnas.2024287118>.
- MOKHTARI, M., 2025. Human Digital Twins in Personalized Healthcare: An Overview & Future Perspectives. [online] pp.1–22. Available at: <<http://arxiv.org/abs/2503.11944>>.
- PELLEGRINO, G., GERVASI, M., ANGELELLI, M. & CORALLO, A., 2025. A Conceptual Framework for Digital Twin in Healthcare : Evidence from a Systematic Meta-Review. pp.7–32. <https://doi.org/10.1007/s10796-024-10536-4>.
- ROOPA, M.S. & VENUGOPAL, K.R., 2025. Digital Twins for Cyber-Physical Healthcare Systems: Architecture, Requirements, Systematic Analysis, & Future Prospects. *IEEE Access*, 13(February), pp.44963–44996. <https://doi.org/10.1109/ACCESS.2025.3547991>.
- SOUHA, A., BENADDI, L., OUADDI, C. & JAKIMI, A., 2024. Comparative analysis of mobile application Frameworks: A developer's guide for choosing the right tool. *Procedia Computer Science*, [online] 236, pp.597–604.
<https://doi.org/10.1016/j.procs.2024.05.071>.
- SPRINT, G., SCHMITTER-EDGEcombe, M. & COOK, D., n.d. Building a Human Digital Twin (HDTwin) Using Large Language Models for Cognitive Diagnosis : Algorithm Development & Validation Corresponding Author : 8. <https://doi.org/10.2196/63866>.
- SUN, T., HE, X., SONG, X., SHU, L. & LI, Z., 2022. The Digital Twin in Medicine : A Key to the Future of Healthcare ? Composition of the Digital Twin. 9(July), pp.1–8. <https://doi.org/10.3389/fmed.2022.907066>.
- TRIVEDI, J., 2024. Security Enhanced Cloud-Based Remote Patient Monitoring System with Human Digital Twin & OPC UA. (May).
- TUDOR, B.H., SHARGO, R., GRAY, G.M., FIERSTEIN, J.L., KUO, F.H., BURTON, R., JOHNSON, J.T., SCULLY, B.B., ASANTE-KORANG, A., REHMAN, M.A. & AHUMADA, L.M., 2025. A scoping review of human digital twins in healthcare applications & usage patterns. *npj Digital Medicine*, 8(1).
<https://doi.org/10.1038/s41746-025-01910-w>.
- TUDU, C., SHARMA, S. & KUMAR, D., 2025. Computational Modeling & Digital Twin Technologies in Medical Device Development. *Biomedical Materials & Devices*, [online] (0123456789).
<https://doi.org/10.1007/s44174-025-00469-0>.
- WAHLSTR, M., 2017. Exploring progressive web applications for health care Developing a PWA to gather patients ' self assessments.
- WU, J., WANG, L., PEI, Q., CUI, X., LIU, F. & YANG, T., 2022. HiTDL: High-Throughput Deep Learning Inference at the Hybrid Mobile Edge. *IEEE Transactions on Parallel & Distributed Systems*, 33(12), pp.4499–4514. <https://doi.org/10.1109/TPDS.2022.3195664>.
- XIE, J., CHEN, Y., CAI, G., CAI, R., HU, Z. & WANG, H., 2023. Tree Visualization by One Table (tvBOT): A web application for visualizing, modifying & annotating phylogenetic trees. *Nucleic Acids Research*, 51(W1), pp.W587–W592. <https://doi.org/10.1093/nar/gkad359>.

Halaman ini sengaja dikosongkan