

**PERANCANGAN WEBSITE DENGAN INTEGRASI CHATBOT
UNTUK OPTIMALISASI LAYANAN PUBLIK DI RUMAH
SAKIT BHAYANGKARA BANDA ACEH**

TUGAS AKHIR

Diajukan Oleh:

RAWASIANA SAIRA

NIM. 220705011

Mahasiswa Fakultas Sains dan Teknologi

Program Studi Teknologi Informasi



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2026 M /1448 H**

LEMBAR PERSETUJUAN

PERANCANGAN WEBSITE DENGAN INTEGRASI CHATBOT UNTUK OPTIMALISASI LAYANAN PUBLIK DI RUMAH SAKIT BHAYANGKARA BANDA ACEH

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi Banda Aceh
Sebagai Salah Satu Persyaratan Penulisan Tugas Akhir
dalam Prodi Teknologi Informasi

Oleh:

Rawasiana Saira
220705011

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui untuk di Munaqasyahkan Oleh:

Pembimbing I,



Nazaruddin Ahmad, M.T.

NIP. 198206052014031002

Pembimbing II,



Malahayati, M.T.

NIP. 198301272015032003

Mengetahui,

Ketua Program Studi Teknologi Informasi



Malahayati, M.T.

NIP. 198301272015032003

LEMBAR PENGESAHAN

PERANCANGAN WEBSITE DENGAN INTEGRASI CHATBOT UNTUK OPTIMALISASI LAYANAN PUBLIK DI RUMAH SAKIT BHAYANGKARA BANDA ACEH

TUGAS AKHIR

Telah Diuji Oleh Dewan Penguji Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)
Dalam Ilmu Teknologi Informasi

Pada Hari/Tanggal : Selasa, 11 Agustus 2026 M
26 Shafar 1448 H

Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,



Nazaruddin Ahmad, M.T.
NIP. 198206052014031002

Sekretaris,



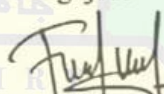
Malahavati, M.T.
NIP. 198301272015032003

Penguji I



Dr. Hendri Ahmadian, M.I.M
NIP. 198301042014031002

Penguji II



Fathiah, M.Eng
NIP. 198606152019032010

Mengetahui :

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh



Prof. Dr. Ar. Muhammad Dirhamsyah, M.T., I.P.U
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini

Nama : Rawasiana Saira

NIM : 220705011

Program Studi : Teknologi Informasi

Fakultas : Sains dan Teknologi

Judul Tugas Akhir : Perancangan Website dengan Integrasi Chatbot untuk
Optimalisasi Layanan Publik di Rumah Sakit Bhayangkara
Banda Aceh

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah tugas akhir orang lain;
3. Tidak menggunakan tugas akhir orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik tugas akhir;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri tugas akhir ini dan mampu bertanggung jawab atas tugas akhir ini.

Bila di kemudian hari ada tuntutan dari pihak lain atas tugas akhir saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang dibuktikan bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 11 Agustus 2026
yang menyatakan,



Rawasiana Saira

ABSTRAK

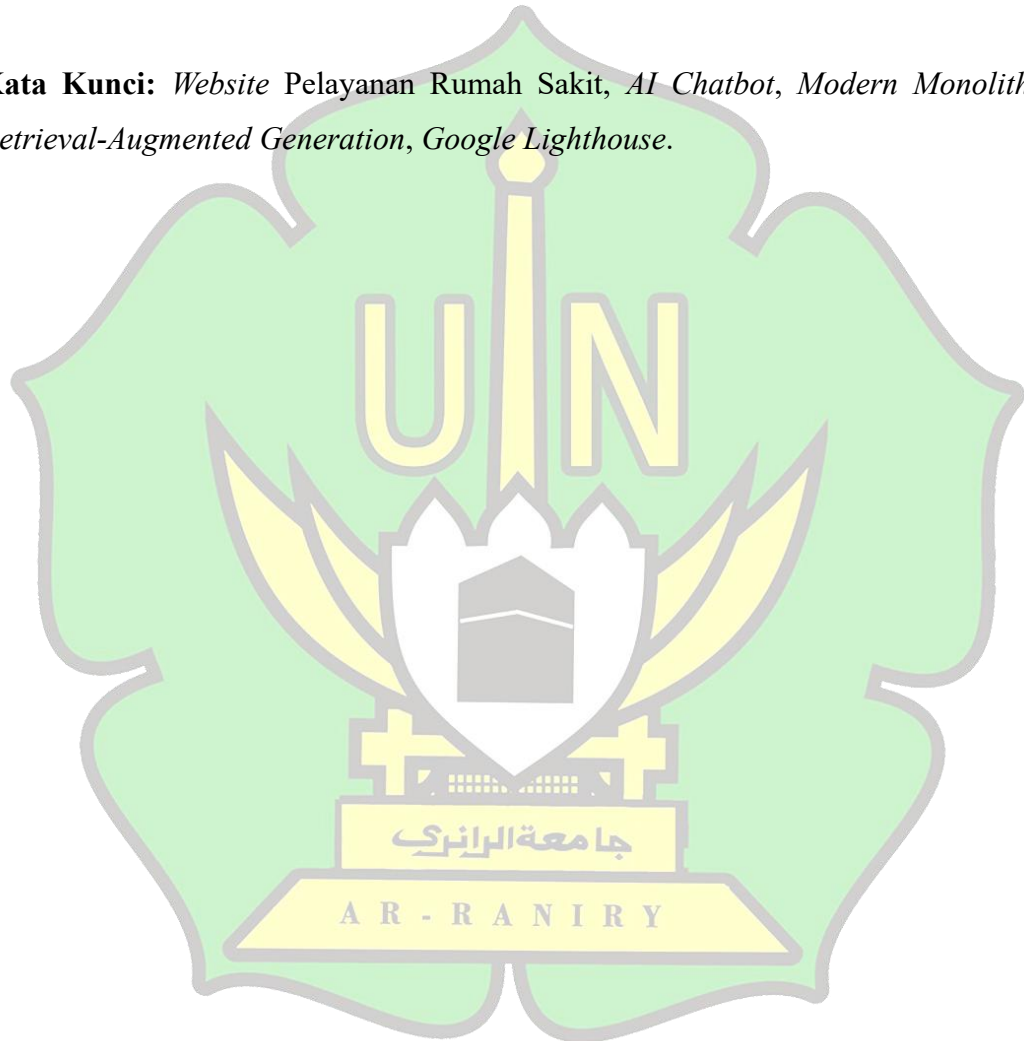
Layanan informasi publik pada instansi kesehatan seperti rumah sakit saat ini menuntut adanya aksesibilitas yang cepat, transparan, dan tersedia setiap saat tanpa hambatan birokrasi. Namun, mayoritas media informasi atau *website* rumah sakit yang ada saat ini masih bersifat statis dan searah, sehingga belum mampu merespons kebutuhan mendesak serta pertanyaan spesifik dari masyarakat secara interaktif. Permasalahan serupa dihadapi oleh Rumah Sakit Bhayangkara Banda Aceh dalam mendiseminasikan informasi krusial seperti jadwal dokter dan ketersediaan obat. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun Sistem Informasi pelayanan publik berbasis *web* yang responsif di Rumah Sakit Bhayangkara Banda Aceh dengan mengintegrasikan fitur asisten virtual (*chatbot*) cerdas sebagai media komunikasi dua arah yang proaktif.

Sistem ini dikembangkan menggunakan metode *Research and Development* (R&D) melalui pendekatan kualitatif dan studi kasus guna menangkap dinamika permasalahan serta kebutuhan riil di lapangan secara komprehensif. Dari sisi arsitektur perangkat lunak, sistem ini menerapkan konsep *Modern Monolith*, di mana penanganan *backend* dan basis data dikelola menggunakan *framework* Laravel 12 dan MySQL, sedangkan bagian *frontend* dibangun secara dinamis memanfaatkan React.js yang dijemput oleh Inertia.js untuk menghasilkan performa *Single Page Application* (SPA) yang sangat responsif. Sementara itu, fitur *chatbot* diimplementasikan menggunakan metode *Retrieval-Augmented Generation* (RAG) yang berbasis pada teknologi *Vector Embeddings*, algoritma *Cosine Similarity*, dan model AI Generatif. Pendekatan ini memastikan bahwa asisten virtual mampu memberikan jawaban yang valid, kontekstual, dan berbahasa natural berdasarkan basis pengetahuan (*Knowledge Base*) resmi rumah sakit.

Pengujian fungsionalitas sistem dilakukan melalui metode *Black Box Testing* dengan hasil yang menunjukkan bahwa seluruh modul utama, mulai dari manajemen jadwal dokter, artikel kesehatan, hingga pembaruan status obat, berjalan dengan valid. Selanjutnya, evaluasi kualitas teknis dilakukan menggunakan alat audit otomatis *Google Lighthouse* pada lingkungan *localhost*. Hasil pengujian menunjukkan skor aktual yang sangat impresif, yaitu metrik

Performance sebesar 91, *Accessibility* sebesar 87, *SEO* sebesar 92, serta capaian skor sempurna 100 pada metrik *Best Practices*. Secara akumulatif, seluruh perolehan metrik ini membuktikan bahwa sistem informasi terintegrasi *chatbot* ini dinyatakan sangat layak dan siap untuk diimplementasikan guna mengoptimalkan penyebaran informasi medis serta meningkatkan mutu pelayanan publik digital di Rumah Sakit Bhayangkara Banda Aceh.

Kata Kunci: *Website Pelayanan Rumah Sakit, AI Chatbot, Modern Monolith, Retrieval-Augmented Generation, Google Lighthouse.*



ABSTRACT

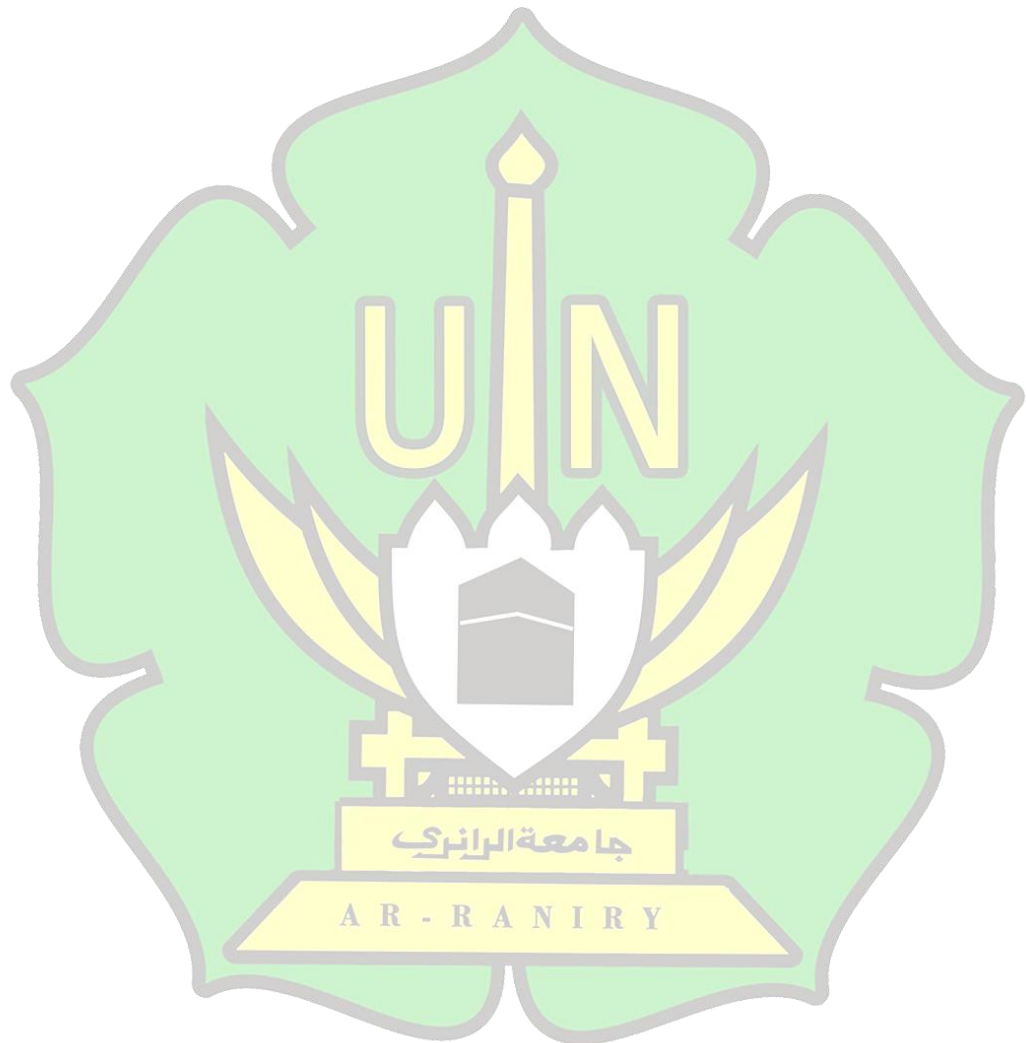
Public information services in healthcare institutions, such as hospitals, currently demand fast, transparent, and round-the-clock accessibility without bureaucratic hurdles. However, the majority of existing hospital websites remain static and one-way, failing to respond interactively to the urgent needs and specific inquiries of the community. A similar problem is faced by Bhayangkara Hospital Banda Aceh in disseminating crucial information such as doctor schedules and medicine availability. Therefore, this study aims to design and develop a responsive web-based public service Information System at Bhayangkara Hospital Banda Aceh by integrating an intelligent virtual assistant (chatbot) feature to serve as a proactive two-way communication medium.

The system was developed using the Research and Development (R&D) method with a qualitative and case study approach to comprehensively capture the real-world problems and requirements in the field. In terms of software architecture, the system implements the Modern Monolith concept, where the backend and database operations are managed using the Laravel 12 framework and MySQL, while the frontend is dynamically built utilizing React.js, bridged by Inertia.js to deliver a highly responsive Single Page Application (SPA) performance. Meanwhile, the chatbot feature is implemented using the Retrieval-Augmented Generation (RAG) method based on Vector Embeddings technology, Cosine Similarity algorithms, and Generative AI models. This approach ensures that the virtual assistant can provide valid, contextual, and naturally formulated responses based on the hospital's official Knowledge Base.

Functional evaluation of the system was conducted via Black Box Testing, demonstrating that all core modules including doctor schedule management, health articles, and medicine stock updates operated validly according to the design. Furthermore, the technical quality evaluation was performed using the Google Lighthouse automated audit tool in a localhost environment. The audit yielded highly impressive actual scores: 91 for Performance, 87 for Accessibility, 92 for SEO, and a perfect score of 100 for the Best Practices metric. Accumulatively, these metric results prove that this chatbot-integrated information system is highly feasible and ready for implementation to optimize the dissemination of medical

information and enhance the efficiency of digital public services at Bhayangkara Hospital Banda Aceh.

Keywords: Hospital Service Website, AI Chatbot, Modern Monolith, Retrieval-Augmented Generation, Google Lighthouse.



KATA PENGANTAR

Bismillahirrahmanirrahim

Assalamu'alaikum warahmatullahi wabarakatu.

Alhamdulillah rabbil'aalamin, segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu Wa Ta'ala, Tuhan Semesta Alam, yang telah melimpahkan rahmat, hidayah, serta inayah-Nya. Atas kehendak-Nya yang memberikan kekuatan, kesabaran, dan kelancaran berpikir, penulis akhirnya dapat menyelesaikan penyusunan laporan Tugas Akhir ini dengan judul **“Perancangan Website dengan Integrasi Chatbot untuk Optimalisasi Layanan Publik di Rumah Sakit Bhayangkara Banda Aceh”**.

Shalawat beriring salam senantiasa tercurah kepada junjungan kita, Nabi Besar Muhammad Shallallahu 'Alaihi Wasallam, beserta keluarga, para sahabat, dan pengikut beliau yang setia hingga akhir zaman. Beliau adalah suri teladan yang telah membawa umat manusia dari zaman kegelapan menuju zaman yang terang benderang dengan cahaya ilmu pengetahuan dan teknologi seperti yang kita rasakan saat ini.

Penyusunan Tugas Akhir ini dimaksudkan sebagai salah satu syarat akademik guna memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh. Melalui penelitian ini, penulis telah merancang dan membangun solusi teknologi berupa integrasi *chatbot* cerdas dan sistem web modern untuk membantu meningkatkan efisiensi serta kualitas pelayanan informasi publik di sektor kesehatan, khususnya di RS Bhayangkara Banda Aceh.

Penulis menyadari sepenuhnya bahwa penyusunan Tugas Akhir ini tidak akan terwujud tanpa adanya bantuan, bimbingan, motivasi, dan doa dari berbagai pihak. Dalam proses pengerjaannya, penulis banyak mendapatkan arahan yang sangat berharga dalam merancang sistem, memahami konsep *Modern Monolith*, hingga melakukan pengujian aplikasi secara komprehensif. Oleh karena itu, dengan segala kerendahan hati dan rasa hormat, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Kedua Orang Tua tercinta, Ayahanda **Saiful Raden** dan Ibunda **Wahyu Wulandari**, yang merupakan sumber motivasi terbesar dalam hidup penulis. Terima kasih yang tak terhingga atas setiap tetes keringat, pengorbanan, kasih sayang, serta doa tulus yang tidak pernah putus mengiringi setiap langkah penulis dalam menuntut ilmu.
2. Adik semata wayang tercinta, **Arief Maulana Saira**, yang selalu menjadi penghibur, teman berbagi cerita, dan penyemangat tersendiri bagi penulis di tengah kesibukan dan kelelahan menyusun Tugas Akhir ini.
3. Ibu **Malahayati, M.T.**, selaku Ketua Program Studi Teknologi Informasi, yang senantiasa memberikan dukungan administrasi serta arahan akademik demi kelancaran studi mahasiswa di lingkungan prodi.
4. Bapak **Nazaruddin Ahmad, M.T.**, selaku Dosen Pembimbing, yang telah dengan sabar meluangkan waktu, tenaga, dan pikirannya untuk membimbing penulis. Terima kasih atas segala masukan kritis, koreksi mendetail, serta diskusi-diskusi konstruktif yang diberikan selama proses perancangan hingga penyelesaian laporan penelitian ini.
5. Bapak/Ibu Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh, yang telah memfasilitasi lingkungan belajar yang kondusif bagi mahasiswa.
6. Bapak/Ibu Dosen Pengajar di Program Studi Teknologi Informasi UIN Ar-Raniry, yang telah ikhlas mentransfer ilmu pengetahuan, wawasan teknologi, serta nilai-nilai etika selama penulis menempuh pendidikan di bangku perkuliahan.
7. Ibu **Cut Ida Rahmadiana, S.Si.**, selaku Staff Operator Prodi Teknologi Informasi yang tercinta, yang selalu tanggap dan sigap dalam membantu segala urusan administrasi perkuliahan dengan tulus dan keramahannya.
8. Kepala dan Seluruh Staf Rumah Sakit Bhayangkara Banda Aceh, yang telah memberikan izin, kesempatan, dan keterbukaan bagi penulis untuk melakukan pengumpulan data, observasi, serta wawancara guna penyelesaian Tugas Akhir ini.
9. Dua sahabat tercinta, **Maqhfirah** dan **Syarifah Annisa**, terima kasih telah menjadi pendengar setia, tempat berkeluh kesah, serta *partner* diskusi yang

selalu meyakinkan penulis bahwa “kita bisa melewati ini bersama-sama” di saat-saat tersulit sekalipun.

10. Rekan-rekan seperjuangan mahasiswa Teknologi Informasi Angkatan 2022, yang telah memberikan semangat, bantuan teknis, dan kebersamaan selama masa-masa perkuliahan hingga penyusunan Tugas Akhir ini.

Penulis menyadari bahwa penulisan Tugas Akhir ini masih belum sempurna, baik dari segi penyusunan kalimat, tata bahasa, maupun kedalaman materi yang disajikan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun demi perbaikan karya-karya ilmiah selanjutnya. Akhir kata, penulis berharap semoga laporan Tugas Akhir ini dapat memberikan manfaat nyata, tidak hanya bagi penulis sebagai syarat kelulusan studi, tetapi juga dapat menjadi referensi yang berguna bagi pengembangan ilmu pengetahuan, khususnya dalam bidang implementasi teknologi informasi di sektor pelayanan publik.

Wassalamu'alaikum Warahmatullahi Wabarakatuh..



DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR	iii
ABSTRAK	iv
ABSTRACT	vi
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian	3
1.5 Batasan Penelitian	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terkait.....	6
2.1.1 Penelitian Terdahulu.....	6
2.1.2 Studi Literatur Komparatif.....	8
2.2 Landasan Konseptual	11
2.2.1 Website sebagai Media Informasi	11
2.2.2 Chatbot sebagai Asisten Layanan.....	12
2.2.3 Arsitektur Modern Monolith (Laravel, Inertia.js, React.js)	14
2.3 RS Bhayangkara Banda Aceh	15
BAB III METODE PENELITIAN	16
3.1 Jenis Penelitian.....	16
3.2 Waktu dan Lokasi Penelitian.....	16

3.3 Teknik Pengumpulan Data	16
3.4 Metode Pengembangan Sistem	17
3.5 Perancangan Sistem	25
3.5.1 Alat – alat Sistem	25
3.5.2 Perancangan Basis Data	33
3.6 Perancangan <i>Interfaces Sistem</i>	38
3.6.1 Antar Muka Administrator	38
3.6.2 Antar Muka Pengguna Umum	41
3.7 Analisis Kebutuhan Sistem	43
3.7.1 Kebutuhan Fungsionalitas Sistem	43
3.7.2 Kebutuhan Non-Fungsionalitas Sistem	45
3.8 Metode Evaluasi Perangkat Lunak	47
3.8.1 Pengumpulan Data Kinerja (<i>Data Acquisition</i>)	47
3.8.2 Logika Keputusan Kelayakan (<i>Decision Logic</i>)	48
BAB IV IMPLEMENTASI	51
4.1 Gambaran Umum Implementasi Sistem	51
4.1.1 Spesifikasi Lingkungan Pengembangan	51
4.2 Implementasi Kode Program dan Antarmuka (<i>User Interface</i>)	51
4.2.1 Implementasi Backend (Laravel 12)	52
4.2.2 Implementasi Sisi Frontend (<i>React.js + Inertia.js</i>)	67
4.3 Pengujian Fungsionalitas Sistem (<i>Black Box Testing</i>)	74
4.3.1 Sasaran Pengujian	74
4.3.2 Skenario dan Indikator Kelayakan	75
4.3.3 Pengujian Berdasarkan Klasifikasi Pengguna	75
4.4 Hasil Pengujian Google Lighthouse	77
BAB V PENUTUP	80
5.1 Kesimpulan	80
5.2 Saran	80
DAFTAR PUSTAKA	81

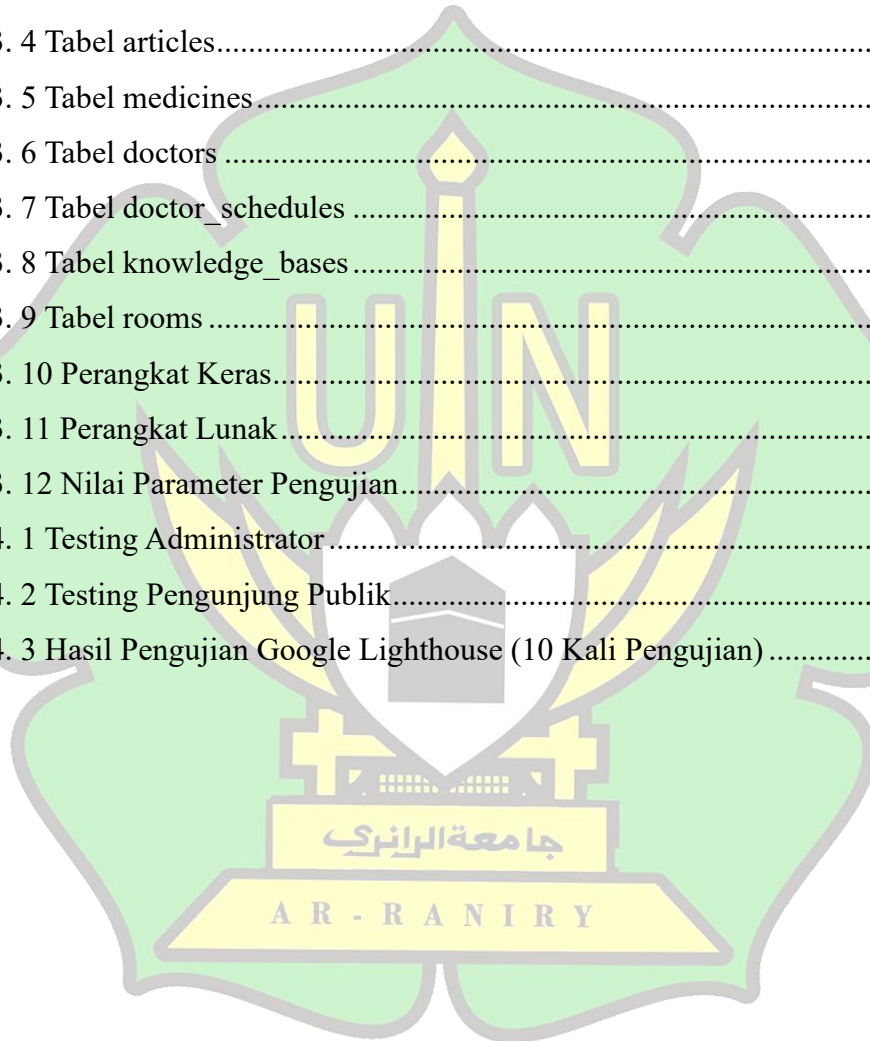
DAFTAR GAMBAR

Gambar 2. 1 Alur logika ChatBot	13
Gambar 2. 2. Arsitektur Modern Monolith	14
Gambar 3. 1. Skema Alur Penelitian.....	17
Gambar 3. 2. Use Case Diagram Sistem.....	26
Gambar 3. 3. Activity Diagram login Admin.....	27
Gambar 3. 4. Activity Diagram Mengelola Artikel.....	28
Gambar 3. 5 Activity Diagram Pengelola Obat	28
Gambar 3. 6 Activity Diagram Mengelola Knowledge base Chatbot.....	29
Gambar 3. 7 Activity Diagram Melihat Artikel dan Obat.....	29
Gambar 3. 8 Activity Diagram Mencari Artikel dan Obat	30
Gambar 3. 9 Activity Diagram Chatbot	31
Gambar 3. 10. UML diagram admin.....	32
Gambar 3. 11. UML diagram Pengguna	32
Gambar 3. 12 ERD Sistem Website RS Bhayangkara	34
Gambar 3. 13 Halaman Login.....	39
Gambar 3. 14 Halaman Register.....	39
Gambar 3. 15 Halaman Dashboard Admin	40
Gambar 3. 16 Halaman Manajemen Artikel	40
Gambar 3. 17 Halaman Manajemen Obat.....	40
Gambar 3. 18 Halaman Manajemen Knowledge Chatbot	41
Gambar 3. 19 Halaman Landing Page.....	41
Gambar 3. 20 Halaman Layanan Medis Unggulan.....	42
Gambar 3. 21 Halaman Artikel	42
Gambar 3. 22 Halaman Informasi & Jadwal Dokter.....	43
Gambar 4. 1 Route halaman utama (landing page) website	52
Gambar 4. 2 Route navigasi publik.....	53
Gambar 4. 3 Route Pengunjung	53
Gambar 4. 4 Route Admin	54
Gambar 4. 5 create tabel user.....	55
Gambar 4. 6 create tabel roles.....	55

Gambar 4. 7 create tabel articles	56
Gambar 4. 8 create tabel medicines	56
Gambar 4. 9 create tabel knowledge_base	57
Gambar 4. 10 create tabel doctors.....	57
Gambar 4. 11 create tabel doctor_schedules	57
Gambar 4. 12 create tabel rooms.....	58
Gambar 4. 13 Validasi Input.....	58
Gambar 4. 14 Inisialisasi API Client dan Pemrosesan Vektor Teks (Embedding)	59
Gambar 4. 15 Eksekusi Algoritma RAG dan Batas Kelayakan (Similarity Threshold)	59
Gambar 4. 16 Metode Native System Instruction Gemini API (Generasi Respon)	60
Gambar 4. 17 Kompilasi Data Struktural (Structured Data Context)	61
Gambar 4. 18 Perancangan Dokumen Instruksi.....	62
Gambar 4. 19 Perhitungan Geometri dan Pencarian Nilai Tertinggi	63
Gambar 4. 20 Rumus Matematis Klasifikasi (Cosine Similarity)	63
Gambar 4. 21 Validasi Data	64
Gambar 4. 22 Otentikasi dan Regenerasi Sesi	64
Gambar 4. 23 Validasi dan Pembuatan Akun Pasien	65
Gambar 4. 24 Login Otomatis.....	65
Gambar 4. 25 Fungsi Keluar Akun	66
Gambar 4. 26 Penentuan Berkas Tampilan Utama	66
Gambar 4. 27 Sinkronisasi Sesi Data Pengguna Terotentikasi.....	67
Gambar 4. 28 Optimalisasi Navigasi dan Sistem Caching Data Artikel.....	67
Gambar 4. 29 Halaman Beranda Website	68
Gambar 4. 30 Halaman Jadwal Dokter	69
Gambar 4. 31 Halaman Informasi Stok Obat.....	70
Gambar 4. 32 Halaman Berita & Informasi	71
Gambar 4. 33 Antarmuka Widget Chatbot	72
Gambar 4. 34 Halaman Dashboard Admin	73

DAFTAR TABEL

Tabel 2. 1 Tinjauan penelitian terdahulu	6
Tabel 2. 2 Perbandingan penelitian terdahulu	8
Tabel 3. 1 List Pertanyaan Pengunjung (FAQ)	18
Tabel 3. 2 Tabel roles	35
Tabel 3. 3 Tabel users	35
Tabel 3. 4 Tabel articles.....	35
Tabel 3. 5 Tabel medicines.....	36
Tabel 3. 6 Tabel doctors	36
Tabel 3. 7 Tabel doctor_schedules	36
Tabel 3. 8 Tabel knowledge_bases	36
Tabel 3. 9 Tabel rooms	37
Tabel 3. 10 Perangkat Keras.....	46
Tabel 3. 11 Perangkat Lunak.....	46
Tabel 3. 12 Nilai Parameter Pengujian.....	48
Tabel 4. 1 Testing Administrator.....	75
Tabel 4. 2 Testing Pengunjung Publik.....	76
Tabel 4. 3 Hasil Pengujian Google Lighthouse (10 Kali Pengujian)	78



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi saat ini menuntut institusi pelayanan publik, termasuk rumah sakit, untuk menyediakan akses informasi yang mudah, cepat, dan tidak terbatas oleh waktu. Media konvensional seperti brosur atau spanduk dinilai memiliki banyak keterbatasan dalam hal jangkauan dan mengharuskan masyarakat untuk datang langsung ke lokasi fisik guna memperoleh informasi. Oleh karena itu, *website* kini menjadi media informasi digital yang paling tepat dan akurat karena kemampuannya dalam menyampaikan informasi melalui teks, gambar, dan video yang saling mendukung satu sama lain guna memperluas jangkauan promosi secara efektif dan efisien.

Urgensi penerapan sistem informasi berbasis *website* di lingkungan rumah sakit telah banyak dibuktikan efektivitasnya oleh penelitian-penelitian terdahulu. Menurut penelitian yang dilakukan oleh Ardiansyah & Lunandi (2021) pada Rumah Sakit Graha Hermine Batam, perancangan *website* sangat krusial untuk mendiseminasikan informasi profil fasilitas medis serta jadwal praktik dokter secara praktis kepada masyarakat luas. Hal selaras juga diungkapkan dalam penelitian Taufan Zakaria (2023) yang menyatakan bahwa perancangan sistem informasi publik di rumah sakit terbukti esensial untuk merapikan manajemen data operasional dan memberikan transparansi layanan, sehingga secara langsung dapat meningkatkan citra instansi di mata masyarakat. Dalam hal kualitas teknis *website*, penelitian Daffa Al-farel & Rizal Dzikrillah (2025) membuktikan bahwa penggunaan *framework* Laravel mampu mengoptimalkan pengembangan aplikasi *web* yang terstruktur dengan performa yang sangat baik.

Meskipun berbagai penelitian terdahulu telah membuktikan tingginya manfaat implementasi *website* bagi rumah sakit, terdapat sebuah celah penelitian (*research gap*) mendasar yang masih perlu diselesaikan. Mayoritas sistem informasi yang telah dikembangkan umumnya memiliki sifat yang pasif dan berjalan satu arah. Dalam konteks medis, sifat pasif ini membuat pengguna dituntut

untuk selalu mencari informasi secara mandiri. Ketika masyarakat memiliki pertanyaan berulang (*Frequently Asked Questions*) yang mendesak, *website* konvensional belum mampu memberikan respons interaktif secara langsung, sehingga beban kerja staf administrasi tidak berkurang secara signifikan. Permasalahan serupa saat ini dihadapi oleh Rumah Sakit Bhayangkara Banda Aceh dalam mendiseminasikan informasi krusial seperti jadwal dokter dan ketersediaan obat. Keterbatasan akses informasi yang proaktif ini dapat menyebabkan citra institusi kurang dikenal oleh masyarakat luas.

Untuk menjembatani celah penelitian tersebut, penelitian ini mengusulkan perancangan *website* dengan menggunakan *landing page* medis yang terstandar, serta mengintegrasikannya dengan teknologi *chatbot* pada Rumah Sakit Bhayangkara Banda Aceh. Integrasi teknologi *chatbot* menjadi solusi krusial untuk mengatasi sifat pasif pada *website*. Merujuk pada Huberta & Wijaya (2023), *chatbot* berfungsi sebagai perangkat lunak simulasi percakapan yang membantu pelayanan informasi serta mampu mengurangi beban kerja staf administrasi melalui kemampuan menjawab pertanyaan berulang secara otomatis selama 24 jam penuh. Penerapan teknologi ini merupakan strategi efisiensi nyata yang menjamin pelayanan publik tetap responsif kapan saja tanpa terkendala batasan jam operasional manusia.

Secara teknis, penelitian ini mengusulkan perancangan sistem menggunakan arsitektur *Modern Monolith* dengan teknologi Laravel 12, Inertia.js, dan React.js. Penggunaan React.js dipilih karena kemampuannya menciptakan antarmuka yang responsif dan modular, yang sangat cocok untuk integrasi *chatbot* modern berbasis pengambilan data terstruktur (*structured data retrieval*). Guna memastikan *website* yang dibangun beroperasi secara optimal, kualitas sistem akan diaudit menggunakan alat *Google Lighthouse*. Menurut penelitian yang dilakukan oleh Setiawan & Yamasari (2025), *Google Lighthouse* merupakan instrumen evaluasi otomatis yang sangat efektif dan terstandar untuk mengukur performa teknis *website* pelayanan publik. Alat ini digunakan secara komprehensif untuk memastikan performa (*performance*), aksesibilitas (*accessibility*), keamanan (*best practices*), dan visibilitas pencarian (SEO) yang optimal, sehingga *website* Rumah

Sakit Bhayangkara dipastikan memenuhi standar pengembangan *web* modern yang inklusif.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana membangun Sistem Informasi layanan publik berbasis *web* yang responsif di RS Bhayangkara Banda Aceh dengan mengintegrasikan fitur interaksi digital guna memudahkan pasien berinteraksi dengan sistem ?
2. Bagaimana melakukan evaluasi terhadap kualitas Sistem Informasi layanan publik RS Bhayangkara Banda Aceh ?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah:

1. Membangun Sistem Informasi layanan publik berbasis *web* yang responsif di RS Bhayangkara Banda Aceh menggunakan Laravel 12 dengan menggunakan *library* Inertia.js dan React.js, serta dilengkapi fitur *chatbot* untuk memberikan kemudahan interaksi bagi pasien.
2. Melakukan evaluasi terhadap kualitas Sistem Informasi layanan publik RS Bhayangkara Banda Aceh.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat bagi berbagai pihak, antara lain:

1. **Bagi Rumah Sakit Bhayangkara Banda Aceh:**
 - a. Meningkatkan efisiensi layanan dengan mengurangi beban kerja dan biaya operasional staf administrasi atau *customer service*.
 - b. Menyediakan layanan informasi yang konsisten, selalu tersedia (24/7), dan dapat merespons kueri pengguna dengan lebih cepat.
 - c. Memperluas jangkauan pasar dan mengoptimalkan strategi pemasaran digital untuk layanan rumah sakit.

2. Bagi Pasien dan Masyarakat Umum:

- a. Memudahkan pengguna mendapatkan jawaban instan dan akurat terhadap pertanyaan umum (FAQ) tanpa terikat jam operasional.
- b. Membantu pasien dan keluarga dalam merencanakan kunjungan melalui akses *real-time* ke informasi jadwal dokter dan ketersediaan layanan.

3. Bagi Penulis:

Mendapatkan pengalaman praktis dalam menerapkan ilmu pengembangan sistem, khususnya dalam merancang (Laravel, Inertia.js, React.js) dan mengimplementasikan teknologi *chatbot* untuk studi kasus nyata.

1.5 Batasan Penelitian

Agar penelitian ini lebih terarah dan fokus, maka diberikan batasan-batasan sebagai berikut:

1. Sistem yang dikembangkan adalah *Website* Informasi Publik, dan tidak mencakup sistem internal rumah sakit yang kompleks seperti Rekam Medis Elektronik (EMR) atau Sistem Informasi Manajemen Rumah Sakit (SIMRS).
2. Fungsionalitas sistem terbatas pada tiga modul informasi utama: Berita/Pengumuman, Manajemen Jadwal Dokter, dan Informasi Ketersediaan Obat.
3. Batasan *Chatbot*: Asisten virtual (*chatbot*) yang dikembangkan berfokus pada penyediaan informasi layanan publik RS Bhayangkara Banda Aceh secara interaktif. Implementasi ini dibatasi pada penggunaan teknologi **Retrieval-Augmented Generation (RAG)**, di mana sistem melakukan pencarian konteks menggunakan algoritma *Cosine Similarity* berdasarkan representasi numerik (*Vector Embeddings*) dari *Knowledge Base* yang tersedia. Jawaban akhir disusun menggunakan model AI Generatif guna menghasilkan respon yang natural dan kontekstual, namun tetap terbatas

pada cakupan data referensi yang telah disediakan oleh administrator rumah sakit.

4. Fungsi *chatbot* terbatas pada penyampaian informasi dan tidak melayani pendaftaran pasien, *booking* online, atau memberikan konsultasi/diagnosis medis.
5. Lingkup implementasi sistem terbatas pada lingkungan studi kasus RS Bhayangkara Banda Aceh.



BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Bagian ini mengulas berbagai studi sebelumnya yang menjadi fondasi penelitian ini. Tinjauan ini dibagi menjadi dua fokus utama: (1) Analisis terhadap penelitian terdahulu yang relevan dengan pengembangan *website* dan implementasi *chatbot*, dan (2) Studi literatur komparatif untuk memetakan posisi dan kebaruan (*novelty*) dari penelitian yang diusulkan.

2.1.1 Penelitian Terdahulu

Tinjauan literatur menunjukkan adanya dua tren penelitian utama yang konvergen: (1) Optimalisasi media informasi digital berbasis *website* sebagai instrumen pemasaran dan layanan; dan (2) implementasi asisten virtual (*chatbot*) untuk otomasi layanan informasi. Temuan-temuan kunci dari penelitian yang relevan disajikan dalam Tabel 2.1.

Tabel 2. 1 Tinjauan penelitian terdahulu

Kategori	Peneliti (Tahun)	Judul / Topik Penelitian	Metode & Temuan Utama
Website RS	Salamudin & Meilantika, (2023)	Perancangan Website RS	Menggunakan metode <i>User Centered Design</i> (UCD).
		dr.Noesmir Baturaja	Penelitian ini menemukan bahwa website rumah sakit krusial untuk mendukung layanan 24 jam dan menjembatani komunikasi pasien tanpa batasan fisik, dengan antarmuka yang disesuaikan kebutuhan pengguna.

Website RS	Ardiansyah & Lunandi, (2021)	Analisa Website RS Graha Hermine Batam	Menggunakan metode <i>Waterfall</i> . Hasil menunjukkan bahwa peralihan dari media promosi konvensional (brosur) ke website dinamis meningkatkan trafik kunjungan dan efisiensi penyebaran informasi medis secara signifikan.
Chatbot	Huberta & Wijaya, (2023)	Perancangan Chatbot Website Program Studi Informatika Menggunakan Framework Codeigniter	Menggunakan <i>Framework CodeIgniter</i> . Chatbot terbukti efektif mengurangi beban kerja staf administrasi dengan menjawab pertanyaan berulang (FAQ) secara otomatis dan instan selama 24 jam penuh.
Chatbot	Dathrao et al., (2025)	<i>Intelligent ChatBot for College Website</i>	Menggunakan <i>React.js</i> dan <i>Structured Data</i> . Menekankan bahwa pendekatan pengambilan data terstruktur dari dokumen resmi lebih akurat dan ringan dibandingkan model NLP berat untuk institusi yang membutuhkan validitas data tinggi.
Evaluasi	Setiawan & Yamasari, (2025)	Evaluasi Performa Website SIKS-NG	Menggunakan <i>Google Lighthouse</i> . Penelitian ini merekomendasikan audit otomatis pada empat pilar: <i>Performance, Accessibility, Best Practices</i> , dan <i>SEO</i> untuk menjamin standar kualitas teknis website pelayanan publik.

Evaluasi	Daffa Al-farel & Rizal Dzikrillah, (2025)	Evaluasi Framework Pengembangan Web	Menggunakan parameter performa pada <i>Laravel</i> . Menyimpulkan bahwa arsitektur MVC pada <i>Laravel</i> memberikan keamanan, struktur kode yang rapi, dan kemudahan pemeliharaan (<i>maintainability</i>) yang tinggi untuk sistem berskala besar.
----------	---	-------------------------------------	---

2.1.2 Studi Literatur Komparatif

Untuk memperjelas posisi penelitian ini, dilakukan perbandingan terhadap beberapa studi terdahulu yang representatif. Perbandingan ini menyoroti perbedaan utama dalam tujuan, teknologi, dan fungsionalitas, sekaligus mengidentifikasi celah penelitian (*research gap*) yang akan diisi. Perbandingan penelitian dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Perbandingan penelitian terdahulu

No	Peneliti (Tahun)	Judul Penelitian	Metode / Teknologi	Hasil / Temuan	Perbedaan dengan Penelitian Ini
1	Salamudin & Meilantika, (2023)	Perancangan Website Rumah Sakit dr.Noesmir Baturaja	<i>User Centered Design</i> (UCD)	Terciptanya desain website yang ramah pengguna untuk mendukung layanan informasi 24 jam.	Penelitian ini fokus pada desain antarmuka (<i>interface</i>). Penelitian yang diusulkan melengkapi aspek desain dengan fungsionalitas cerdas (Chatbot) dan arsitektur

					<i>Modern Monolith.</i>
2	Ardiansyah & Lunandi, (2021)	Perancangan Dan Analisa Website RS Graha Hermine Batam	<i>Waterfall / PHP Native</i>	Website meningkatkan trafik kunjungan dan menggantikan media promosi cetak yang mahal.	Sistem sebelumnya masih menggunakan pembaruan konten manual. Penelitian yang diusulkan menggunakan Laravel 12 (CMS) yang lebih dinamis dan terintegrasi sistem <i>chat</i> .
3	Huberta & Wijaya, (2023)	Perancangan Chatbot Website Program Studi Informatika	<i>CodeIgniter / Rule-Based</i>	<i>Chatbot</i> berhasil mengurangi beban kerja admin dalam menjawab FAQ akademik.	Teknologi yang digunakan (<i>CodeIgniter</i>) bersifat tradisional. Penelitian yang diusulkan menggunakan React.js & Inertia untuk antarmuka <i>chat</i> yang responsif (SPA).
4	Dathrao et al., (2025)	<i>Intelligent ChatBot For Enhanced</i>	<i>React.js / Structured Data</i>	Metode <i>structured data retrieval</i>	Fokus domain penelitian ini adalah

		<i>College Website Interaction</i>		terbukti lebih akurat dan ringan dibanding NLP berat untuk data institusi.	Pendidikan (Kampus). Penelitian yang diusulkan mengadaptasi teknologi tersebut untuk domain Kesehatan (Jadwal Dokter & Obat).
5	Setiawan & Yamasari, (2025)	Evaluasi Performa Website SIKS-NG dengan Google Lighthouse	Audit <i>Google Lighthouse</i>	Menemukan bahwa evaluasi performa, SEO, dan aksesibilitas krusial untuk standar layanan publik.	Ini adalah penelitian evaluasi murni. Penelitian yang diusulkan adalah Rancang Bangun (Development) yang menggunakan standar evaluasi ini sebagai validasi akhir.
6	Daffa Al-farel & Rizal Dzikrillah, (2025)	Evaluasi Framework Pengembangan Web KinarBhusana	Audit Framework <i>Laravel</i>	Menyimpulkan Laravel memiliki keamanan dan struktur MVC terbaik untuk pengembangan web.	Fokus pada analisis <i>framework</i> . Penelitian yang diusulkan merupakan implementasi nyata Laravel 12 yang

					digabungkan dengan React.js (<i>Modern Monolith</i>).
7	Penelitian yang Diusulkan (2025)	Perancangan Website dengan Integrasi Chatbot untuk Optimalisasi Layanan Publik di RS Bhayangkara	Modern Monolith (Laravel 12, Inertia, React) + Lighthouse	Target: Website responsif dengan fitur <i>chatbot</i> medis otomatis yang teruji performanya.	(Posisi Penelitian) Mengintegrasikan seluruh keunggulan di atas: Arsitektur modern, Chatbot akurat berbasis data terstruktur, dan Evaluasi standar industri.

2.2 Landasan Konseptual

2.2.1 Website sebagai Media Informasi

Website merupakan kumpulan halaman digital yang saling terhubung dalam jaringan internet sebagai media penyampaian informasi multimedia. Salamudin dan Meilantika (2023) menegaskan bahwa, dalam penerapannya di institusi kesehatan, keberadaan website sangat krusial untuk mendukung operasional layanan yang berjalan selama 24 jam penuh. Instrumen vital ini berfungsi menjembatani komunikasi antara penyedia layanan dan masyarakat, sehingga memungkinkan pasien mengakses informasi tanpa batasan waktu.

Selain aspek ketersediaan layanan, efisiensi operasional dan promosi menjadi pertimbangan utama dalam pengembangan sistem berbasis web. Ardiansyah dan Lunandi (2021) memaparkan bahwa implementasi website dinamis mampu menggantikan metode konvensional, seperti penyebaran brosur fisik, yang sering memakan biaya tinggi. Peralihan ke platform digital ini

mampu meningkatkan trafik kunjungan secara signifikan serta memudahkan pasien mendapatkan rujukan informasi medis yang akurat.

Penelitian ini mengadopsi konsep *Landing Page* yang disesuaikan secara spesifik untuk kebutuhan sektor medis. Bugaeva dan Shustova (2025) menjelaskan bahwa, berbeda dari situs komersial biasa, *landing page* medis menuntut struktur ketat yang berfokus pada satu layanan spesifik. Elemen pemicu kepercayaan seperti profil dokter, sertifikasi, dan legalitas layanan ditonjolkan untuk menghilangkan keraguan pasien.

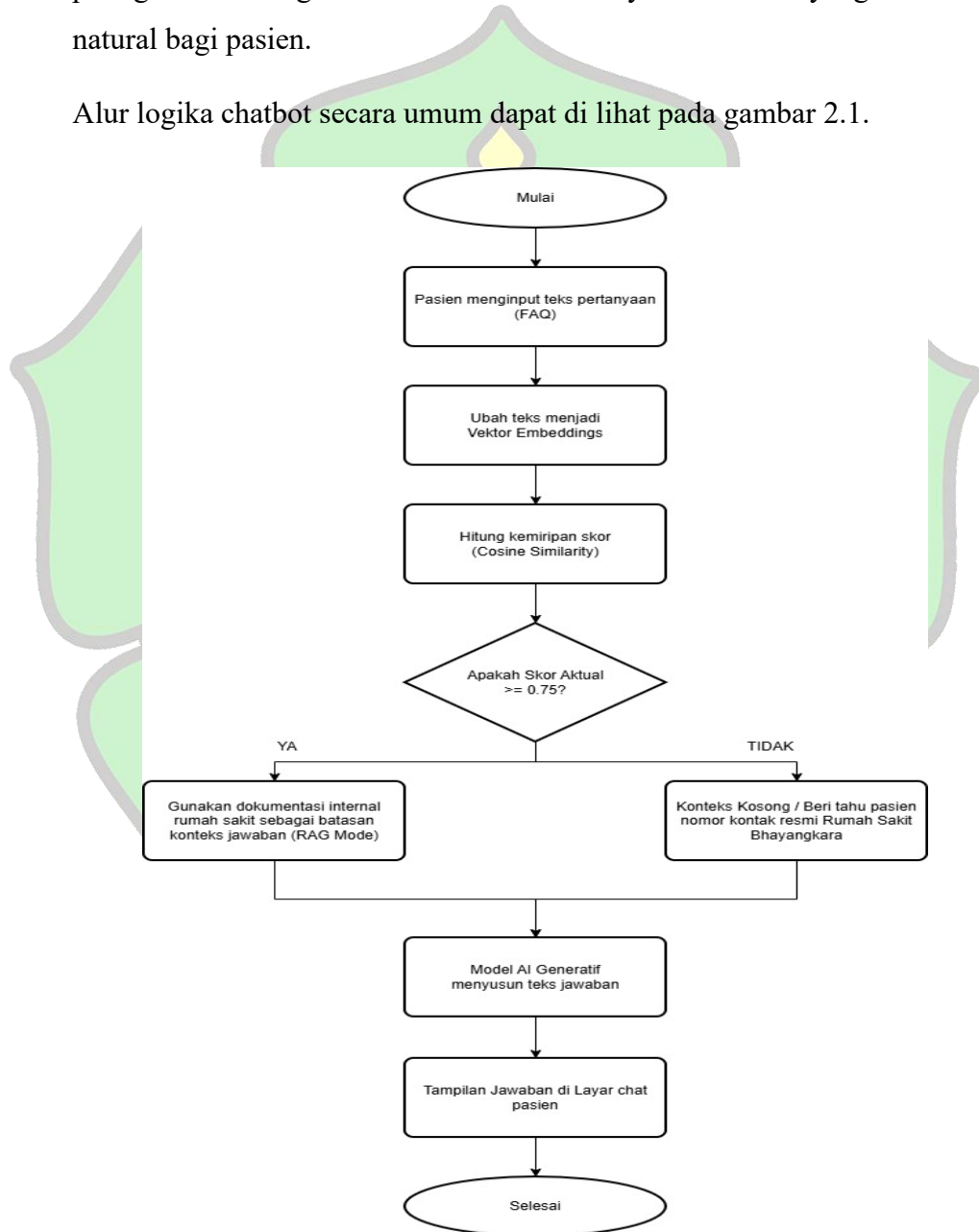
2.2.2 Chatbot sebagai Asisten Layanan

Chatbot merupakan perangkat lunak berbasis kecerdasan buatan yang mensimulasikan percakapan manusia untuk mengotomatisasi pelayanan publik. Huberta dan Wijaya (2023) menyatakan bahwa *chatbot* sangat efektif memangkas beban kerja administratif dengan merespons pertanyaan berulang (*FAQ*) secara instan selama 24 jam penuh. Namun, pada instansi kesehatan seperti RS Bhayangkara Banda Aceh, penggunaan model AI generatif murni berisiko memicu halusinasi data (informasi fiktif). Guna menjamin validitas informasi medis, sistem menerapkan tiga kombinasi teknologi utama:

1. ***Retrieval-Augmented Generation (RAG)*** RAG adalah arsitektur yang mengombinasikan pencarian dokumen terstruktur (*retrieval*) dengan kemampuan generasi bahasa alami AI (*generation*). RAG membatasi ruang lingkup jawaban AI agar hanya merujuk pada pangkalan data (*knowledge base*) resmi rumah sakit, sehingga risiko misinformasi pada data jadwal dokter dan stok obat dapat diminimalisasi.
2. ***Vector Embeddings (Representasi Vektor)*** *Vector embeddings* adalah metode *text mining* yang menerjemahkan kalimat manusia menjadi deretan angka (koordinat ruang multidimensi). Berbeda dengan pencarian kata kunci (*rule-based*) yang kaku, teknologi ini mampu memahami kedekatan makna dan hubungan semantik antar-kata secara mendalam. Seluruh data operasional rumah sakit dikonversi menjadi bentuk biner numerik ini sebelum dicocokkan dengan pertanyaan pasien.

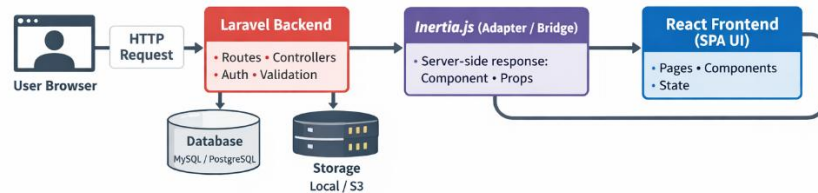
3. **Cosine Similarity (Algoritma Pencocokan)** *Cosine Similarity* menurut (Supiyanto & Sriyono, 2023) adalah metode geometris untuk menghitung tingkat kemiripan dua objek berbobot dengan mengukur nilai kosinus sudut antar-vektornya. Nilai kalkulasinya bergerak dari angka 0 (tidak mirip) hingga 1 (identik/sangat mirip). Melalui batas ambang kelayakan (*similarity threshold*) yang ketat, sistem akan mencabut dokumen *knowledge base* yang paling akurat sebagai landasan AI untuk menyusun balasan yang aman dan natural bagi pasien.

Alur logika chatbot secara umum dapat di lihat pada gambar 2.1.



Gambar 2. 1 Alur logika ChatBot

2.2.3 Arsitektur Modern Monolith (Laravel, Inertia.js, React.js)



Gambar 2. 2. Arsitektur Modern Monolith

Untuk membangun sistem yang terintegrasi ini, penelitian mengadopsi arsitektur *modern monolith*. Arsitektur ini menggabungkan kekuatan *backend* dan *frontend* dalam satu *codebase* namun tetap memberikan pengalaman pengguna yang dinamis.

1. **Laravel 12 (Backend):** Laravel adalah *framework* PHP modern yang menyediakan struktur *Model-View-Controller* (MVC) . Dalam sistem ini, Laravel akan berfungsi sebagai "otak" aplikasi, mengelola seluruh logika bisnis, interaksi *database*, autentikasi admin, dan menyediakan *Content Management System* (CMS) untuk mengelola data jadwal, obat, berita, dan *knowledge base chatbot* (Huberta & Wijaya, 2023).
2. **React.js (Frontend):** React adalah pustaka JavaScript untuk membangun antarmuka pengguna yang dinamis. Penggunaannya sejalan dengan tren penelitian *chatbot* modern yang mengutamakan antarmuka responsif. React akan digunakan untuk membangun semua komponen *frontend*, termasuk *widget chatbot* (Dathrao et al., 2025; Keertthi G & M, n.d.)

3. **Inertia.js (Jembatan/Bridge):** Inertia.js adalah "lem" yang menghubungkan *backend* Laravel dengan *frontend* React.js. Teknologi ini memungkinkan *controller* Laravel untuk berkomunikasi langsung dengan komponen React, sehingga menghasilkan pengalaman pengguna *Single Page Application* (SPA) yang cepat tanpa perlu membangun dan mengelola RESTful API yang terpisah.
4. **Google Lighthouse:** *Tool* ini akan digunakan sebagai alat bantu evaluasi untuk mengaudit *landing page* yang dihasilkan. Audit akan difokuskan pada aspek Performa, Aksesibilitas, dan *Search Engine Optimization* (SEO) untuk memastikan kualitas sistem.

2.3 RS Bhayangkara Banda Aceh

Rumah Sakit Bhayangkara Banda Aceh merupakan salah satu institusi pelayanan kesehatan yang berada dibawah naungan Kepolisian Negara Republik Indonesia (POLRI), terkhususnya POLDA Aceh. RS ini berada di Jalan Cut Nyak Dhien No. 23, Lamteumen Barat, Kecamatan Jaya Baru, Kota Banda Aceh.

RS Bhayangkara Banda Aceh ini mulai dibangun pada bulan Mei 2001 dan rampung pada Januari 2002. Selanjutnya RS ini mulai beroperasi pada tanggal 9 September 2002. Lalu pada tanggal 26 Desember 2004 rumah sakit ini juga terdampak bencana tsunami yang melanda Aceh dan mengalami kerusakan parah yang berakibat lumpuhnya pelayanan kesehatan total. Pada Februari 2005 IGD mulai beroperasi kembali dan pada Mei 2005 ruang operasi sudah bisa di lakukan Tindakan operasi untuk para pasien.

Awalnya, RS Bhayangkara Banda Aceh ditetapkan sebagai Rumah Sakit Tipe D pada tahun 2002. Pada 17 Maret 2011, rumah sakit ini meraih Akreditasi Penuh Tingkat Dasar melalui Sertifikasi Akreditasi Rumah Sakit Nomor: HK.03.05/III/763/11. Kemudian, pada 17 Januari 2018, rumah sakit ini dinyatakan Lulus Akreditasi Rumah Sakit Tingkat Perdana berdasarkan Keputusan Ketua KARS Nomor: KARSSERT/669/I/2018. Pada 22 Agustus 2024, klasifikasi rumah sakit ini meningkat dari tingkat IV menjadi tingkat III, menjadikannya salah satu rumah sakit rujukan terpercaya di Provinsi Aceh (Rumkit, 2023).

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam skripsi ini adalah penelitian pengembangan sistem (*Information System Development Research*). Secara umum, penelitian pengembangan sistem merupakan suatu proses ilmiah yang berfokus pada perancangan, pembuatan, pengujian, dan evaluasi suatu produk perangkat lunak berbasis komputer guna menyelesaikan permasalahan operasional tertentu atau memenuhi kebutuhan informasi dalam suatu organisasi. (Filza Maharani et al., 2024)

Sebagaimana dijelaskan dalam hasil kajian literatur oleh Filza Maharani et al., (2024), pengembangan sistem informasi bertujuan untuk mengelola, memproses, dan menyajikan data secara lebih efisien sehingga mampu mendukung pengambilan keputusan serta meningkatkan kualitas pelayanan publik. Dalam penelitian ini, jenis pengembangan sistem diarahkan untuk menghasilkan produk berupa platform website berbasis *framework* modern dan *asisten virtual* berteknologi kecerdasan buatan (*AI Chatbot*) yang dirancang khusus untuk mengoptimalkan alur informasi dan pelayanan informasi medis di Rumah Sakit Bhayangkara Banda Aceh.

3.2 Waktu dan Lokasi Penelitian

Penelitian ini dilakukan dimulai dari tanggal 05 Januari 2026 sampai dengan 24 Juli 2026 dalam membangun website responsif dan interaktif untuk instansi rumah sakit. Lokasi penelitian ini dilakukan pada Rumah Sakit Bhayangkara Banda Aceh yang beralamat pada Jl. Cut Nyak Dhien No.23, Lamtemen Bar., Kec. Jaya Baru, Kota Banda Aceh.

3.3 Teknik Pengumpulan Data

Pada penelitian ini peneliti menggunakan 3 teknik pengumpulan data yang dibutuhkan dalam perancangan website ini, sebagai berikut:

1. **Wawancara:** Merupakan teknik pengumpulan data atau informasi terkait yang dibutuhkan melalui pertanyaan-pertanyaan yang diajukan oleh peneliti terhadap pihak yang bersangkutan dengan penelitian ini. Data yang di peroleh dari wawancara ini merupakan data primer (Sanjaya, 2021).
2. **Observasi:** Merupakan Teknik pengumpulan data dengan pengamatan langsung pada lokasi dan meninjau kinerjanya secara langsung. Objek dari penelitian ini adalah Rumah Sakit Bhayangkara Banda Aceh yang menjadi tempat penelitian ini (Sanjaya, 2021).
3. **Studi Literatur:** Teknik ini merupakan Teknik pengumpulan data melalui berbagai teori yang berasal dari berbagai referensi penelitian, seperti jurnal penelitian, media-media cetak, internet, serta berbagai sumber lainnya yang berkaitan dengan penelitian ini untuk mendukung perancangan website.

3.4 Metode Pengembangan Sistem

Pada penelitian ini, penulis menggunakan metode pengembangan sistem yakni *Research and Development* (R&D) yang merupakan metode yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut. Berikut alur kerja dari metode penelitian R&D dapat dilihat pada Gambar 3.1.



Gambar 3. 1. Skema Alur Penelitian

Berikut penjelasan dari Gambar 3.1 skema alur penelitian:

1. Potensi dan Permasalahan

Pada tahap ini bertujuan untuk mengetahui permasalahan yang ada pada lapangan penelitian. Peneliti melakukan observasi langsung ke lapangan untuk melihat bagaimana alur sistem informasi di rumah sakit berjalan. Setelah observasi dilakukan, peneliti menemukan bahwa Rumah Sakit Bhayangkara Banda Aceh belum memiliki *website* resmi yang memberikan layanan informasi publik (seperti jadwal dokter dan stok obat) secara digital dan proaktif.

2. Pengumpulan Data

Pada tahap ini, peneliti melakukan sesi wawancara dan observasi lebih mendalam dengan pihak instansi rumah sakit, khususnya pada bagian IT dan administrasi. Tahap ini bertujuan untuk mengumpulkan seluruh data riil yang diperlukan untuk penelitian, termasuk data FAQ (*Frequently Asked Questions*) yang nantinya akan menjadi *knowledge base* untuk fitur *chatbot*. Berikut pertanyaan pengunjung (FAQ) yang di dapatkan dari hasil wawancara dengan pihak RS di sajikan dalam tabel 3.1.

Tabel 3. 1 List Pertanyaan Pengunjung (FAQ)

No	Kategori Layanan	Pertanyaan Pengunjung (FAQ)	Isi Konten Knowledge Base (Database Riil)
1	Lokasi & Kontak Resmi	Di mana alamat Rumah Sakit Bhayangkara Banda Aceh? Bagaimana cara menuju lokasi RS Bhayangkara?	Alamat Rumah Sakit Bhayangkara Banda Aceh berada di Jl. Cut Nyak Dhien No.23, Lamtemen Bar., Kec. Jaya Baru, Kota Banda Aceh, Aceh 23232.

2	Cakupan Pasien & Jaminan	Apakah RS Bhayangkara melayani pasien BPJS Kesehatan? Siapa saja kelompok masyarakat yang dapat berobat di RS Bhayangkara?	Rumah Sakit Bhayangkara ini melayani pengguna atau pasien yang menggunakan bpjs. RS Bhayangkara melayani masyarakat umum, peserta BPJS Kesehatan, serta anggota Polri/TNI beserta keluarganya.
3	Alur Pendaftaran BPJS	Bagaimana alur dan syarat pendaftaran rawat jalan memakai BPJS? Berkas apa saja yang harus dibawa saat mendaftar ke poliklinik?	Alur pendaftaran menggunakan bpjs di RS Bhayangkara Banda Aceh: 1. Datang ke bagian Pendaftaran Poli dengan membawa KTP, Kartu BPJS Kesehatan, dan Surat Rujukan dari Faskes Tingkat 1 (Puskesmas/Klinik/Dr. Keluarga). 2. Ambil nomor antrean pendaftaran. 3. Setelah berkas diverifikasi, Anda akan diarahkan menuju Poliklinik Spesialis tujuan.
4	Layanan Gawat	Apakah pasien darurat harus membawa surat	Untuk kondisi gawat darurat (kategori Triage Merah/Kuning), pasien dapat langsung ditangani di IGD 24

	Darurat (IGD)	rujukan faskes 1 ke IGD? Bagaimana prosedur pelayanan pasien BPJS di IGD 24 Jam?	jam tanpa rujukan Faskes 1. Verifikasi BPJS dapat dilakukan setelah pasien ditangani.
5	Tata Tertib Jam Besuk	Kapan jadwal jam besuk pasien rawat inap? Jam berapa keluarga diperbolehkan menjenguk pasien?	Waktu kunjungan (jam besuk) pasien rawat inap: • Siang: Pukul 11.00 – 14.00 WIB • Sore/Malam: Pukul 17.00 – 21.00 WIB Pengunjung diimbau menjaga ketenangan dan kebersihan lingkungan rumah sakit.
6	Fasilitas Kamar Rawat Inap	Kelas kamar rawat inap apa saja yang tersedia? Berapa kapasitas tempat tidur pada masing-masing tipe ruangan?	RS Bhayangkara Banda Aceh menyediakan beberapa pilihan kelas kamar rawat inap: • VVIP / VIP: 1 tempat tidur per ruangan, fasilitas lengkap, privasi maksimal. • Kelas 1: 2 tempat tidur per ruangan. • Kelas 2: 3–4 tempat tidur per ruangan.

			• Kelas 3: 5–6 tempat tidur per ruangan (tersedia untuk peserta BPJS).
7	Layanan Medis & Fasilitas Penunjang	Layanan medis apa saja yang disediakan oleh rumah sakit? Fasilitas penunjang apa saja yang ada di RS Bhayangkara?	RS Bhayangkara Banda Aceh menyediakan berbagai layanan kesehatan komprehensif, antara lain: 1. IGD 24 Jam (Pelayanan gawat darurat non-stop) 2. Poliklinik Spesialis (Penyakit Dalam, Bedah, Anak, Kebidanan & Kandungan/OBGYN, Saraf, THT, Mata, Kulit, Jiwa, dll.) 3. Instalasi Rawat Inap (VIP, Kelas 1, 2, dan 3) 4. Instalasi Farmasi / Apotek 24 Jam 5. Laboratorium Klinik 6. Radiologi & USG 7. Fisioterapi & Rehabilitasi Medik 8. Layanan Kedokteran Kepolisian (Dokkes): Medical Check Up (MCU), Visum et Repertum, dll.

8	Pemeriksaan Bebas Narkoba (SKBN)	Bagaimana syarat dan prosedur pembuatan Surat Bebas Narkoba (SKBN)? Apakah RS Bhayangkara melayani tes urine narkoba untuk keperluan kerja/sekolah?	Prosedur pembuatan Surat Keterangan Bebas Narkoba (SKBN) di RS Bhayangkara: 1. Membawa KTP/identitas diri asli dan pas foto 3x4. 2. Mendaftar di bagian administrasi/laboratorium untuk pemeriksaan narkoba. 3. Mengambil sampel urine di laboratorium yang diawasi petugas. 4. Pemeriksaan menggunakan parameter tes narkoba standar (Amphetamine, THC, Morphine, dll.). 5. Hasil dan surat keterangan dapat diambil setelah proses verifikasi dokter penanggung jawab.
9	Pemeriksaan Visum et Repertum	Bagaimana alur dan syarat pemeriksaan visum bagi korban kekerasan/laka lantas? Siapa yang berhak mengajukan	Prosedur permohonan Visum et Repertum di RS Bhayangkara Banda Aceh: 1. Pembuatan visum harus disertai Surat Permintaan Visum (SPV) resmi dari pihak kepolisian/penyidik. 2. Korban didampingi oleh petugas kepolisian atau

		pembuatan visum medis?	keluarga datang ke IGD/instalasi forensik. 3. Dokter pemeriksa melakukan pemeriksaan fisik/medis dan mendokumentasikan luka atau temuan klinis. 4. Hasil visum et repertum diserahkan secara resmi dan rahasia kepada pihak penyidik yang berwenang.
--	--	------------------------	--

3. Desain Produk

Setelah data-data yang dibutuhkan terkumpul, peneliti mulai membuat rancangan awal aplikasi. Berikut perancangan teknis pada tahap ini:

- **Perancangan ERD (Entity Relationship Diagram) & Basis Data:** Peneliti merancang struktur *database* yang akan digunakan sistem (seperti tabel dokter, poli, obat, dan *chatbot*) beserta relasi antar datanya menggunakan ERD.
- **Desain UI (User Interface):** Peneliti membuat *wireframe* atau *mockup* antarmuka halaman *website* (seperti halaman *landing page*, *dashboard* admin, dan tampilan *widget chatbot*) agar gambaran visual sistem terlihat jelas.
- **Pemodelan Sistem:** Menggunakan *Unified Modeling Language* (UML) seperti *Use Case* dan *Activity Diagram* untuk memetakan alur kerja aktor (Admin dan *User*).

4. Validasi Sistem

Rancangan sistem (*desain UI*, *ERD*, dan *UML*) yang telah dibuat kemudian divalidasi oleh ahli dan calon pengguna, yaitu dosen pembimbing, staf IT, serta petugas rumah sakit. Hal ini dilakukan untuk memastikan bahwa desain antarmuka dan alur *database* sudah sesuai dengan kebutuhan fungsional rumah sakit.

5. Revisi Desain & Implementasi Kode

Berdasarkan masukan dari tahap validasi, peneliti melakukan perbaikan terhadap desain UI, alur *input* data, dan struktur rancangan basis data agar lebih optimal.

- **Penerapan Arsitektur Modern Monolith:** Tepat setelah desain final disetujui, tahap penulisan kode (*coding*) dimulai. Pada titik inilah sistem mulai dibangun menggunakan **Arsitektur Modern Monolith**, di mana *backend* dikembangkan menggunakan **Laravel 12**, sementara *frontend* (UI yang sudah didesain) dirender menggunakan **React.js** yang dijembatani oleh **Inertia.js** untuk menghasilkan antarmuka yang sangat responsif.

6. Uji Coba Produk

Setelah sistem selesai dibangun menjadi sebuah aplikasi web yang utuh, dilakukan pengujian teknis awal secara internal. Pengujian pada tahap ini menggunakan metode **Black Box Testing** untuk memastikan setiap fungsi sistem (seperti form *login*, fitur pencarian jadwal, operasi CRUD admin, dan respons *chatbot*) berjalan dengan baik dan tidak ada fungsi yang gagal (*error*).

7. Revisi Produk

Berdasarkan hasil temuan dari *Black Box Testing*, peneliti melakukan *debugging* atau perbaikan terhadap *bug* dan kesalahan logika program (*error code*) yang ditemukan selama masa pengujian produk berlangsung.

8. Uji Coba Pemakaian & Evaluasi Kualitas

Sistem yang sudah dirancang, diperbaiki, dan berjalan stabil kemudian diuji kembali secara menyeluruh pada lingkungan *localhost*.

- **Evaluasi Google Lighthouse:** Pada tahap akhir inilah evaluasi kinerja menggunakan *Google Lighthouse* masuk. *Website* yang sudah berjalan di *localhost* diaudit secara komprehensif untuk memastikan bahwa sistem beroperasi secara optimal pada 4 metrik utama: **Performa** (kecepatan *loading* kerangka *React.js*), **Aksesibilitas** (kemudahan UI dibaca oleh semua kalangan), **Best**

Practices (standar keamanan dan penulisan kode), serta **SEO** (kemudahan visibilitas sistem di mesin pencari).

Melalui tahapan ini, penelitian tidak hanya menghasilkan produk teknologi, tetapi juga memastikan bahwa produk tersebut valid, efektif, dan sesuai dengan kebutuhan nyata pengguna di RS Bhayangkara Banda Aceh.

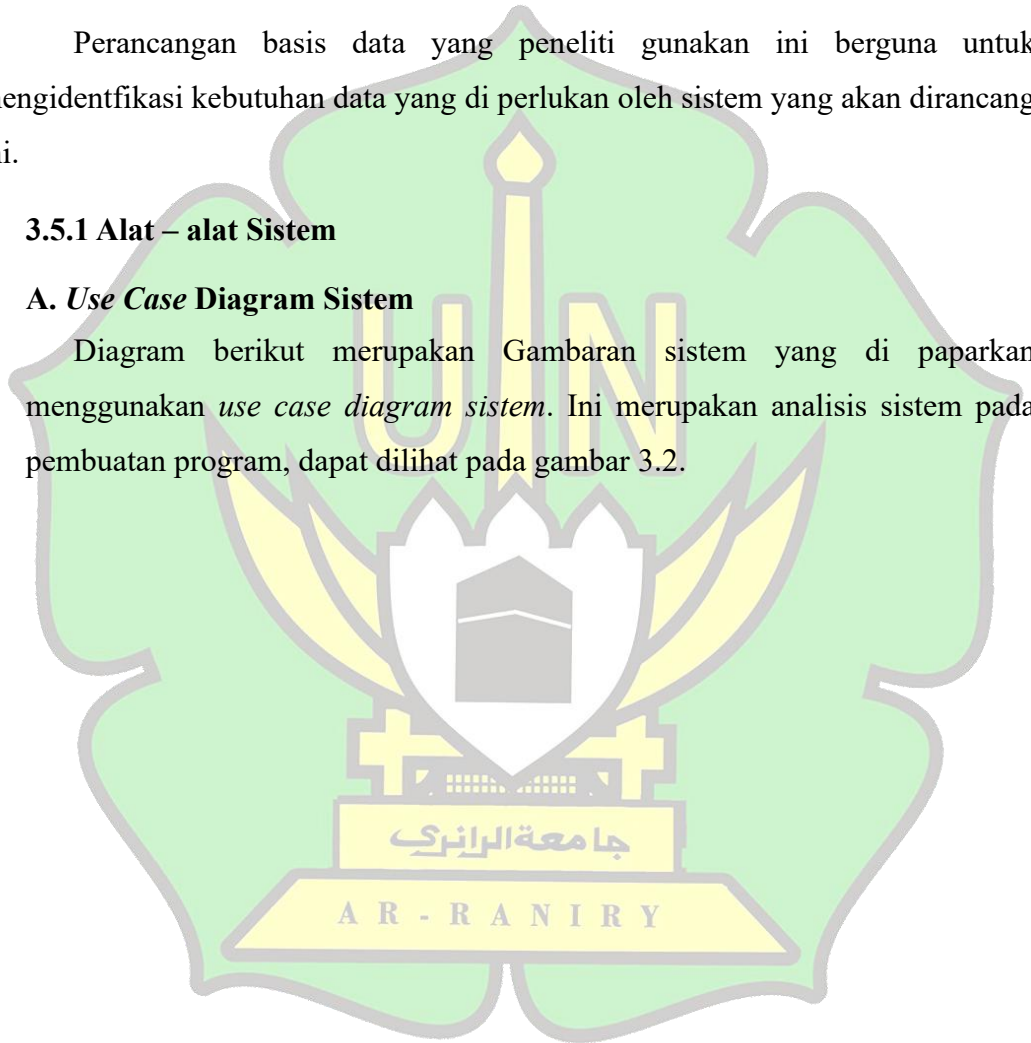
3.5 Perancangan Sistem

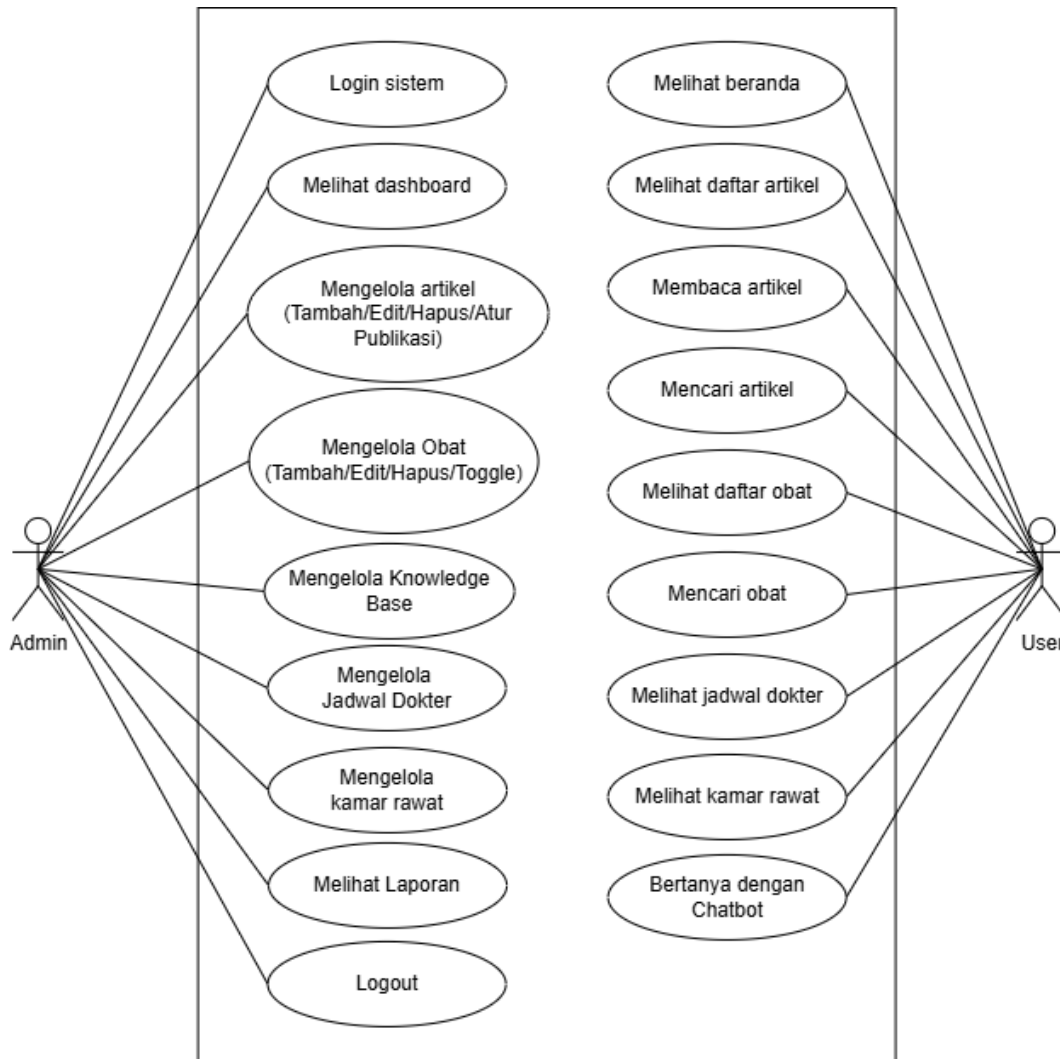
Perancangan basis data yang peneliti gunakan ini berguna untuk mengidentifikasi kebutuhan data yang diperlukan oleh sistem yang akan dirancang ini.

3.5.1 Alat – alat Sistem

A. *Use Case Diagram* Sistem

Diagram berikut merupakan Gambaran sistem yang dipaparkan menggunakan *use case diagram sistem*. Ini merupakan analisis sistem pada pembuatan program, dapat dilihat pada gambar 3.2.





Gambar 3. 2. Use Case Diagram Sistem

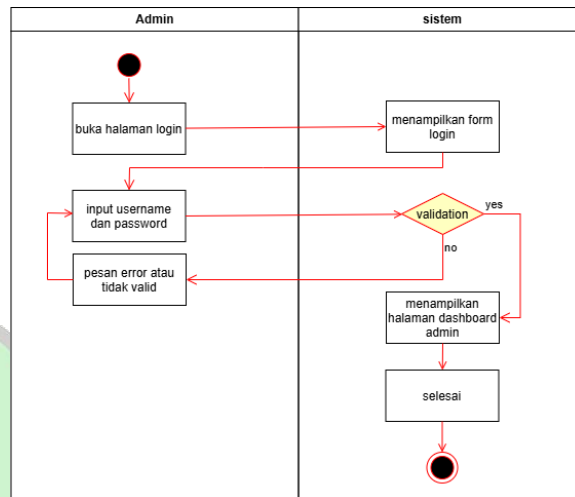
B. Activity Diagram

Activity Diagram merupakan sebuah representasi visual yang digunakan untuk menggambarkan alur kerja atau urutan aktivitas dalam suatu sistem maupun proses bisnis. Perlu ditekankan bahwa diagram ini berfokus pada aktivitas-aktivitas yang dilaksanakan oleh sistem itu sendiri, bukan oleh aktor yang berinteraksi dengan sistem. Oleh karena itu, inti dari *Activity Diagram* adalah memvisualisasikan rangkaian aktivitas yang dapat dijalankan oleh sistem dalam konteks proses yang relevan.

a. Activity Diagram Login Admin

Diagram ini menjelaskan bagaimana pengguna atau admin melakukan proses login. Pengguna membuka halaman login, mengisi email dan

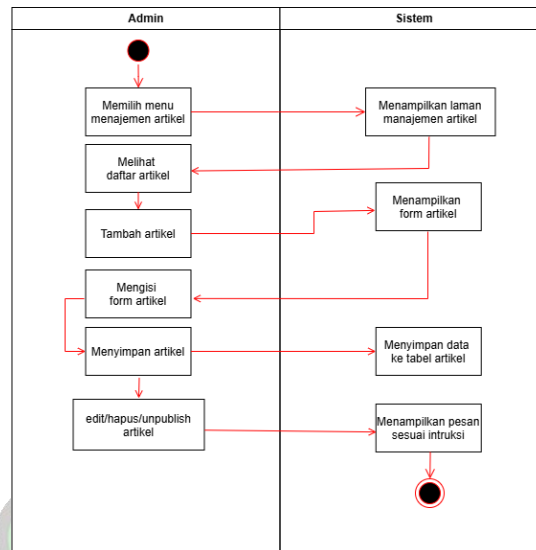
password, lalu sistem memvalidasi input tersebut. Jika data benar, sistem memberikan akses sesuai role; jika salah, sistem menampilkan pesan error. Pada Gambar 3.3 menunjukan alur yang menggambarkan proses autentikasi dasar dalam sistem.



Gambar 3. 3. Activity Diagram login Admin

b. *Activity Diagram Mengelola Artikel*

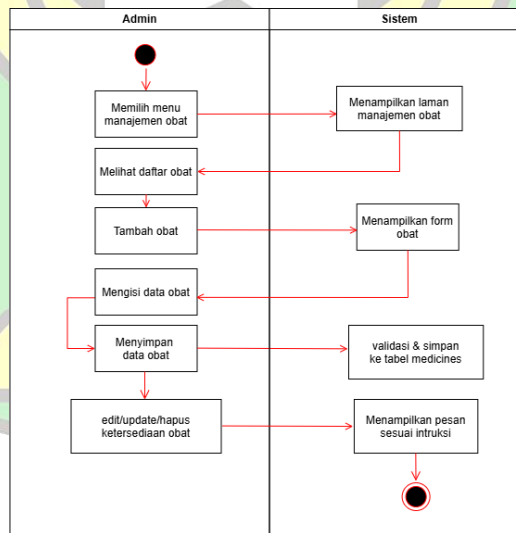
Diagram ini menjelaskan bagaimana admin mengelola data artikel. Admin dapat menambah artikel baru, mengedit artikel, menghapus artikel, atau mengubah status publikasi. Sistem memvalidasi data lalu memperbarui informasi di tabel artikel. Pada Gambar 3.4 menunjukan alur yang mencerminkan proses kerja admin pada modul manajemen artikel.



Gambar 3. 4. Activity Digram Mengelola Artikel

c. *Activity Diagram Mengelola Obat*

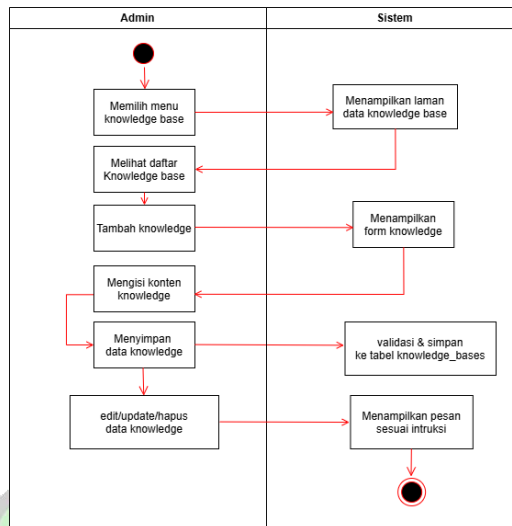
Admin membuka menu manajemen obat dan dapat melakukan penambahan, pengeditan, penghapusan, atau mengubah status ketersediaan obat. Sistem melakukan validasi dan menyimpan perubahan ke database. Gambar 3.5 ini menunjukkan proses administrasi data obat yang dilakukan oleh admin.



Gambar 3. 5 Activity Diagram Pengelola Obat

d. *Activity Diagram Mengelola Knowledge Base Chatbot*

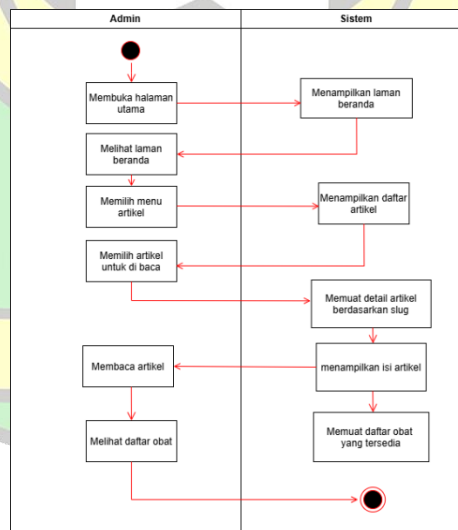
Admin mengelola data pengetahuan yang digunakan chatbot. Admin dapat menambahkan entri baru atau menghapus entri lama. Sistem menyimpan perubahan pada tabel *knowledge_bases*. Gambar 3.6 ini menggambarkan proses pemeliharaan data pendukung fitur chatbot.



Gambar 3. 6 Activity Diagram Mengelola Knowledge base Chatbot

e. *Activity Diagram Melihat Artikel dan Obat*

Diagram ini menunjukkan proses ketika pengguna membuka halaman utama, memilih menu Artikel atau Obat, dan melihat konten yang tersedia. Sistem menampilkan daftar artikel yang sudah dipublikasikan atau daftar obat yang tersedia. Pengguna dapat memilih salah satu artikel untuk melihat detailnya. Gambar 3.7 menunjukkan alur yang menggambarkan navigasi informasi oleh pengguna.

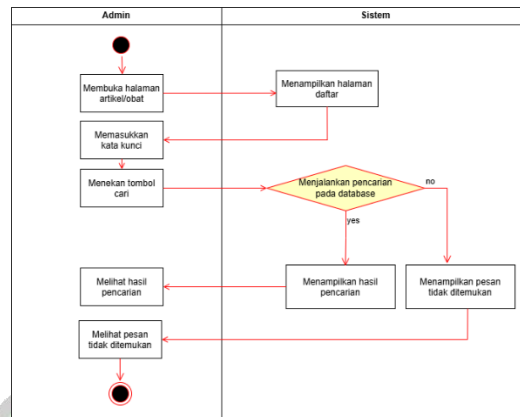


Gambar 3. 7 Activity Diagram Melihat Artikel dan Obat

f. *Activity Diagram Mencari Artikel dan Obat*

Proses pencarian dimulai ketika pengguna memasukkan kata kunci. Sistem memproses kata kunci tersebut, mencari kecocokan di database, lalu menampilkan hasilnya. Jika tidak ditemukan, sistem menampilkan pesan

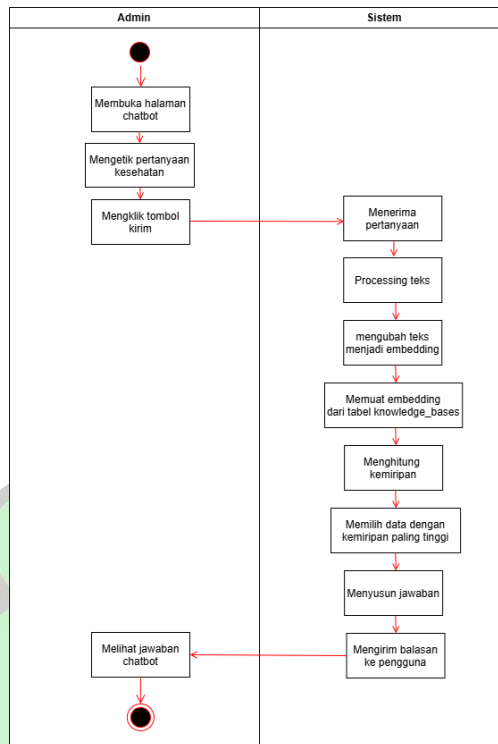
“data tidak ditemukan”. Gambar 3.8 ini menunjukkan interaksi pengguna dengan fitur pencarian terintegrasi.



Gambar 3. 8 Activity Diagram Mencari Artikel dan Obat

g. *Activity Diagram Chatbot*

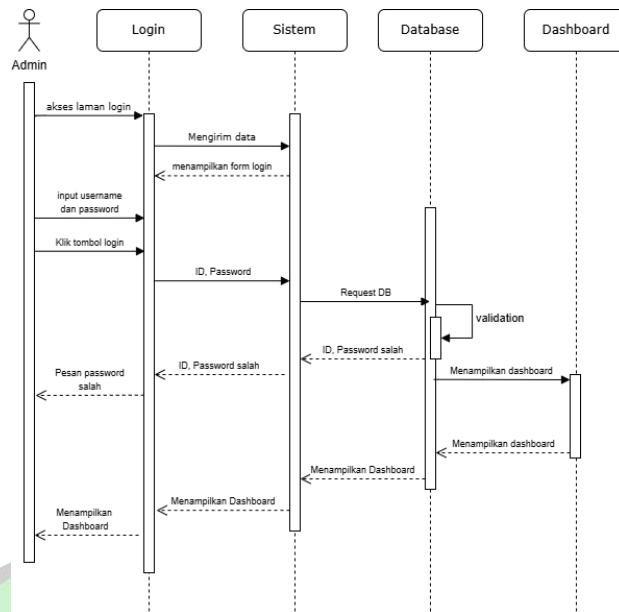
Pengguna mengetik pertanyaan lalu mengirimkannya. Sistem mengolah pertanyaan dengan mengubahnya menjadi embedding, mencocokkan dengan *knowledge base*, dan mengirimkan jawaban yang paling sesuai. Pengguna dapat melanjutkan percakapan atau mengakhirinya. Gambar 3.9 ini menggambarkan proses komunikasi interaktif antara pengguna dan chatbot cerdas.



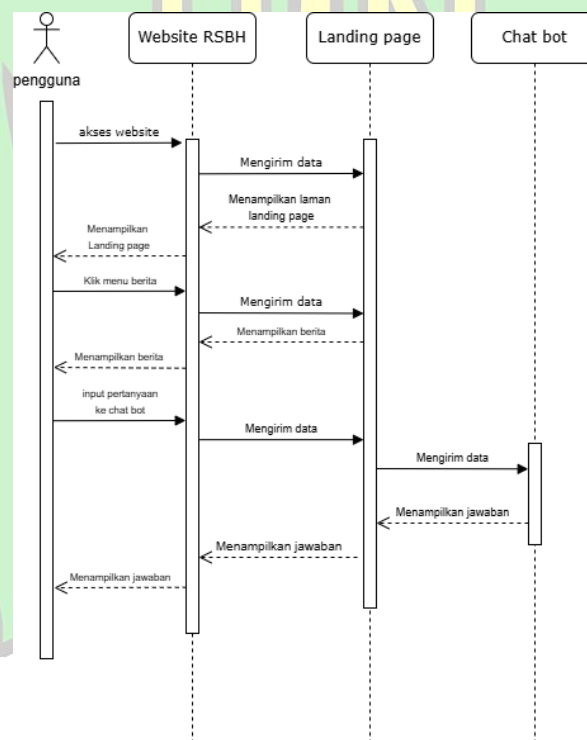
Gambar 3. 9 Activity Diagram Chatbot

C. Sequence Diagram

Sequence Diagram adalah suatu jenis diagram yang menampilkan bagaimana interaksi antara objek pada sistem atau sering disebut dengan diagram UML. Diagram ini menggambarkan bagaimana alur atau interaksi antar objek bekerja dalam suatu sistem dalam mengirimkan pesan. Berikut diagram UML pada sistem website RS Bhayangkara Aceh yang di tunjukkan pada Gambar 3.10 dan Gambar 3.11.



Gambar 3. 10. UML diagram admin

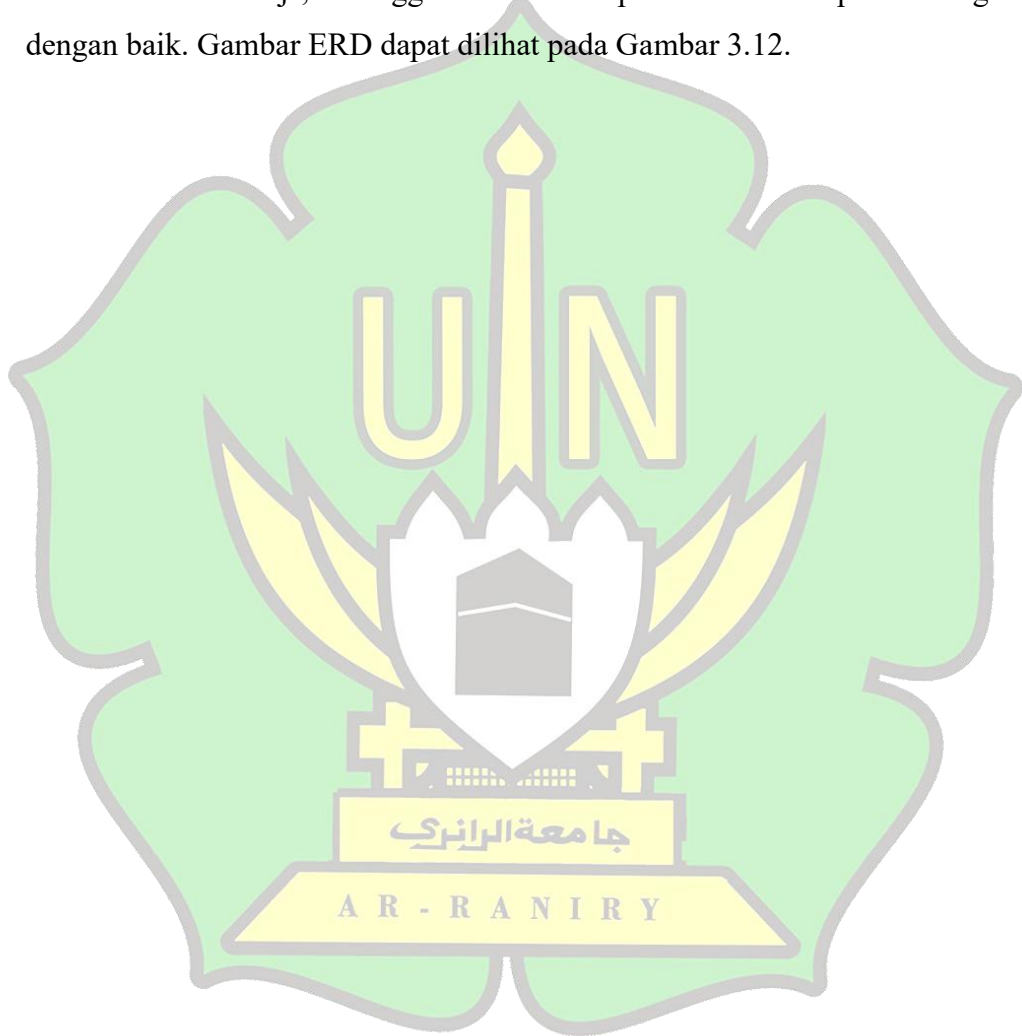


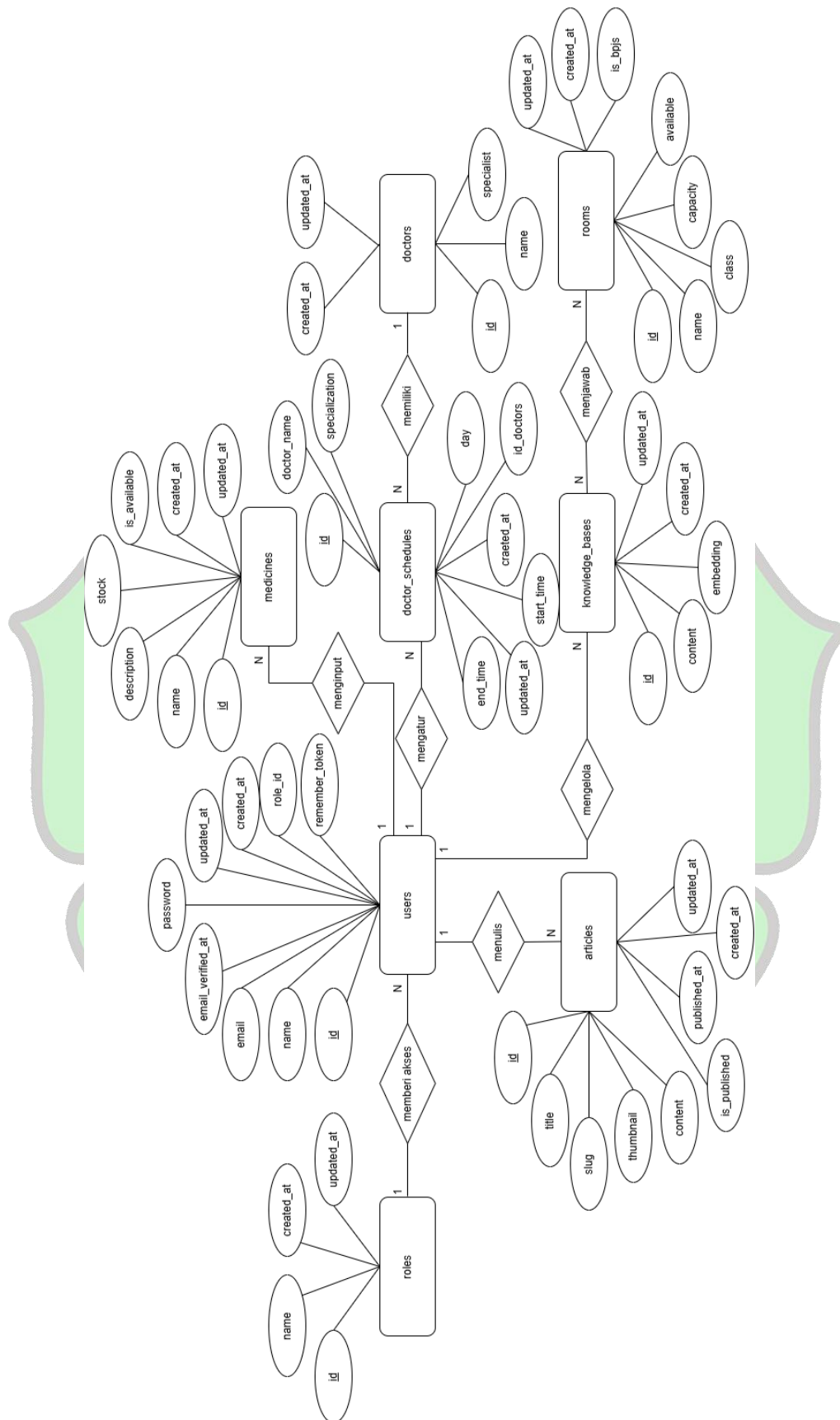
Gambar 3. 11. UML diagram Pengguna

3.5.2 Perancangan Basis Data

A. ERD (*Entity Realtionship Diagram*)

Entity Relationship Diagram (ERD) adalah model konseptual yang digunakan untuk menggambarkan hubungan antar data dalam suatu sistem informasi secara logis dan terstruktur. ERD menjadi dasar penting dalam perancangan database pada sistem informasi manajemen rumah sakit berbasis Laravel dan React.js, sehingga seluruh komponen sistem dapat terintegrasi dengan baik. Gambar ERD dapat dilihat pada Gambar 3.12.





Gambar 3. 12 ERD Website RS Bhayangkara

B. Metadata Database

Metadata *database* merupakan data yang menjelaskan tentang data-data yang terdapat dan tersimpan dalam database sistem.

Tabel 3. 2 Tabel roles

Nama Field	Tipe Data	Keterangan
id	BIGINT, PK	Primary key
name	VARCHAR(50)	Nama peran (admin, user, dll)
created_at	TIMESTAMP, NULL	Waktu dibuat
updated_at	TIMESTAMP, NULL	Waktu diperbarui

Tabel 3. 3 Tabel users

Nama Field	Tipe Data	Keterangan
id	BIGINT, PK	Primary key
name	VARCHAR(100)	Nama pengguna
email	VARCHAR(100), UNIQUE	Email untuk login
email_verified_at	TIMESTAMP, NULL	Waktu verifikasi email
password	VARCHAR(255)	Password terenkripsi (hashed)
role_id	BIGINT, FK, NULL	Relasi ke <code>roles.id</code>
remember_token	VARCHAR(100), NULL	Token remember me
created_at	TIMESTAMP, NULL	
updated_at	TIMESTAMP, NULL	

Tabel 3. 4 Tabel articles

Nama Field	Tipe Data	Keterangan
id	BIGINT, PK	Primary key
title	VARCHAR(200)	Judul artikel
slug	VARCHAR(200), UNIQUE	Slug URL
thumbnail	VARCHAR(255), NULL	Path gambar thumbnail
content	LONGTEXT	Isi artikel
is_published	BOOLEAN	Status publikasi (1 = tampil, 0 = draft)
published_at	TIMESTAMP, NULL	Tanggal publikasi
created_at	TIMESTAMP, NULL	
updated_at	TIMESTAMP, NULL	

Tabel 3. 5 Tabel medicines

Nama Field	Tipe Data	Keterangan
id	BIGINT, PK	Primary key
name	VARCHAR(150)	Nama obat
description	TEXT, NULL	Deskripsi / kegunaan obat
stock	INT	Jumlah stok
is_available	BOOLEAN	Status tersedia (1) / tidak tersedia (0)
created_at	TIMESTAMP, NULL	
updated_at	TIMESTAMP, NULL	

Tabel 3. 6 Tabel doctors

Nama Field	Tipe Data	Keterangan
id	INTEGER, PK	Primary key (autoincrement)
name	VARCHAR	Nama lengkap dokter beserta gelar
specialist	VARCHAR	Bidang keahlian spesialisasi kedokteran
created_at	DATETIME	Waktu data ditambahkan
updated_at	DATETIME	Waktu data diperbarui

Tabel 3. 7 Tabel doctor_schedules

Nama Field	Tipe Data	Keterangan
id	INTEGER, PK	Primary key (autoincrement)
doctor_id	INTEGER, FK	Foreign key, berelasi dengan tabel doctors(id)
day	VARCHAR	Hari pelayanan praktik (Senin s.d. Minggu)
time_start	TIME	Jam mulai dimulainya pelayanan medis
time_end	TIME	Jam berakhirnya waktu pelayanan dinas
is_available	TINYINT(1)	Status kehadiran dinas (default '1' = Hadir/Aktif)
created_at	DATETIME	Waktu jadwal ditambahkan
updated_at	DATETIME	Waktu jadwal diubah

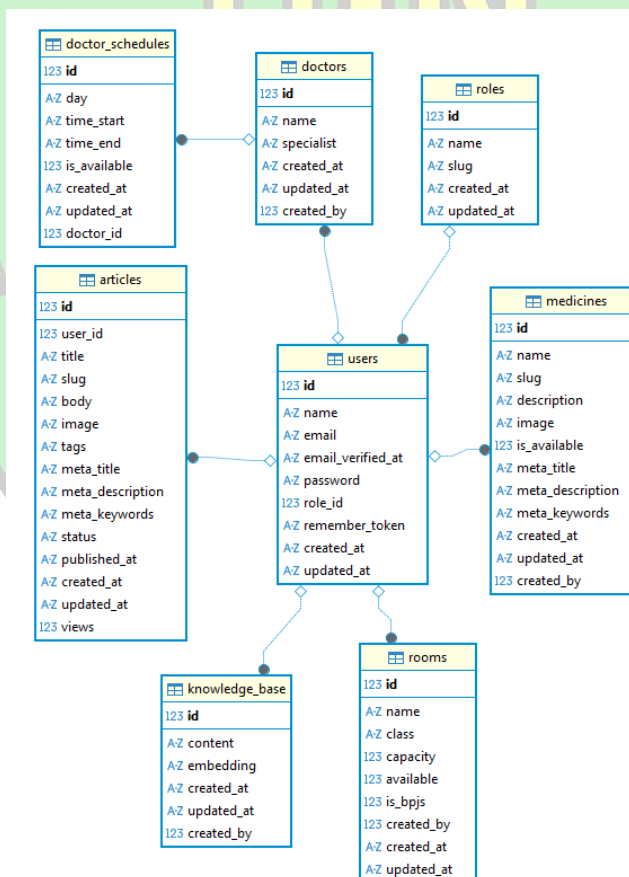
Tabel 3. 8 Tabel knowledge_bases

Nama Field	Tipe Data	Keterangan
id	INTEGER, PK	Primary key (autoincrement)
content	TEXT	Teks dokumen narasi panduan pelayanan/medis RS
embedding	TEXT	Larik (<i>array</i>) koordinat angka biner vektor semantik
created_at	DATETIME	Waktu data ditambahkan
updated_at	DATETIME	Waktu data diperbarui

Tabel 3. 9 Tabel rooms

Nama Field	Tipe Data	Keterangan
id	INTEGER, PK	Primary key (autoincrement)
name	VARCHAR	Nama ruangan rawat inap
class	VARCHAR	Kelas fasilitas ruangan (misal: VIP, Kelas 1, dll)
capacity	INTEGER	Total kapasitas kasur dalam kamar (default '0')
available	INTEGER	Sisa jumlah kasur kosong yang tersedia (default '0')
is_bpjs	TINYINT(1)	Indikator dukungan jaminan (1 = BPJS, 0 = Umum)
created_at	DATETIME	Waktu data ruangan ditambahkan
updated_at	DATETIME	Waktu data ruangan diperbarui

Pada Gambar 3.13 ini merupakan relasi tabel-tabel dalam database yang sudah di paparkan diatas dengan tabel user menjadi pusat kendali operasi sistem.



Gambar 3. 13 Relasi antar tabel database

3.6 Perancangan *Interfaces Sistem*

Pada penelitian ini, antarmuka dibangun menggunakan kombinasi teknologi Laravel, React.js, dan Inertia.js, yang menghasilkan tampilan berbasis *Single Page Application* (SPA) dengan karakteristik cepat, responsif, dan interaktif.

Secara keseluruhan, antarmuka pada sistem ini terbagi menjadi dua kelompok utama, yaitu antarmuka untuk pengguna umum dan antarmuka untuk administrator (*dashboard admin*).

3.6.1 Antar Muka Administrator

a. Halaman Login dan Register

Halaman Login pada *Gambar 3.13* berfungsi untuk autentikasi sebelum masuk ke dashboard admin. User memasukkan email dan kata sandi, kemudian diarahkan ke halaman pengelolaan konten jika berhasil masuk. Halaman ini memastikan hanya admin yang berhak dapat mengakses fitur manajemen.

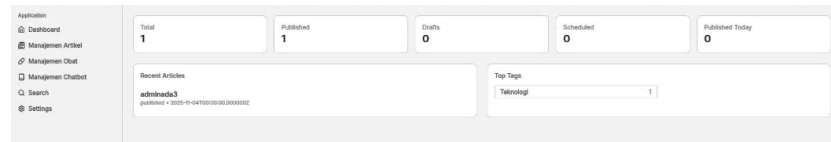
Halaman Register pada *Gambar 3.14* digunakan untuk membuat akun admin baru. Form ini berisi input nama, email, kata sandi, dan konfirmasi kata sandi. Walaupun tampil untuk umum, fitur ini sebenarnya diperuntukkan bagi administrator yang akan mengelola sistem melalui dashboard.

Gambar 3. 14 Halaman Login

Gambar 3. 15 Halaman Register

b. Halaman Dashboard Admin

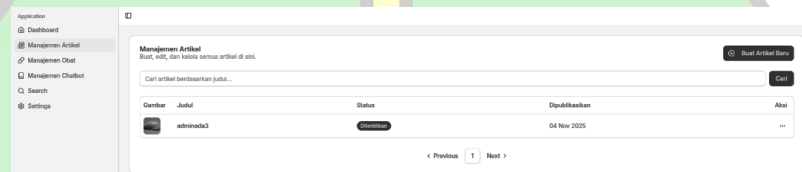
Halaman Dashboard Admin pada *Gambar 3.15* menampilkan ringkasan informasi utama terkait aktivitas sistem. Pada bagian kiri terdapat sidebar navigasi untuk mengakses modul seperti artikel, obat, dan chatbot. Bagian utama halaman berisi beberapa kartu statistik, yaitu jumlah total artikel, artikel yang dipublikasikan, draft, artikel terjadwal, dan publikasi hari ini. Di bawahnya ditampilkan daftar artikel terbaru serta tag yang paling sering digunakan. Antarmuka ini dirancang untuk memberikan gambaran cepat kepada administrator mengenai kondisi dan aktivitas konten dalam sistem.



Gambar 3. 16 Halaman Dashboard Admin

c. Halaman Manajemen Artikel

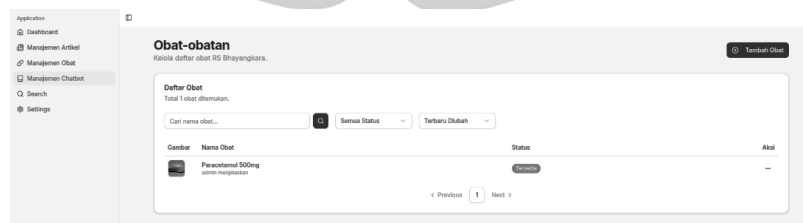
Halaman Manajemen Artikel pada Gambar 3.16 digunakan admin untuk melihat, mencari, dan mengelola artikel. Tabel menampilkan gambar, judul, status, dan tanggal publikasi, serta tombol aksi untuk pengeditan. Di kanan atas terdapat tombol **“Buat Artikel Baru”** untuk menambah artikel baru. Antarmuka ini memudahkan admin mengatur seluruh konten artikel secara cepat dan terorganisir.



Gambar 3. 17 Halaman Manajemen Artikel

d. Halaman Manajemen Obat

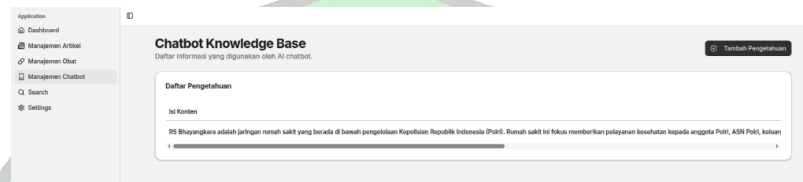
Halaman Manajemen Obat pada Gambar 3.17 digunakan administrator untuk mengelola daftar obat yang tersedia di RS Bhayangkara. Pada halaman ini admin dapat mencari obat berdasarkan nama, memfilter status, serta mengurutkan data. Tabel obat menampilkan gambar, nama obat, deskripsi singkat, status ketersediaan, dan tombol aksi untuk pengelolaan. Di bagian kanan atas terdapat tombol **“Tambah Obat”** untuk menambahkan data obat baru. Antarmuka ini dirancang agar admin dapat memperbarui dan mengatur informasi obat dengan cepat dan efisien.



Gambar 3. 18 Halaman Manajemen Obat

e. Halaman Manajemen Knowledge base Chatbot

Halaman pada *Gambar 3.18* digunakan administrator untuk mengelola daftar informasi yang menjadi sumber pengetahuan bagi chatbot. Daftar konten ditampilkan dalam bentuk list yang memuat isi pengetahuan secara ringkas. Admin dapat menambahkan informasi baru melalui tombol **“Tambah Pengetahuan”** yang tersedia di bagian kanan atas. Antarmuka ini memudahkan admin memperbarui dan memperluas data yang digunakan chatbot untuk menjawab pertanyaan pengguna.

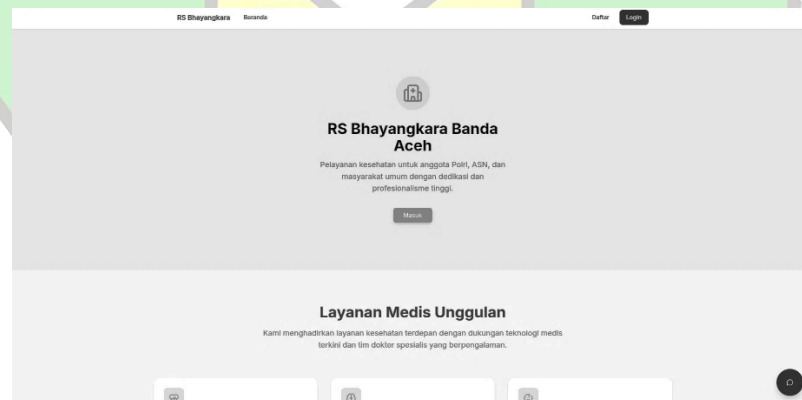


Gambar 3. 19 Halaman Manajemen Knowledge Chatbot

3.6.2 Antar Muka Pengguna Umum

a. Halaman Landing Page

Halaman landing page pada *Gambar 3.19* menampilkan informasi utama mengenai RS Bhayangkara Banda Aceh, termasuk deskripsi singkat layanan serta navigasi menuju halaman login dan pendaftaran. Halaman ini berfungsi sebagai pintu masuk bagi pengguna umum untuk mengenal sistem sebelum mengakses fitur lainnya.

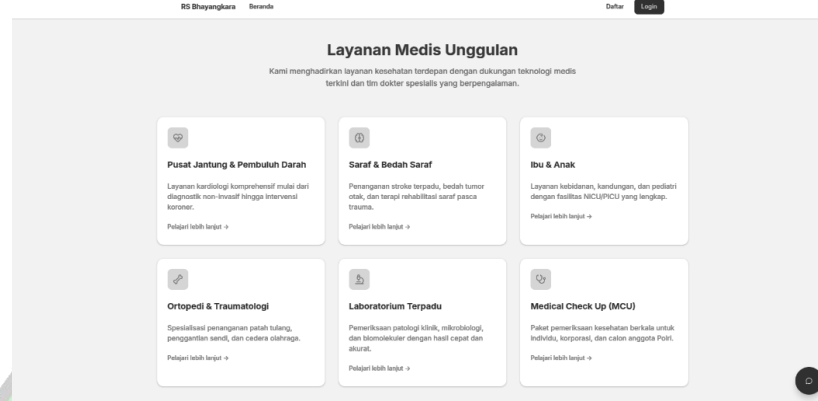


Gambar 3. 20 Halaman Landing Page

b. Halaman Layanan Medis Unggulan

Halaman layanan medis unggulan pada *Gambar 3.20* menampilkan berbagai jenis layanan kesehatan utama yang tersedia di RS Bhayangkara.

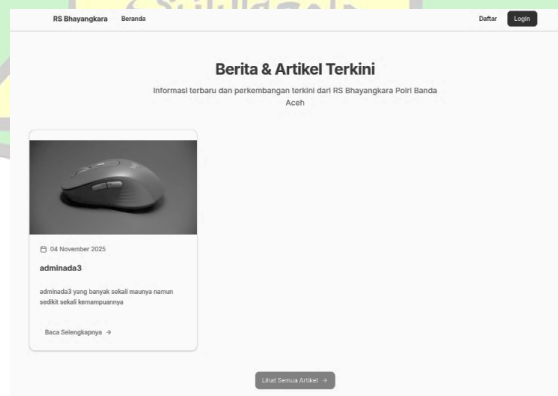
Setiap layanan disajikan dalam bentuk kartu yang berisi nama layanan, deskripsi singkat, serta tautan untuk melihat informasi lebih lanjut. Tampilan ini memudahkan pengguna umum untuk mengenali fasilitas medis yang disediakan oleh rumah sakit.



Gambar 3. 21 Halaman Layanan Medis Unggulan

c. Halaman Artikel

Halaman artikel pada Gambar 3.21 menampilkan daftar berita dan artikel terbaru yang dipublikasikan oleh RS Bhayangkara Banda Aceh. Setiap artikel ditampilkan dalam bentuk kartu yang memuat gambar, tanggal publikasi, judul, dan ringkasan singkat. Pengguna dapat membuka detail artikel melalui tombol “Baca Selengkapnya” atau melihat seluruh artikel melalui tombol “Lihat Semua Artikel”. Halaman ini berfungsi sebagai sumber informasi terkini bagi masyarakat.

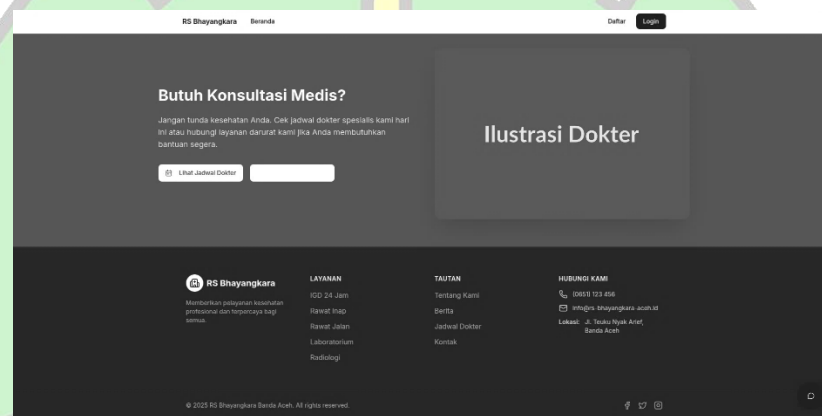


Gambar 3. 22 Halaman Artikel

d. Halaman informasi & jadwal dokter

Laman ini menampilkan bagian informasi konsultasi medis yang berfungsi sebagai pengingat bagi pengguna untuk memeriksa jadwal dokter yang tersedia. Melalui tombol “Lihat Jadwal Dokter”, pengguna dapat diarahkan langsung ke halaman jadwal dokter untuk mendapatkan informasi layanan medis yang sedang bertugas.

Pada bagian bawah laman ditampilkan footer yang memuat identitas rumah sakit, daftar layanan utama, tautan navigasi penting, serta informasi kontak resmi. Elemen ini dirancang untuk memberikan kemudahan akses terhadap informasi dasar dan meningkatkan profesionalitas tampilan sistem. Ini ditunjukkan pada *Gambar 3.22*.



Gambar 3. 23 Halaman Informasi & Jadwal Dokter

3.7 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahapan paling fundamental dalam siklus pengembangan perangkat lunak (Software Development Life Cycle). Tahapan ini bertujuan untuk mengidentifikasi, menganalisis, dan mendokumentasikan spesifikasi teknis dari sistem yang akan dibangun. Analisis ini didasarkan sepenuhnya pada hasil temuan data kualitatif yang diperoleh melalui observasi lapangan terhadap alur pelayanan dan wawancara mendalam dengan pemangku kepentingan (stakeholder) di Rumah Sakit Bhayangkara Banda Aceh.

3.7.1 Kebutuhan Fungsionalitas Sistem

Kebutuhan fungsional mendefinisikan serangkaian perilaku, fitur, dan layanan spesifik yang wajib disediakan oleh sistem untuk memfasilitasi

interaksi digital antara rumah sakit dan pasien. Kebutuhan ini dirinci berdasarkan hak akses aktor pengguna sistem:

A. Kebutuhan Pengunjung (Pasien dan Masyarakat Umum)

Pengunjung diklasifikasikan sebagai pengguna eksternal yang mengakses sistem untuk memperoleh informasi layanan kesehatan secara mandiri tanpa memerlukan proses *login*. Fitur-fitur fungsional yang wajib tersedia meliputi:

1. **Akses Informasi Jadwal Dokter yang Komprehensif:** Sistem harus mampu menyajikan data jadwal praktik dokter (baik Spesialis maupun Umum) secara akurat, *real-time*, dan terstruktur. Informasi ini mencakup nama lengkap dokter, gelar spesialisasi, poliklinik tempat bertugas, serta hari dan jam operasional praktik. Fitur ini dirancang untuk menjawab keluhan pasien terkait ketidakpastian jam praktik yang sering menyebabkan penumpukan antrean di ruang tunggu.
2. **Mesin Pencari Ketersediaan Obat (Formularium):** Sistem harus menyediakan antarmuka pencarian (*search engine*) yang responsif, di mana pasien dapat memasukkan nama obat untuk memverifikasi status ketersediaannya di instalasi farmasi RS Bhayangkara. Fitur ini berfungsi untuk meningkatkan transparansi layanan farmasi dan meminimalisir kunjungan fisik pasien yang hanya bertujuan untuk menanyakan stok obat.
3. **Interaksi Otomatis Melalui Chatbot Cerdas:** Sistem harus mengintegrasikan modul *chatbot* yang beroperasi selama 24 jam sehari (24/7). Modul ini berfungsi sebagai asisten virtual yang mampu memberikan respons otomatis dan instan terhadap pertanyaan berulang (*Frequently Asked Questions*) yang sering diajukan pasien, seperti alur pendaftaran, persyaratan jaminan kesehatan (BPJS/Asuransi), jam besuk, dan lokasi poliklinik. Keberadaan fitur ini diharapkan dapat mengurangi beban kerja staf administrasi dalam menjawab pertanyaan dasar.
4. **Portal Informasi dan Edukasi Kesehatan:** Sistem harus memfasilitasi penyebaran informasi publik melalui fitur berita kegiatan rumah sakit,

artikel edukasi kesehatan, serta profil layanan unggulan. Hal ini bertujuan untuk membangun kepercayaan publik (*public trust*) dan menjadikan website sebagai rujukan informasi resmi instansi.

B. Kebutuhan Administrator (Staf Rumah Sakit)

Administrator diklasifikasikan sebagai pengguna internal yang memiliki otoritas penuh untuk mengelola basis data dan konten sistem melalui halaman *back-office*. Fitur-fitur fungsional yang wajib tersedia meliputi:

1. **Mekanisme Autentikasi Keamanan:** Sistem wajib memfasilitasi mekanisme keamanan akses melalui halaman *login*. Administrator harus memasukkan kredensial (*username* dan *password*) yang valid dan terenkripsi untuk dapat mengakses panel kendali (*dashboard*). Hal ini krusial untuk mencegah manipulasi data oleh pihak yang tidak berwenang.
2. **Manajemen Data Medis Dinamis (CRUD):** Sistem harus menyediakan fungsi manajemen data (*Create, Read, Update, Delete*) yang mudah digunakan (*user-friendly*). Administrator harus dapat:
 - Menambah, mengubah, atau menghapus data dokter dan jadwal praktiknya secara cepat apabila terjadi perubahan jadwal mendadak.
 - Memperbarui status stok obat (Tersedia/Habis) sesuai dengan kondisi riil di gudang farmasi.
3. **Manajemen Publikasi Konten:** Administrator harus memiliki akses untuk mengelola konten informatif pada halaman depan website. Fungsi ini mencakup kemampuan untuk menulis, menyunting, dan mempublikasikan berita terbaru, pengumuman darurat, atau artikel kesehatan, sehingga informasi yang diterima masyarakat selalu mutakhir (*up-to-date*).

3.7.2 Kebutuhan Non-Fungsionalitas Sistem

A. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan berupa sebuah laptop dengan spesifikasi yang cukup untuk mendukung kebutuhan pengembangan sistem modern, mulai dari proses kompilasi kode yang berat hingga pelaksanaan pengujian aplikasi secara bersamaan.

Tabel 3. 10 Perangkat Keras

Komponen	Spesifikasi	Peran dalam Pengembangan
Processor	Intel(R) Core(TM) i5-12450H	Menjalankan proses kompilasi kode, server lokal, serta virtualisasi secara responsif tanpa hambatan.
RAM	8 GB DDR5	Memungkinkan multitasking yang lancar antara editor kode, browser dengan banyak tab, serta aplikasi pengujian sistem.
Storage	512 GB NVMe PCIe 4.0 SSD	Mempercepat waktu booting sistem, pembukaan proyek, serta proses baca/tulis data secara signifikan.
Graphics Card	NVIDIA® GeForce RTX™ 3050 atau setara	Memberikan akselerasi pada aplikasi desain UI/UX serta memastikan rendering antarmuka yang mulus.

B. Perangkat Lunak (Software)

Perangkat lunak yang digunakan membentuk suatu tumpukan teknologi (tech stack) yang saling terintegrasi untuk mendukung seluruh tahapan pengembangan sistem, mulai dari proses perancangan, implementasi, hingga pengujian.

Tabel 3. 11 Perangkat Lunak

Kategori	Nama Perangkat	Fungsi / Peran dalam Proyek
Sistem Operasi	Windows 11 (64-bit)	Menjadi platform utama untuk menjalankan seluruh alat pengembangan seperti Laravel, MySQL, Node.js, dan editor kode.
Backend Development	Laravel Framework	Mengelola routing, controller, model, middleware, dan logika bisnis sistem.
Backend Runtime	PHP 8.x	Bahasa pemrograman utama untuk mendukung proses backend sistem.
Dependency Manager	Composer	Mengelola instalasi paket Laravel dan library pendukung.

Frontend Development	React.js	Membangun tampilan antarmuka berbasis SPA (Single Page Application).
Frontend Bridge	Inertia.js	Menghubungkan Laravel dengan React tanpa memerlukan API REST tambahan.
Asset Builder	Vite	Melakukan kompilasi dan bundling asset frontend seperti CSS dan JavaScript.
Database Server	MySQL / MariaDB	Digunakan untuk menyimpan data artikel, obat, pengguna, knowledge base, dan jadwal dokter.
Code Editor	Visual Studio Code	Digunakan untuk menulis kode PHP, JavaScript, React, serta konfigurasi proyek.
Version Control	Git & GitHub	Mengelola versi source code dan mendukung kolaborasi.
Diagram Tools	PlantUML / Draw.io / Graphviz	Membuat diagram perancangan seperti ERD, Use Case, dan Activity Diagram.
Web Browser	Google Chrome	Menguji antarmuka aplikasi dan melakukan inspeksi tampilan serta fungsionalitas sistem.

3.8 Metode Evaluasi Perangkat Lunak

Evaluasi sistem dalam penelitian ini dilakukan dengan pendekatan kombinasi antara audit teknis otomatis dan mekanisme pendukung keputusan (*decision support*) berbasis aturan (*rule-based*). Tujuan dari metode ini adalah untuk memberikan penilaian objektif mengenai apakah sistem Website dan Chatbot RS Bhayangkara sudah memenuhi standar kualitas industri dan layak untuk diluncurkan.

Metode ini mengintegrasikan perangkat lunak **Google Lighthouse** sebagai penyedia data kinerja dan **Matriks Keputusan Kelayakan** sebagai kerangka kerja pengambilan kesimpulan.

3.8.1 Pengumpulan Data Kinerja (*Data Acquisition*)

Tahap pertama adalah akuisisi data kualitas teknis menggunakan perangkat pengujian otomatis **Google Lighthouse**. Perangkat ini melakukan simulasi interaksi pengguna pada website dan menghasilkan skor (skala 0-100) pada empat parameter utama yang menjadi indikator input sistem:

1. **Performa (*Performance*):** Indikator kecepatan waktu muat halaman (*loading*) dan responsivitas.
2. **Aksesibilitas (*Accessibility*):** Indikator kemudahan akses bagi berbagai karakteristik pengguna.
3. **Praktik Terbaik (*Best Practices*):** Indikator keamanan dan standar kode pemrograman modern.
4. **SEO (*Search Engine Optimization*):** Indikator keterbacaan struktur web oleh mesin pencari.

3.8.2 Logika Keputusan Kelayakan (*Decision Logic*)

Setelah skor teknis didapatkan, data tersebut diproses menggunakan **Logika Keputusan Berbasis Ambang Batas (*Threshold-based Decision Making*)**. Dalam model ini, keputusan kelayakan tidak ditentukan melalui perhitungan pembobotan yang rumit, melainkan melalui perbandingan langsung terhadap **Standar Minimum Kelayakan** yang telah ditetapkan.

Peneliti menetapkan standar kelayakan berdasarkan klasifikasi nilai yang dikeluarkan oleh Google Developers (Google Developers, n.d.), yaitu:

1. **Skor 90-100 (Hijau):** Kualitas Sempurna (*Excellent*).
2. **Skor 50-89 (Oranye):** Kualitas Sedang/Perlu Perbaikan (*Needs Improvement*).
3. **Skor 0-49 (Merah):** Kualitas Buruk (*Poor*).

Untuk kepentingan pelayanan publik di RS Bhayangkara yang menuntut keandalan tinggi, sistem pendukung keputusan ini menerapkan aturan ketat (*decision rule*) sebagai yang telah dipaparkan pada Tabel 3.10.

Tabel 3. 12 Nilai Parameter Pengujian

No	Parameter Uji	Target Skor Minimum (Ambang Batas)	Logika Keputusan
1	Performa	≥ 80	Jika skor di atas 80, maka LAYAK . Jika di bawah, maka Wajib Optimasi Ulang .

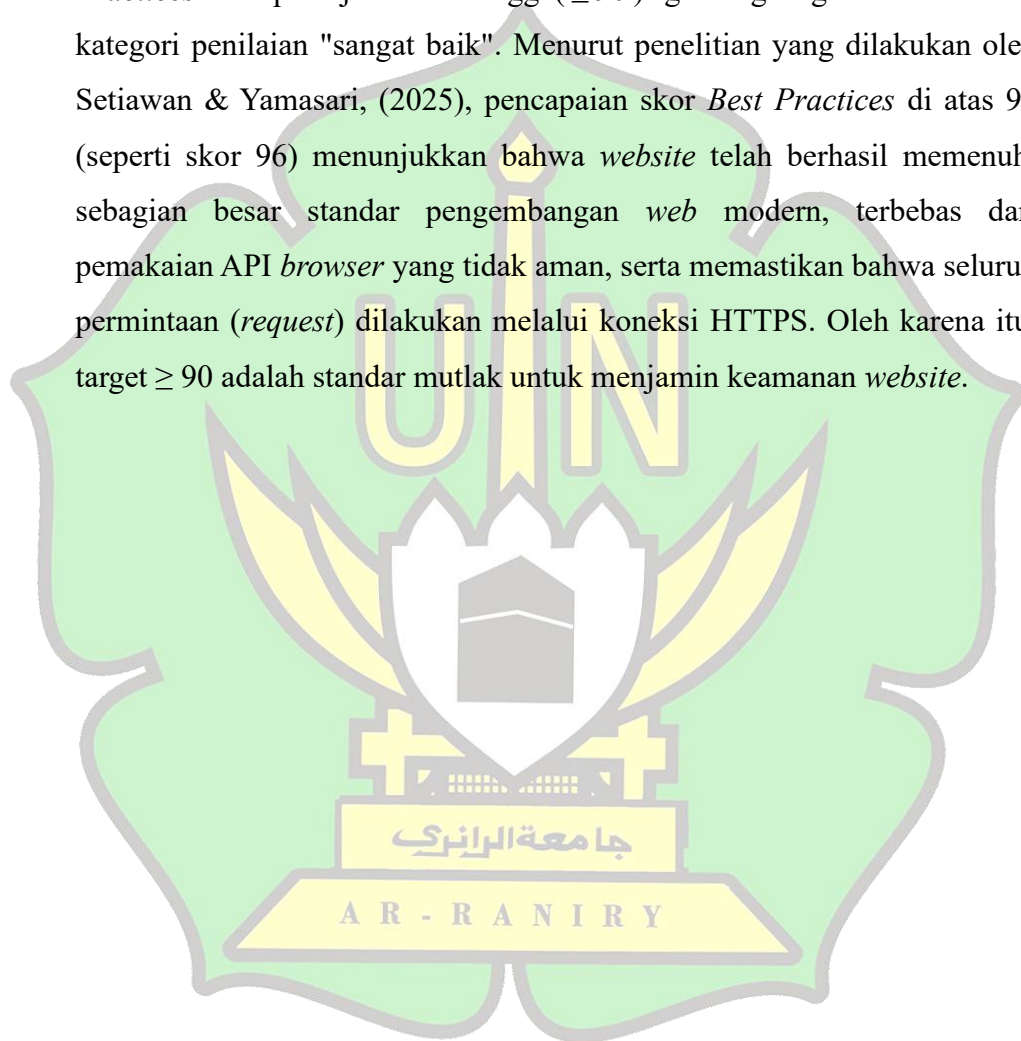
2	Aksesibilitas	≥ 85	Jika skor di atas 85, maka LAYAK . Jika di bawah, maka Wajib Revisi Desain .
3	Best Practices	≥ 90	Jika skor di atas 90, maka LAYAK . Jika di bawah, maka Cek Keamanan .
4	SEO	≥ 80	Jika skor di atas 80, maka LAYAK . Jika di bawah, maka Optimasi Meta Data .

Penetapan target skor minimum (ambang batas) untuk setiap parameter uji pada tabel tersebut dirumuskan dengan mengacu pada standar evaluasi *Google Lighthouse* yang telah dikaji dalam berbagai literatur akademik. Menurut penelitian yang dilakukan oleh Tifara et al., (2025), skor pengujian *Google Lighthouse* diklasifikasikan ke dalam tiga kategori utama, yaitu skor 90-100 untuk kategori baik (hijau), skor 50-89 untuk kategori perlu perbaikan (oranye), dan skor 0-49 untuk kategori buruk (merah). Klasifikasi ini sejalan dengan penelitian yang dilakukan oleh Setiawan & Yamasari, (2025), di mana rentang 50-89 dianggap memiliki kinerja yang cukup meskipun masih membutuhkan optimalisasi, sedangkan rentang 90-100 masuk ke dalam kategori sangat baik. Berdasarkan landasan tersebut, rasionalisasi penentuan angka pada tabel adalah sebagai berikut:

1. **Performa (≥ 80) dan SEO (≥ 80):** Menurut penelitian yang dilakukan oleh Daffa Al-farel & Rizal Dzikrillah, (2025), skor performa *website* sangat dipengaruhi oleh tingkat kompleksitas sebuah halaman, seperti banyaknya kueri *database*, beban gambar, maupun penggunaan komponen dinamis yang dapat memperlambat waktu muat halaman. Karena sulitnya mencapai skor sempurna pada *website* yang bersifat dinamis, target ≥ 80 yang berada di batas atas kategori "cukup" (rentang 50-89) sudah sangat rasional untuk menetapkan kelayakan bahwa *website* dimuat secara responsif.
2. **Aksesibilitas (≥ 85):** Menurut penelitian yang dilakukan oleh Setiawan & Yamasari, (2025), perolehan skor aksesibilitas sebesar 85 telah menunjukkan bahwa performa antarmuka *website* sudah cukup baik.

Meskipun pada perolehan angka 85 mungkin masih terdapat beberapa kendala minor seperti ketiadaan label aksesibel pada tombol atau rasio kontras warna yang belum sempurna, skor tersebut sudah memenuhi standar kelayakan yang wajar sebelum *website* dinyatakan wajib melakukan revisi desain secara besar-besaran.

3. **Best Practices (≥ 90)**: Berbeda dengan parameter performa, target *Best Practices* ditetapkan jauh lebih tinggi (≥ 90) agar langsung masuk ke dalam kategori penilaian "sangat baik". Menurut penelitian yang dilakukan oleh Setiawan & Yamasari, (2025), pencapaian skor *Best Practices* di atas 90 (seperti skor 96) menunjukkan bahwa *website* telah berhasil memenuhi sebagian besar standar pengembangan *web* modern, terbebas dari pemakaian API *browser* yang tidak aman, serta memastikan bahwa seluruh permintaan (*request*) dilakukan melalui koneksi HTTPS. Oleh karena itu, target ≥ 90 adalah standar mutlak untuk menjamin keamanan *website*.



BAB IV

IMPLEMENTASI

4.1 Gambaran Umum Implementasi Sistem

Implementasi sistem informasi terintegrasi *chatbot* di RS Bhayangkara Banda Aceh merupakan tahap transformasi desain menjadi kode program fungsional. Tahapan ini menjembatani fase Desain Produk dan Uji Coba Produk dalam metode *Research and Development* (R&D). Sistem dibangun menggunakan arsitektur *Modern Monolith*, di mana sisi *backend* dikelola dengan Laravel 12 untuk menangani logika bisnis serta algoritma *chatbot*. Sementara itu, sisi *frontend* menggunakan React.js guna menyajikan antarmuka yang cepat dan dinamis layaknya *Single Page Application* (SPA). Kedua lapisan teknologi ini dijemput secara efisien oleh Inertia.js, memungkinkan pengiriman data tanpa perlu membangun *REST API* secara manual.

4.1.1 Spesifikasi Lingkungan Pengembangan

Sistem ini dibangun dengan menggunakan spesifikasi perangkat lunak sebagai berikut:

- a. **Framework Utama (Backend):** Laravel 12
- b. **Antarmuka Pengguna (Frontend):** React.js dengan protokol Inertia.js (*Server-Side Routing*)
- c. **Algoritma Pemrosesan Teks (Chatbot):** *Vector Embeddings*, *Cosine Similarity*, dan *Generative AI* atau sering disebut dengan (*Retrieval-Augmented Generation* atau RAG)

4.2 Implementasi Kode Program dan Antarmuka (*User Interface*)

Bagian ini menguraikan implementasi kode program untuk komponen-komponen utama penyusun sistem. Pembahasan akan diklasifikasikan berdasarkan lapisan arsitekturnya, yang meliputi lapisan *Backend* (berfokus pada logika bisnis dan API), lapisan *Frontend* (menangani antarmuka dan interaksi pengguna), serta lapisan Basis Data (*Database*).

4.2.1 Implementasi Backend (Laravel 12)

Bagian *backend* berfungsi sebagai 'otak' sistem yang membereskan proses-proses utama. Tugasnya meliputi menjalankan algoritma *chatbot* agar dapat membalas pesan pasien secara otomatis, mengatur data jadwal praktik dokter, serta mengelola informasi ketersediaan obat.

a. Manajemen Rute dan Pengendali (Routing & Controllers)

Pada bagian ini, backend diimplementasikan untuk mengatur semua alur permintaan (request) dari pengguna. Laravel bertugas mendefinisikan rute (URL) apa saja yang bisa diakses, lalu mengarahkannya ke Controller yang tepat. Misalnya, saat pengguna mengakses menu "Jadwal Dokter", sistem akan mengarahkannya ke *JadwalDokterController* untuk memproses permintaan tersebut sebelum ditampilkan ke layar.

```
Route::get('/', function () {  
    $latest_articles = Article::published()->latest()->take(3)->get();  
  
    $todays_schedules = DoctorSchedule::with('doctor')->today()  
        ->where('is_available', true)  
        ->orderBy('time_start')  
        ->take(6)  
        ->get();  
  
    $featured_medicines = \App\Models\Medicine::where('is_available', true)  
        ->orderBy('name')  
        ->take(6)  
        ->get();  
  
    return Inertia::render('Welcome', [  
        'articles' => $latest_articles,  
        'schedules' => $todays_schedules,  
        'medicines' => $featured_medicines,  
    ]);  
})->middleware(TrackVisitor::class)->name('home');
```

Gambar 4. 1 Route halaman utama (landing page) website

Gambar 4.1 menunjukkan sintak untuk memuat halaman utama (landing page) website dengan mengambil data tiga artikel terbaru, enam

jadwal praktik dokter hari ini, dan enam daftar obat yang tersedia secara dinamis untuk dikirim ke komponen antarmuka React via Inertia.js

```
Route::get('/artikel', [ArticleController::class, 'publicIndex'])->name('articles.public.index');
Route::get('/artikel/{slug}', [ArticleController::class, 'publicShow'])->name('articles.public.show');
Route::get('/jadwal-dokter', [DoctorScheduleController::class, 'publicList'])->name('schedules.public.list');
Route::get('/obat-obatan', [MedicineController::class, 'publicIndex'])->name('medicines.public.index');
```

Gambar 4. 2 Route navigasi publik

Gambar 4.2 menunjukkan sintak untuk menyediakan jalur navigasi publik yang dinamis bagi pengunjung website guna mengakses halaman daftar artikel kesehatan, detail isi artikel berdasarkan parameter teks (*slug*), papan informasi jadwal praktik dokter, serta modul pengecekan daftar obat-obatan yang tersedia di rumah sakit

```
Route::middleware('guest')->group(function () {
    Route::get('masuk', [AuthController::class, 'showLogin'])->name('login');
    Route::post('masuk', [AuthController::class, 'login']);
    Route::get('daftar', [AuthController::class, 'showRegister'])->name('register');
    Route::post('daftar', [AuthController::class, 'register']);
});

Route::post('logout', [AuthController::class, 'logout'])->name('logout');
```

Gambar 4. 3 Route Pengunjung

Gambar 4.3 menunjukkan sintak untuk mengelola siklus otentikasi akun, di mana kelompok *middleware* guest membatasi akses pengisian formulir pendaftaran dan masuk sistem khusus bagi pengunjung yang belum login, sementara rute logout menyediakan jalur aman untuk memutus sesi aktif pengguna.

```

Route::middleware(['auth', 'role:admin'])
->prefix('admin')
->group(function () {
    Route::get('/dashboard', [DashboardController::class, 'index'])->name('dashboard');
    Route::resource('articles', ArticleController::class);
    Route::post('/upload-image', [ArticleController::class, 'uploadEditorImage'])->name('upload-image');
    Route::patch('/medicines/{medicine}/toggle', [MedicineController::class, 'toggleAvailability'])->name('medicines.toggle-availability');
    Route::resource('medicines', MedicineController::class);
    Route::resource('knowledge', KnowledgeBaseController::class)
        ->only(['index', 'create', 'store', 'edit', 'update', 'destroy']);
    Route::resource('doctors', DoctorController::class);
    Route::resource('schedules', App\Http\Controllers\DoctorScheduleController::class)->except(['show']);
    Route::resource('rooms', App\Http\Controllers\RoomController::class)->except(['show']);

    Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
    Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
    Route::put('/password', [ProfileController::class, 'updatePassword'])->name('profile.password.update');
});

```

Gambar 4. 4 Route Admin

Gambar 4.4 menunjukkan sintak untuk memproteksi area *back-office* menggunakan keamanan berlapis (*auth* dan *role:admin*) dengan *prefix* URL */admin* guna mengelola data analitik *dashboard*, profil administrator, serta mengotomatisasi fungsi CRUD data master rumah sakit seperti artikel, inventaris obat, basis pengetahuan *chatbot*, profil dokter, jadwal praktik, dan ketersediaan kamar rawat inap.

b. Pengelolaan Basis Data dan ORM (Object-Relational Mapping)

Backend bertugas menjembatani aplikasi dengan database MySQL. Implementasi ini mencakup pembuatan Migration (struktur tabel otomatis) untuk tabel-tabel krusial seperti tabel *users* (admin), *jadwal_dokter*, *ketersediaan_obat*, dan *knowledge_base* (untuk *chatbot*). Selain itu, backend menggunakan fitur Eloquent ORM dari Laravel agar proses CRUD (Create, Read, Update, Delete) data oleh admin dapat dilakukan dengan aman dan efisien.

```
Schema::create('users', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('email')->unique();
    $table->timestamp('email_verified_at')->nullable();
    $table->string('password');
    $table->unsignedBigInteger('role_id')->nullable()->after('id');
    $table->foreign('role_id')->references('id')->on('roles')->onDelete('set null');
    $table->rememberToken();
    $table->timestamps();
});
```

Gambar 4. 5 create tabel user

```
Schema::create('roles', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('slug')->unique();
    $table->timestamps();
});
```

Gambar 4. 6 create tabel roles

A R - R A N I R Y

```

schema::create('articles', function (Blueprint $table) {
    $table->id();
    $table->foreignId('user_id')->nullable()->constrained()->onDelete('set null');
    $table->string('title');
    $table->string('slug')->unique();
    $table->longText('body')->nullable();
    $table->string('image')->nullable();
    $table->json('tags')->nullable();

    // SEO & metadata
    $table->string('meta_title')->nullable();
    $table->string('meta_description')->nullable();
    $table->string('meta_keywords')->nullable();

    // Status
    $table->enum('status', ['draft', 'published'])->default('draft');
    $table->timestamp('published_at')->nullable();
    $table->timestamps();
});

```

Gambar 4. 7 create tabel articles

```

Schema::create('medicines', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('slug')->unique();
    $table->string('description')->nullable();
    $table->string('image')->nullable();
    $table->boolean('is_available')->default(true);

    //Metadata
    $table->string('meta_title')->nullable();
    $table->string('meta_description')->nullable();
    $table->string('meta_keywords')->nullable();
    $table->timestamps();
});

```

Gambar 4. 8 create tabel medicines

```
Schema::create('knowledge_base', function (Blueprint $table) {
    $table->id();
    $table->longText('content');
    $table->longText('embedding');
    $table->timestamps();
});
```

Gambar 4. 9 create tabel knowledge_base

```
Schema::create('doctors', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('specialist');
    $table->timestamps();
});
```

Gambar 4. 10 create tabel doctors

```
Schema::create('doctor_schedules', function (Blueprint $table) {
    $table->id();
    $table->string('doctor_name');
    $table->string('specialist'); // Poli / Spesialis (e.g. "Penyakit Dalam")
    $table->enum('day', ['Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu', 'Minggu']);
    $table->time('time_start');
    $table->time('time_end');
    $table->boolean('is_available')->default(true); // Untuk status "Cuti" atau "Hadir"
    $table->timestamps();
});
```

Gambar 4. 11 create tabel doctor_schedules

```
Schema::create('rooms', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('class');
    $table->integer('capacity')->default(0);
    $table->integer('available')->default(0);
    $table->boolean('is_bpjs')->default(false);
    $table->foreignId('created_by')->nullable()->constrained('users')->onDelete('cascade');
    $table->timestamps();
});
```

Gambar 4. 12 create tabel rooms

c. Implementasi Logika Pemrosesan Chatbot

Implementasi ini mencakup pembuatan sebuah Controller (*ChatbotController*) di dalam Laravel yang bertugas menerima teks pertanyaan dari pasien. Sistem kemudian mengubah teks tersebut menjadi representasi numerik (*Vector Embeddings*) menggunakan model AI, lalu menjalankan perhitungan matematis menggunakan algoritma Cosine Similarity. Fungsi ini akan membandingkan vektor input pasien dengan kumpulan vektor data *Knowledge Base* di database untuk mencari skor kemiripan tertinggi. Jika ditemukan data yang relevan, data tersebut dijadikan konteks informasi, yang selanjutnya diproses oleh model AI Generatif untuk menyusun dan mengembalikan jawaban akhir yang natural kepada pasien.

```
public function handle(Request $request)
{
    $validated = $request->validate([
        'message' => 'required|string|min:1|max:500',
    ]);

    $userQuestion = $validated['message'];
    $ragContext = null;
}
```

Gambar 4. 13 Validasi Input

Gambar 4.13 menunjukkan sintak untuk memvalidasi kueri atau obrolan masuk dari pasien. Teks pertanyaan wajib diisi (required), berformat

karakter string, minimal terdiri dari 1 karakter, dan dibatasi maksimal 500 karakter untuk mencegah beban komputasi berlebih (*denial of service*) pada sistem AI.

```
try {
    $apiKey = env('GEMINI_API_KEY');
    if (empty($apiKey)) {
        throw new \Exception('GEMINI_API_KEY is not set in .env file.');
```

Gambar 4. 14 Inisialisasi API Client dan Pemrosesan Vektor Teks (Embedding)

Gambar 4.14 menunjukkan sintaks yang membaca kunci API dari berkas konfigurasi lingkungan (.env) untuk mengaktifkan SDK Gemini Client. Jika tabel knowledge_base berisi data, sistem memicu model AI gemini-embedding-2 untuk mengonversi teks input pengguna menjadi bentuk koordinat multidimensi (*Vector Embeddings*) yang disimpan ke dalam variabel \$questionVector

```
$bestMatch = $this->findBestMatch($questionVector, $allNotes);

$similarityThreshold = 0.75;
if ($bestMatch && $bestMatch['score'] >= $similarityThreshold) {
    $ragContext = $bestMatch['note']->content;
}
}
```

Gambar 4. 15 Eksekusi Algoritma RAG dan Batas Kelayakan (Similarity Threshold)

Gambar 4.15 menunjukkan Sintaks untuk memanggil fungsi internal findBestMatch untuk membandingkan vektor pertanyaan dengan seluruh

dataset latih di database SQLite. Dipatok batas ambang kelayakan (*similarity threshold*) yang ketat sebesar **0.75**. Jika skor kemiripan sama atau lebih besar dari 0.75, teks dokumen resmi rumah sakit tersebut sah dicabut untuk dijadikan variabel \$ragContext (konteks jawaban AI).

```
$systemPrompt = $this->createSystemPrompt($ragContext);

// Use the native systemInstruction API to prevent prompt injection
$response = $client->generativeModel(model: 'gemini-2.5-flash')
->withSystemInstruction($Gemini\Data\Content::parse($systemPrompt))
->generateContent($userQuestion);

return response()->json(['reply' => $response->text()]);

} catch (Throwable $e) {
    report($e);
    return response()->json(['reply' => 'Maaf, terjadi kesalahan pada sistem AI kami. Silakan coba lagi nanti.'], 500);
}
```

Gambar 4. 16 Metode Native System Instruction Gemini API (Generasi Respon)

Gambar 4.16 menunjukkan sintaks untuk memanfaatkan model gemini-2.5-flash dengan menyuntikkan instruksi melalui fitur bawaan `withSystemInstruction`. Pendekatan ini merupakan proteksi keamanan terbaik untuk mengisolasi peran AI dan meminimalkan celah manipulasi perintah oleh peretas (*prompt injection*). Hasil generasi teks dikembalikan ke *widget chat* pasien dalam bentuk format JSON responsif.

```

private function getPublicHospitalDataContext(): string
{
    $medicines = Medicine::where('is_available', true)->select('name', 'description')->get()->toArray();

    $schedules = DoctorSchedule::with(['doctor' => function ($query) {
        $query->select('id', 'name', 'specialist');
    }])
    ->where('is_available', true)
    ->select('doctor_id', 'day', 'time_start', 'time_end')
    ->get()->toArray();

    $rooms = Room::select('name', 'class', 'capacity', 'available', 'is_bpjs')->get()->toArray();

    $context = "DATA RUMAH SAKIT SAAT INI:\n\n";
    // ... [looping data obat, jadwal dokter, dan kamar ke dalam string] ...
    return $context;
}

```

Gambar 4. 17 Kompilasi Data Struktural (Structured Data Context)

Gambar 4.17 menunjukkan sintaks untuk melakukan kueri langsung (*direct query*) ke database SQLite menggunakan Eloquent ORM untuk mengambil data terstruktur real-time rumah sakit, yang meliputi nama obat aktif, jadwal dinas harian dokter penanggung jawab, serta status ketersediaan sisa bed kamar rawat inap BPJS/Umum untuk digabungkan menjadi teks narasi utuh.

```

private function createSystemPrompt(?string $ragContext): string
{
    $basePrompt = "Anda adalah Asisten Virtual AI dari RS Bhayangkara Banda Aceh.
    Sapa pengguna dengan ramah (misalnya 'Halo!' atau 'Tentu, saya bantu').
    Jawablah dalam bahasa Indonesia yang profesional, sopan, dan empatik.
    Identitas Anda adalah asisten, bukan dokter. Jangan pernah memberikan diagnosis medis.
    Anda akan diberikan Data Rumah Sakit (seperti daftar obat dan jadwal dokter) serta Konteks Tambahan.
    Gunakan data tersebut untuk menjawab pertanyaan pengguna.
    Jika pengguna menanyakan informasi spesifik yang tidak ada dalam data atau konteks (seperti biaya perawatan spesifik atau
    ketersediaan kamar),
    katakan dengan sopan bahwa Anda tidak memiliki informasi tersebut dan sarankan untuk menghubungi resepsionis kami di
    (0651) 123-456 atau datang langsung ke rumah sakit.";

    $hospitalData = $this->getPublicHospitalDataContext();

    $fullPrompt = $basePrompt . "\n\n" . "=== DATA STRUKTURAL (JADWAL & OBAT) ===\n" . $hospitalData;

    if ($ragContext) {
        $fullPrompt .= "\n\n=== KONTEKS TAMBAHAN DARI KNOWLEDGE BASE ===\n" . $ragContext;
    }

    return $fullPrompt;
}

```

Gambar 4. 18 Perancangan Dokumen Instruksi

Gambar 4.18 menunjukkan sintaks untuk menyusun draf perintah utama (*system prompt*). Di dalamnya diatur pembatasan ketat (AI dilarang memberikan diagnosis medis), digabungkan dengan data dinas rumah sakit (*\$hospitalData*), serta ditambahkan teks dokumen eksternal (*\$ragContext*) jika lolos dari penyaringan ambang batas algoritma sebelumnya.

```
private function findBestMatch(array $questionVector, $allNotes): ?array
{
    $bestScore = -1.0;
    $bestNote = null;

    foreach ($allNotes as $note) {
        if (empty($note->embedding) || !is_array($note->embedding)) {
            continue;
        }

        $score = $this->cosineSimilarity($questionVector, $note->embedding);

        if ($score > $bestScore) {
            $bestScore = $score;
            $bestNote = $note;
        }
    }

    if ($bestNote === null) {
        return null;
    }

    return ['note' => $bestNote, 'score' => $bestScore];
}
```

Gambar 4. 19 Perhitungan Geometri dan Pencarian Nilai Tertinggi

Gambar 4.19 menunjukkan sintaks untuk melakukan pencarian berulang (*looping*) untuk menyaring baris pengetahuan yang memiliki kecocokan makna tertinggi.

```
private function cosineSimilarity(array $a, array $b): float
{
    $dotProduct = 0.0;
    $magA = 0.0;
    $magB = 0.0;
    $count = count($a);

    if ($count === 0 || $count !== count($b)) {
        return 0.0;
    }

    for ($i = 0; $i < $count; $i++) {
        $dotProduct += ($a[$i] ?? 0) * ($b[$i] ?? 0);
        $magA += ($a[$i] ?? 0) ** 2;
        $magB += ($b[$i] ?? 0) ** 2;
    }

    $magA = sqrt($magA);
    $magB = sqrt($magB);

    if ($magA == 0.0 || $magB == 0.0) {
        return 0.0;
    }

    return $dotProduct / ($magA * $magB);
}
```

Gambar 4. 20 Rumus Matematis Klasifikasi (Cosine Similarity)

Gambar 4.20 mengimplementasikan rumus aljabar linear murni untuk mendeteksi tingkat kemiripan makna kueri secara semantik.

d. Autentikasi dan Keamanan Sistem (Middleware)

Backend juga diimplementasikan untuk menjaga keamanan sistem, khususnya pada halaman Dashboard Admin. Laravel mengelola proses otentikasi (verifikasi username dan password admin yang dienkripsi). Selain itu, diterapkan Middleware yang berfungsi sebagai "penjaga pintu" untuk memastikan bahwa halaman manajemen data (tambah/edit jadwal dan obat) hanya bisa diakses oleh admin yang sudah berhasil login, sementara pengguna biasa atau peretas akan ditolak aksesnya.

```
public function login(Request $request): RedirectResponse
{
    $credentials = $request->validate([
        'email' => ['required', 'email'],
        'password' => ['required'],
    ]);
}
```

Gambar 4. 21 Validasi Data

Gambar 4.21 menunjukkan sintaks untuk menangani validasi awal terhadap data email dan kata sandi yang dikirimkan oleh pengguna melalui formulir login.

```
if (Auth::attempt($credentials, $request->boolean('remember'))) {
    $request->session()->regenerate();

    return redirect()->intended('/');
};

return back()->withErrors([
    'email' => 'Email yang anda masukkan tidak terdaftar.',
    'password' => 'Password yang anda masukkan salah.',
])->withInput($request->only('email', 'password'));
```

Gambar 4. 22 Otentikasi dan Regenerasi Sesi

Gambar 4.22 menunjukkan sintaks untuk melakukan verifikasi keamanan kredensial ke database serta memperbarui sesi pengguna jika otentikasi berhasil.

```

public function register(Request $request): RedirectResponse {
    $validated = $request->validate([
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'email'],
        'password' => ['required'],
    ]);

    $user = User::create([
        'name' => $validated['name'],
        'email' => $validated['email'],
        'password' => Hash::make($validated['password']),
        'role_id' => 3,
    ]);
}

```

Gambar 4. 23 Validasi dan Pembuatan Akun Pasien

Gambar 4.23 menunjukkan sintaks untuk menangani pendaftaran pengguna baru, enkripsi kata sandi, dan penentuan tingkatan hak akses default.

```

Auth::login($user);

return redirect()->intended('/');
}

```

Gambar 4. 24 Login Otomatis

Gambar 4.24 menunjukkan sintaks untuk memberikan kenyamanan pengalaman pengguna agar akun baru bisa langsung aktif berselancar di dalam sistem.

```

public function logout(Request $request): RedirectResponse {

    Auth::logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();

    return redirect()->route('login');

}

```

Gambar 4. 25 Fungsi Keluar Akun

Gambar 4.25 menunjukkan sintaks untuk menangani pemutusan hubungan komunikasi asinkron antara pengguna dengan hak akses di server.

e. Penjemabatan Data ke Frontend (Integrasi Inertia.js)

Berbeda dengan sistem API tradisional (REST API), backend pada sistem Modern Monolith ini diimplementasikan agar dapat langsung mengirimkan data ke komponen antarmuka (React.js). Saat Controller di Laravel selesai mengambil data dari database (misalnya daftar stok obat), backend akan membungkus data tersebut dan mengirimkannya melalui protokol Inertia.js sebagai props, sehingga halaman web dapat dimuat dengan sangat cepat tanpa memuat ulang (reload) halaman.

```

protected $rootView = 'app';

```

Gambar 4. 26 Penentuan Berkas Tampilan Utama

Gambar 4.26 menunjukkan sintaks untuk menentukan file HTML dasar yang akan digunakan Laravel untuk memuat aplikasi monolith React Anda.

```

public function share(Request $request): array
{
    return array_merge(parent::share($request), [
        'auth' => [
            'user' => $request->user(),
        ],
    ],

```

Gambar 4. 27 Sinkronisasi Sesi Data Pengguna Terautentikasi

Gambar 4.27 menunjukkan sintaks untuk otomatis membagikan data profil pengguna yang sedang login ke seluruh komponen halaman React tanpa perlu melakukan kueri berulang.

```

'navbar' => [
    'featured_articles' => Cache::remember('navbar_articles', 60, function () {
        return Article::published()
            ->latest()
            ->take(2)
            ->select('id', 'title', 'slug', 'image', 'published_at') // Select only what you need
            ->get();
    }),
],
]);

```

Gambar 4. 28 Optimalisasi Navigasi dan Sistem Caching Data Artikel

Gambar 4.28 menunjukkan sintaks untuk membagikan data artikel unggulan untuk komponen bilah navigasi (*navbar*) dengan memanfaatkan memori sementara (*cache*) demi menghemat beban kueri pangkalan data SQLite.

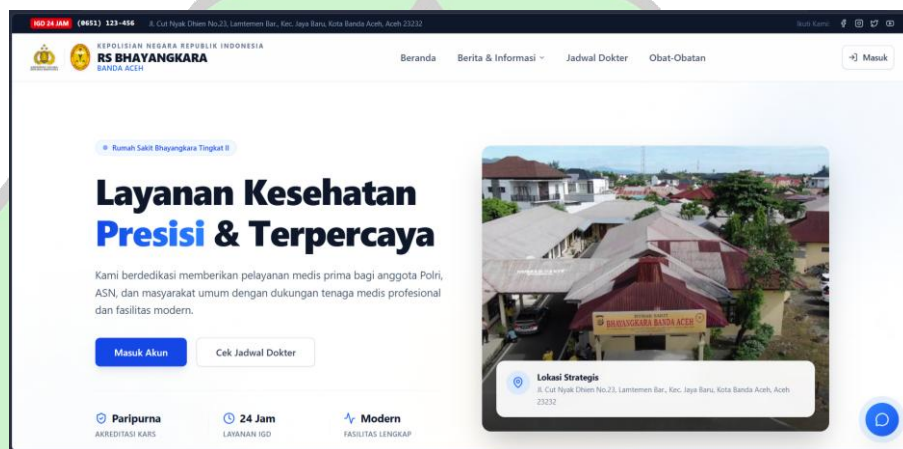
4.2.2 Implementasi Sisi Frontend (*React.js* + *Inertia.js*)

Sisi antarmuka pengguna (*frontend*) pada sistem informasi Rumah Sakit Bhayangkara Banda Aceh dikembangkan menggunakan pustaka **React.js** yang diintegrasikan dengan *backend* **Laravel** melalui protokol **Inertia.js**. Penerapan arsitektur *Modern Monolith* ini memungkinkan aplikasi beroperasi sebagai *Single Page Application* (SPA), di mana transisi antarhalaman seperti perpindahan dari informasi jadwal dokter ke pengecekan stok obat terjadi

secara instan tanpa memerlukan muat ulang seluruh sumber daya halaman (*full reload*).

Secara teknis, pendekatan ini memberikan pengalaman pengguna yang sangat responsif dan mulus layaknya aplikasi *native*, yang sangat krusial bagi masyarakat dalam mengakses informasi medis secara cepat. Selain itu, penggunaan React.js memungkinkan antarmuka dibangun secara modular, sehingga setiap komponen seperti *widget chatbot* dapat berinteraksi secara asinkron tanpa mengganggu aktivitas navigasi pengguna di halaman lainnya.

a. Halaman Beranda (*Landing Page*)

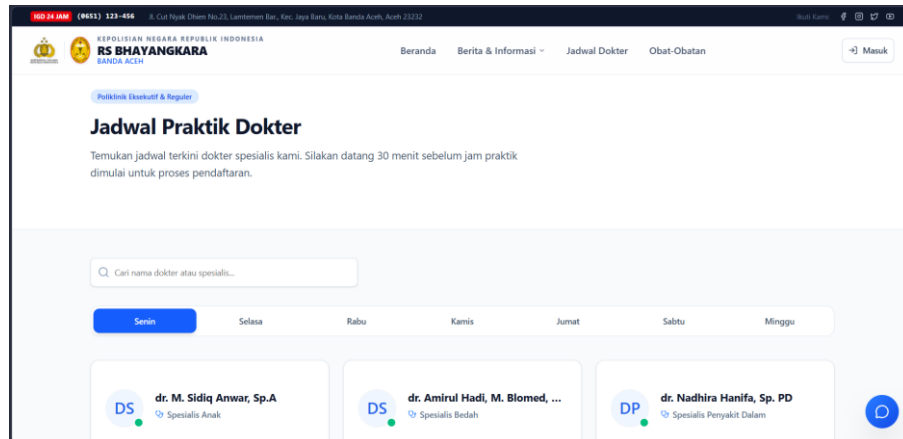


Gambar 4. 29 Halaman Beranda Website

Pada Gambar 4.1 menunjukkan halaman Beranda sistem yang menampilkan profil singkat Rumah Sakit Bhayangkara Banda Aceh dan layanan informasi unggulan instansi. Di bagian utama terdapat elemen visual *Hero Section* berupa gambar latar dan kalimat sambutan, serta tombol akses cepat (*Quick Access*) yang mengarahkan pengguna ke fitur-fitur layanan krusial tanpa harus mencari di menu navigasi utama. Halaman ini berfungsi sebagai titik awal navigasi yang intuitif bagi masyarakat. Komponen utama yang ditampilkan adalah:

- 1) Logo dan profil singkat rumah sakit.
- 2) Tombol akses cepat menuju menu Jadwal Dokter, Stok Obat, dan Berita.
- 3) Informasi kontak darurat (*emergency call*) dan jam operasional pelayanan.
- 4) *Widget* asisten virtual (*chatbot*) yang melayang di sudut layar dan siap menerima interaksi pasien.

b. Halaman Jadwal Dokter

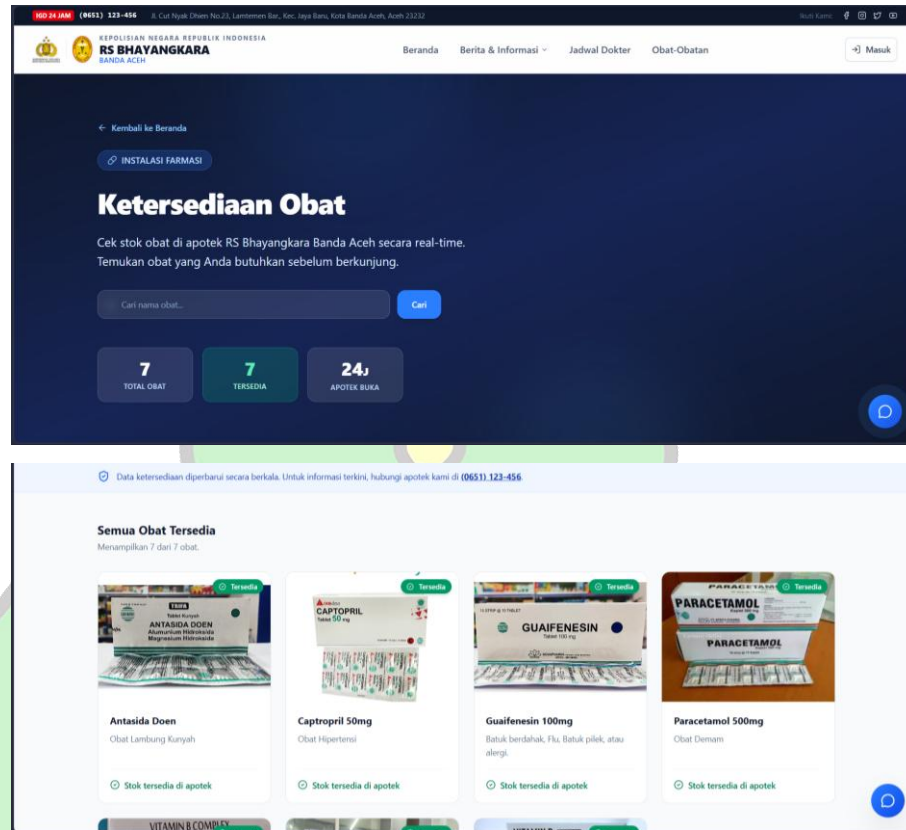


Gambar 4. 30 Halaman Jadwal Dokter

Pada Gambar 4.2 menunjukkan halaman Jadwal Praktik Dokter, yang menampilkan daftar tenaga medis yang beroperasi di rumah sakit secara dinamis. Di bagian utama terdapat kolom pencarian (*search bar*) untuk memudahkan pasien menemukan dokter spesialis yang dibutuhkan dengan cepat. Selain itu, sistem menyajikan informasi hasil pencarian dalam bentuk kartu profil dokter yang terstruktur. Halaman ini berfungsi untuk memberikan transparansi waktu layanan medis kepada publik. Komponen utama yang ditampilkan adalah:

- 1) Kolom pencarian nama dokter dan *dropdown* filter poliklinik.
- 2) Kartu informasi visual dokter (foto, nama, dan gelar spesialisasi).
- 3) Tabel ringkasan hari dan jam praktik untuk masing-masing dokter.

c. Halaman Informasi Stok Obat

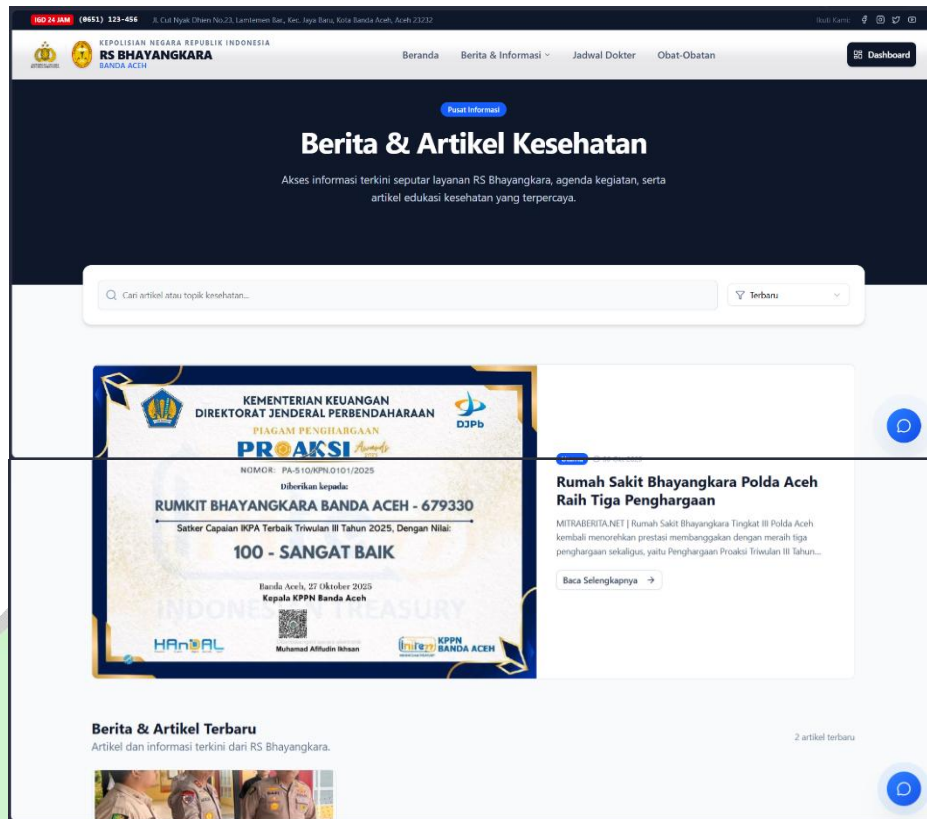


Gambar 4. 31 Halaman Informasi Stok Obat

Pada Gambar 4.3 menunjukkan halaman Informasi Stok Obat, yang menampilkan daftar inventaris farmasi yang tersedia di rumah sakit. Di bagian utama terdapat tabel informasi yang memuat nama obat beserta indikator status ketersediaannya (seperti "Tersedia" atau "Habis"). Halaman ini sangat membantu pasien dalam memastikan ketersediaan obat secara *real-time* sebelum mereka datang langsung ke instalasi farmasi. Komponen utama yang ditampilkan adalah:

- 1) Kolom pencarian cerdas untuk mencari nama obat spesifik.
- 2) Daftar nama obat beserta jenis/kategorinya.
- 3) Label status ketersediaan dengan indikator warna visual (misal: hijau untuk tersedia, merah untuk habis).

d. Halaman Berita & Informasi



Gambar 4. 32 Halaman Berita & Informasi

Pada Gambar 4.4 menunjukkan halaman Berita dan Edukasi, yang menampilkan kumpulan artikel kesehatan, pengumuman resmi instansi, serta kegiatan terbaru dari Rumah Sakit Bhayangkara Banda Aceh. Di bagian utama terdapat susunan kartu berita (*news cards*) yang dirancang secara visual untuk menarik minat baca pengunjung. Selain itu, sistem menyediakan kolom pencarian judul berita di bagian atas agar memudahkan pencarian berita yang diinginkan. Halaman ini berfungsi sebagai portal literasi kesehatan dan sarana komunikasi proaktif antara pihak rumah sakit dengan masyarakat. Komponen utama yang ditampilkan adalah:

- 1) Kolom pencarian untuk mencari judul berita atau pengumuman spesifik.
- 2) Kartu informasi berita yang memuat gambar *thumbnail*, judul artikel, cuplikan isi teks, dan tanggal publikasi.

e. Antarmuka *Widget Chatbot* (Asisten Virtual)

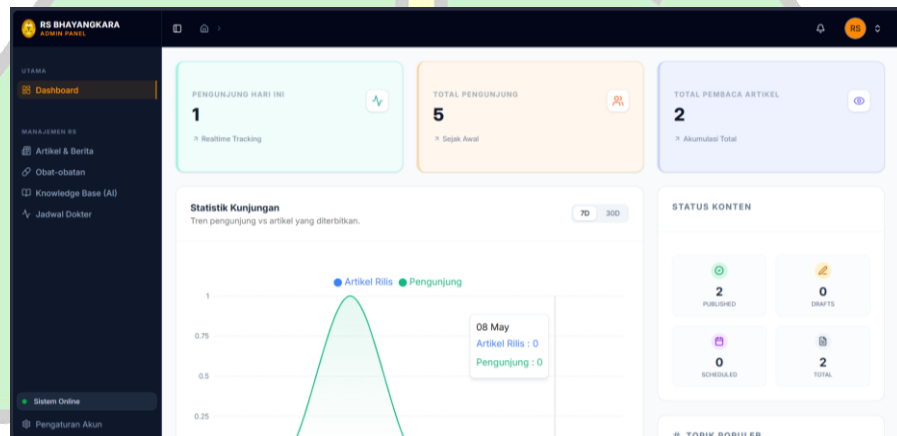


Gambar 4. 33 Antarmuka *Widget Chatbot*

Pada Gambar 4.5 menunjukkan antarmuka *Widget Chatbot*, yang menampilkan jendela percakapan interaktif yang terintegrasi dengan pemrosesan kecerdasan buatan. Di bagian utama, terdapat ruang obrolan (*chat area*) berdesain *bubble chat* yang menampilkan riwayat pesan antara pengguna dan asisten virtual secara *real-time*. Di bagian bawah antarmuka, disediakan kolom input teks bagi pasien untuk mengetikkan pertanyaan seputar layanan rumah sakit, dilengkapi dengan tombol kirim. Antarmuka ini dirancang secara melayang (*floating*) pada sudut layar sehingga selalu dapat diakses tanpa mengganggu aktivitas navigasi pengguna di halaman lain. Fitur ini berfungsi sebagai pusat layanan informasi proaktif 24 jam yang siap memberikan jawaban natural dan akurat berdasarkan data *Knowledge Base* instansi untuk meminimalisasi beban kerja staf administrasi. Komponen utama yang ditampilkan adalah:

- 1) Ikon *widget* melayang (*floating trigger button*) untuk membuka atau menyembunyikan jendela obrolan.
- 2) Ruang riwayat percakapan yang membedakan pesan input dari pengguna dan pesan balasan dari sistem secara visual.
- 3) Kolom input teks dan tombol aksi pengiriman pesan (ikon *send*).
- 4) Indikator "sedang mengetik" (*typing indicator*) yang memberikan umpan balik visual kepada pengguna selama sistem memproses vektor algoritma dan menyusun jawaban akhir.

e. Halaman Dashboard Admin



Gambar 4. 34 Halaman Dashboard Admin

Pada Gambar 4.5 menunjukkan halaman *Dashboard* Administrator, yang menampilkan ringkasan data analitik kunjungan dan status operasional konten secara *real-time*. Di bagian utama, terdapat kartu-kartu metrik (*Summary Cards*) yang menonjolkan statistik lalu lintas pengguna, seperti jumlah pengunjung hari ini, total pengunjung keseluruhan, dan akumulasi pembaca artikel. Selain itu, terdapat visualisasi grafik garis (*line chart*) pada bagian "Statistik Kunjungan" yang memetakan tren kunjungan berbanding dengan perilsan artikel dalam rentang waktu tertentu. Halaman ini berfungsi sebagai pusat kendali pemantauan (*monitoring*) utama bagi petugas administrasi rumah sakit untuk mengevaluasi jangkauan informasi digital sebelum melakukan pengelolaan data lebih lanjut. Komponen utama yang ditampilkan adalah:

- 1) **Panel Navigasi Samping (Sidebar):** Terletak di sebelah kiri, memuat identitas instansi (RS Bhayangkara Admin Panel) beserta daftar menu manajemen operasional seperti Artikel & Berita, Obat-obatan, *Knowledge Base* (AI), dan Jadwal Dokter, dilengkapi dengan indikator status *Sistem Online* serta Pengaturan Akun di bagian bawah.
- 2) **Kartu Statistik Utama (Top Summary Cards):** Menampilkan metrik krusial sistem dengan antarmuka yang bersih, mencakup data "Pengunjung Hari Ini", "Total Pengunjung", dan "Total Pembaca Artikel".
- 3) **Grafik Analitik Interaktif:** Menampilkan kurva tren yang membandingkan jumlah pengunjung dengan artikel yang dirilis, dilengkapi tombol filter periode pemantauan (7 Hari atau 30 Hari terakhir).
- 4) **Panel Status Konten:** Memberikan ringkasan numerik secara cepat mengenai siklus publikasi, yang mencakup jumlah artikel yang sudah diterbitkan (*Published*), draf, terjadwal (*Scheduled*), dan total keseluruhan artikel.
- 5) **Bilah Navigasi Atas (Top Bar):** Menyediakan elemen utilitas tambahan, termasuk ikon notifikasi lonceng dan ikon profil pengguna yang sedang masuk (*login*).

4.3 Pengujian Fungsionalitas Sistem (*Black Box Testing*)

4.3.1 Sasaran Pengujian

Metode *Black Box* diaplikasikan guna mengevaluasi kelayakan operasional dari *website* pelayanan terintegrasi *chatbot* di RS Bhayangkara Banda Aceh. Pengujian ini berpusat sepenuhnya pada antarmuka dan interaksi pengguna tanpa perlu meninjau kerangka internal kode program. Adapun sasaran pokok dari uji coba ini meliputi:

1. Memverifikasi fungsionalitas seluruh modul utama (Otentikasi, *Dashboard*, Pengelolaan Artikel, Manajemen Obat, Jadwal Dokter, dan Layanan Asisten Virtual).

2. Mengevaluasi mekanisme pembatasan hak akses (*access control*), guna memastikan setiap jenis pengguna hanya mampu mengoperasikan fitur yang sejalan dengan otoritasnya.
3. Memastikan validitas respons sistem, baik yang berupa penyaringan data, transisi halaman, maupun kualitas interaksi percakapan kecerdasan buatan.

4.3.2 Skenario dan Indikator Kelayakan

1. **Mekanisme Pengujian:** Penguji melakukan serangkaian instruksi langsung pada elemen *frontend* aplikasi (seperti menekan tombol navigasi, mengisi formulir data, dan memasukkan kueri pencarian) untuk melihat respons dari sisi *backend*.
2. **Indikator Kelayakan:** Sebuah fitur disimpulkan berstatus **Valid** apabila respons atau hasil aktual yang dimunculkan oleh aplikasi selaras dengan target yang telah dirumuskan pada perancangan sistem awal.

4.3.3 Pengujian Berdasarkan Klasifikasi Pengguna

Proses validasi sistem dibagi menjadi dua segmen utama berdasarkan level interaksinya, yakni pada sisi pengelola (Administrator) dan sisi pengunjung (Masyarakat Umum).

A. Pengujian pada Otoritas Administrator (Staf / IT RS Bhayangkara)

Administrator memegang kendali penuh atas sistem manajemen konten (CMS) dan bertugas melatih basis pengetahuan (*knowledge base*) yang akan digunakan oleh *chatbot*. Berikut modul pengujiannya di lampirkan pada Tabel 4.1.

Tabel 4.1 Testing Administrator

Modul Uji	Skenario Pengujian
Auth	Uji coba akses masuk (Login)
Auth	Uji coba pengakhiran sesi (Logout)
Dashboard	Validasi visualisasi data analitik
Manajemen Artikel	Penambahan publikasi edukasi baru
Manajemen Obat	Penambahan & pembaruan status ketersediaan farmasi
Manajemen Dokter	Penambahan nama-nama dokter

Manajemen Jadwal Dokter	Pendaftaran jadwal praktik dokter baru
Manajemen kamar	Penambahan & pembaruan status kamar yang tersedia
Knowledge Base	Melatih kapabilitas chatbot
Laporan	Melihat dan mencari laporan jadwal dokter dan obat

B. Pengujian pada Sisi Pengunjung Publik (Pasien / Masyarakat)

Pengguna pada level ini memiliki keleluasaan untuk mengeksplorasi layanan transparansi informasi medis secara mandiri dan berinteraksi dengan asisten virtual kapan saja. Berikut modul pengujiannya di lampirkan pada *Tabel 4.2*.

Tabel 4. 2 Testing Pengunjung Publik

Modul Uji	Skenario Pengujian
Navigasi Utama	Pemuatan elemen Landing Page
Pencarian Jadwal	Pemfilteran jadwal dokter secara dinamis
Pencarian Obat	Verifikasi ketersediaan stok farmasi
Portal Berita	Transisi antarhalaman daftar berita
Kamar Rawat	Melihat kamar rawat yang tersedia
AI Chatbot	Pengujian kapabilitas penjawab interaktif
AI Chatbot	Pencegahan pengiriman pesan kosong

A R - R A N I R Y

4.4 Hasil Pengujian Google Lighthouse

Pengujian Google Lighthouse merupakan tahapan evaluasi teknis yang dilakukan untuk mengukur kualitas, performa, dan kapabilitas arsitektur website pelayanan publik Rumah Sakit Bhayangkara Banda Aceh. Sesuai dengan standar pengujian yang telah ditetapkan pada Bab III, alat audit otomatis ini digunakan untuk memvalidasi empat metrik utama pengembangan aplikasi web modern. Untuk menjamin akurasi, objektivitas, serta konsistensi data di tengah fluktuasi jaringan, pengujian dilakukan secara berulang sebanyak 10 kali eksekusi.

Adapun rincian evaluasi dari keempat metrik berdasarkan hasil pengujian berulang tersebut adalah sebagai berikut:

1. Kecepatan Rendering (*Performance*):

Metrik ini mengukur seberapa cepat antarmuka website termuat dan siap berinteraksi dengan pengguna. Pengujian *Performance* memvalidasi efisiensi kompilasi komponen React.js dan kecepatan transmisi data JSON melalui Inertia.js. Pengujian dilakukan 10 kali untuk memastikan halaman krusial seperti tabel Jadwal Dokter dan daftar Stok Obat tetap stabil dimuat dengan latensi minim meskipun terjadi fluktuasi kecepatan internet.

2. Tingkat Aksesibilitas Antarmuka (*Accessibility*):

Metrik ini mengevaluasi tingkat inklusivitas desain antarmuka bagi seluruh kalangan masyarakat. Pengujian memvalidasi elemen visual website seperti rasio kontras warna pada label status obat (Tersedia/Habis), ukuran tipografi artikel kesehatan, hingga atribut navigasi (*aria-labels*) pada tombol Widget Chatbot agar mudah diakses oleh pasien lanjut usia atau pengguna dengan gangguan penglihatan.

3. Keamanan dan Efisiensi Kode (*Best Practices*):

Metrik ini mengukur kepatuhan aplikasi terhadap standar pengembangan dan keamanan web global. Pengujian ini secara spesifik memvalidasi bahwa lalu lintas data pada *backend* Laravel berjalan secara aman dan bebas dari *error* sintaksis.

4. Visibilitas pada Mesin Telusur (*SEO*):

Metrik *Search Engine Optimization* (SEO) menguji seberapa baik struktur HTML sistem dirancang agar mudah diindeks oleh mesin pencari seperti

Google melalui validasi *meta-tags*, deskripsi halaman, serta format tautan (*slug*) pada portal berita dan *landing page*.

Rincian hasil pengujian Google Lighthouse untuk setiap kali eksekusi (1x hingga 10x) dapat dilihat pada Tabel 4.3 berikut.

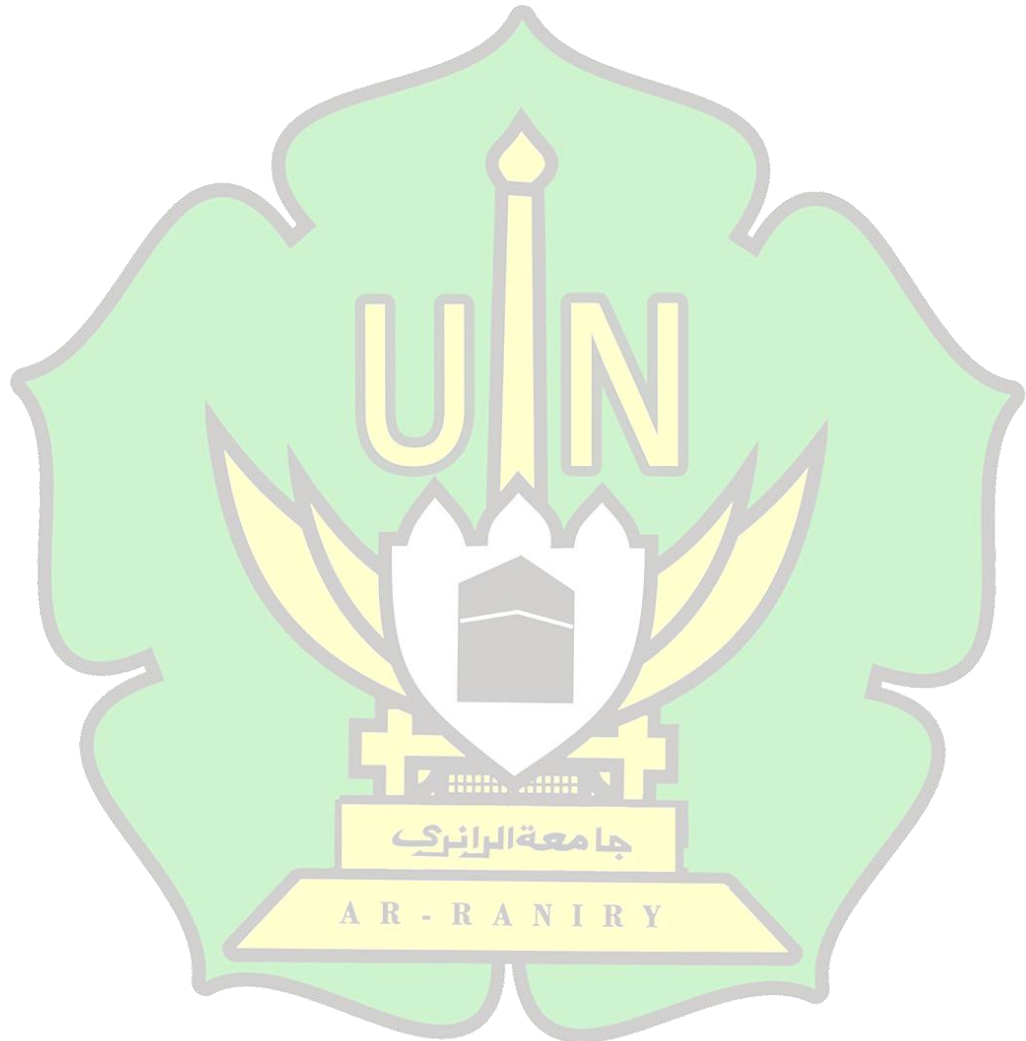
Tabel 4. 3 Hasil Pengujian Google Lighthouse (10 Kali Pengujian)

Pengujian Ke-	Performance	Accessibility	Best Practices	SEO
1	91	85	100	92
2	89	85	100	92
3	89	85	100	92
4	90	85	100	92
5	89	85	100	92
6	89	85	100	92
7	87	85	100	92
8	90	85	100	92
9	91	85	100	92
10	89	85	100	92
Rentang Skor	87-91	85	100	92
Rata-Rata	89.4	85.0	100.0	92.0

Hasil pengujian komprehensif dari 10 kali pengujian pada Tabel 4.3 menunjukkan bahwa arsitektur *Modern Monolith* pada sistem informasi RS Bhayangkara Banda Aceh berhasil melampaui seluruh target kelayakan secara konsisten. Pencapaian rata-rata skor sempurna (100) pada metrik *Best Practices* membuktikan bahwa penulisan kode sangat efisien dan konsisten memenuhi standar keamanan dalam mengelola data medis publik. Hal ini sejalan dengan perolehan rata-rata skor *Performance* yang tinggi (89.4) dengan rentang nilai(87-91), mengonfirmasi bahwa kombinasi React.js dan Inertia.js sukses mereduksi beban komputasi secara stabil.

Selanjutnya, pada aspek visibilitas dan inklusivitas, sistem juga mencatatkan rata-rata skor yang sangat stabil yaitu sebesar 92.0 untuk SEO dan 85.0 untuk

Accessibility. Stabilitas angka dari 10 kali pengujian ini mengonfirmasi bahwa antarmuka rumah sakit dirancang secara matang agar mudah diindeks oleh algoritma pencarian Google, sekaligus tetap ramah dan terbaca jelas oleh berbagai kalangan masyarakat. Secara akumulatif, seluruh perolehan metrik teknis Google Lighthouse ini membuktikan bahwa perangkat lunak yang dibangun berstatus **LAYAK** untuk beroperasi sebagai portal utama pelayanan publik RS Bhayangkara Banda Aceh.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada sistem informasi Rumah Sakit Bhayangkara Banda Aceh, maka dapat ditarik kesimpulan sebagai berikut:

1. **Pengembangan Sistem Informasi Layanan Publik yang Responsif:**

Penelitian ini telah berhasil membangun sebuah platform informasi pelayanan publik berbasis *web* yang responsif melalui penerapan arsitektur *Modern Monolith* menggunakan kombinasi *framework* Laravel 12, Inertia.js, dan React.js. Hasil pengujian menunjukkan bahwa sistem yang dikembangkan dapat menyajikan data dinamis seperti jadwal praktik dokter dan status ketersediaan obat secara baik.

2. **Evaluasi Kualitas Sistem:** Berdasarkan pengujian *Black Box*, seluruh fungsi pada modul *Administrator* maupun Pengunjung Publik terbukti berjalan valid dan sesuai rancangan. Ditambah dengan pengujian teknis berulang sebanyak 10 kali yang melampaui ambang batas minimal mencakup rata-rata *Performance* (89,4), *Accessibility* (85,0), *SEO* (92,0), serta *Best Practices* (100,0) sistem ini dinyatakan **layak** digunakan untuk mendukung layanan publik digital.

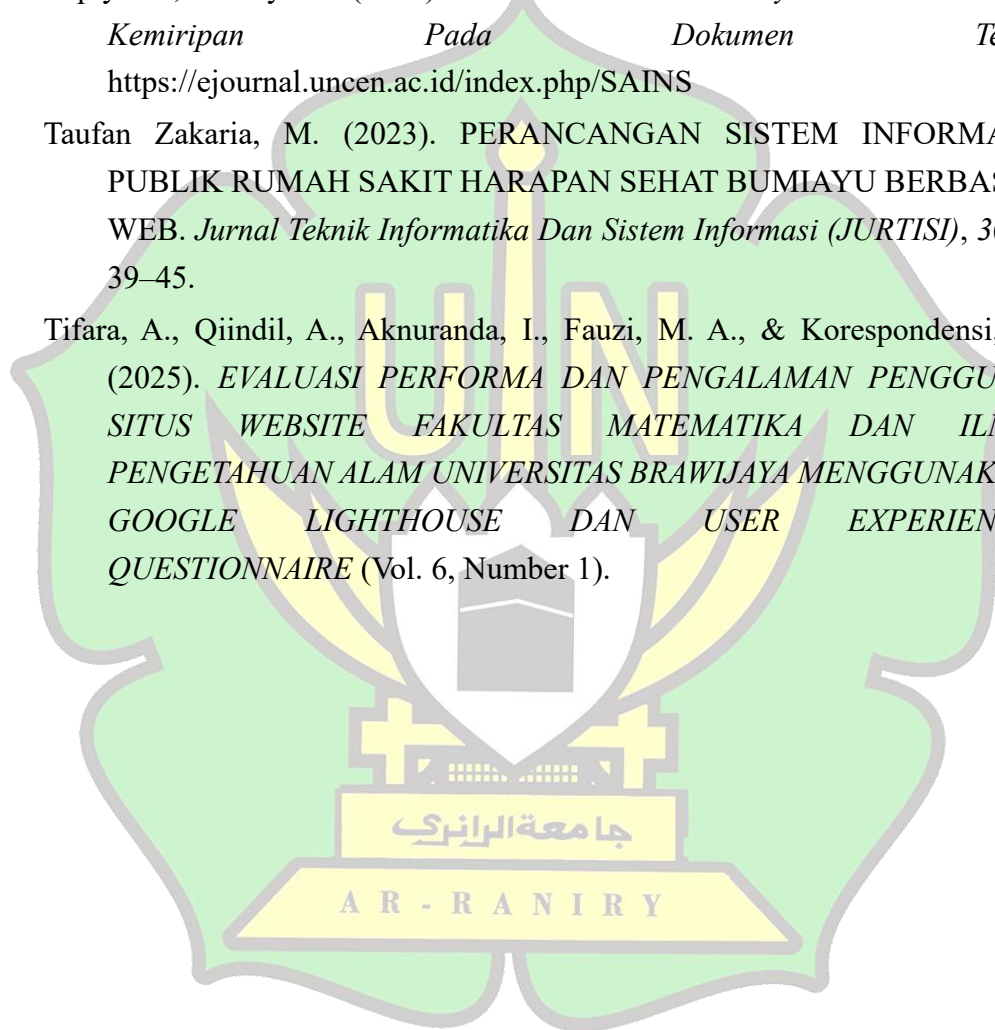
5.2 Saran

Untuk pengembangan sistem di masa mendatang, disarankan untuk menambahkan fitur pendaftaran dan pengambilan nomor antrean poliklinik secara *online*, mengintegrasikan platform web dan *AI Chatbot* secara langsung dengan SIMRS internal agar pembaruan data medis tersinkronisasi secara otomatis (*real-time*).

DAFTAR PUSTAKA

- Ardiansyah, M., & Lunandi, A. (2021). *Perancangan Dan Analisa Website Rumah Sakit Graha Hermine Batam* (Vol. 1, Number 1). <https://journal.uib.ac.id/index.php/combines>
- Bugaeva, I., & Shustova, M. (2025). Landing page in medical sphere as an effective means of influence over the target audience. *Communication Studies*, 389. [https://doi.org/10.24147/2413-6182.2025.12\(2\).389-405](https://doi.org/10.24147/2413-6182.2025.12(2).389-405)
- Daffa Al-farel, M., & Rizal Dzikrillah, A. (2025). Evaluasi Framework Pengembangan Web KinarBhusana: Studi Performance, SEO, dan Accessibility Menggunakan Laravel dan Bootstrap. *Bulletin of Computer Science Research*, 5(3), 235–242. <https://doi.org/10.47065/bulletincsr.v5i3.499>
- Dathrao, B., Ashutosh, B., Madhu, Chathurya, Y., & Kiran, U. (2025). *Intelligent ChatBot For Enhanced College Website Interaction*.
- Filza Maharani, N., Nugraha, M. A., Aldiansyah, M., & Pibriana, D. (2024). *Analisis Metode Dan Bidang Pengembangan Sistem Informasi Menggunakan Systematic Literature Review Analysis of Methods and Areas of Information System Development using Systematic Literature Review*. 5(1), 45–56.
- Google Developers. (n.d.). *Pemberian skor performa Lighthouse*. Web.Dev. Retrieved January 21, 2026, from <https://developer.chrome.com/docs/lighthouse/performance/performance-scoring?hl=id>
- Huberta, B., & Wijaya, A. B. (2023). PERANCANGAN CHATBOT WEBSITE PROGRAM STUDI INFORMATIKA MENGGUNAKAN FRAMEWORK CODEIGNITER. *Jurnal Informatika Dan Teknik Elektro Terapan*, 11(3). <https://doi.org/10.23960/jitet.v11i3.3225>
- Keertthi G, A., & M, Y. (n.d.). *Website with AI chatbot oriented around*.
- Rumkit, B. (2023). *DASAR HUKUM*.
- Salamudin, S., & Meilantika, D. (2023). Perancangan Website Rumah Sakit dr.Noesmir Baturaja Menggunakan Metode User Centered Design (UCD). *Jurnal Ilmiah FIFO*, 14(2), 193. <https://doi.org/10.22441/fifo.2022.v14i2.009>

- Sanjaya, T. (2021). *Perancangan Dan Implementasi Company Profile Berbasis Website Sebagai Media Promosi di PT. Hassani Can Packaging* (Vol. 1, Number 1). <https://journal.uib.ac.id/index.php/conescintech>
- Setiawan, A., & Yamasari, Y. (2025). Evaluasi Performa Website Sistem Informasi Kesejahteraan Sosial (SIKS-NG) Di Kelurahan Lembeyan Kulon Kabupaten Magetan Dengan Google Lighthouse. *Journal of Informatics and Computer Science*, 07. <https://siks.kemensos.go.id>
- Supiyanto, & Sriyono. (2023). *Metode Cosine Similarity Untuk Mendeteksi Kemiripan Pada Dokumen Teks*. <https://ejournal.uncen.ac.id/index.php/SAINS>
- Taufan Zakaria, M. (2023). PERANCANGAN SISTEM INFORMASI PUBLIK RUMAH SAKIT HARAPAN SEHAT BUMIAYU BERBASIS WEB. *Jurnal Teknik Informatika Dan Sistem Informasi (JURTISI)*, 3(1), 39–45.
- Tifara, A., Qiindil, A., Aknuranda, I., Fauzi, M. A., & Korespondensi, P. (2025). *EVALUASI PERFORMA DAN PENGALAMAN PENGGUNA SITUS WEBSITE FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS BRAWIJAYA MENGGUNAKAN GOOGLE LIGHTHOUSE DAN USER EXPERIENCE QUESTIONNAIRE* (Vol. 6, Number 1).



LAMPIRAN

HASIL PENGUJIAN BLACKBOX TESTING

1. Hasil Pengujian pada Otoritas Administrator (Staf / IT RS Bhayangkara)

Akun pengujian : rara@gmail.com | Password

Tabel hasil pengujian

Modul Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Status
Auth	Uji coba akses masuk (Login)	Memasukkan email dan kata sandi yang terdaftar pada form login, lalu menekan tombol "Masuk".	Sistem memvalidasi kredensial, berhasil otentikasi, dan mengarahkan admin ke halaman Dashboard.	Valid
Auth	Uji coba pengakhiran sesi (Logout)	Menekan opsi/tombol "Keluar" (Logout) pada menu profil administrator.	Sesi aktif dihapus secara aman dan sistem mengembalikan tampilan ke halaman login.	Valid
Dashboard	Validasi visualisasi data analitik	Membuka halaman Dashboard untuk memantau ringkasan statistik sistem.	Metrik angka pengunjung, jumlah artikel, jadwal dokter, serta grafik statistik tampil presisi sesuai database.	Valid
Manajemen Artikel	Penambahan publikasi edukasi baru	Mengisi formulir artikel baru (judul, slug, gambar, isi konten) lalu	Data artikel berhasil tersimpan di database dan tampil pada daftar	Valid

		menekan tombol simpan.	artikel serta halaman portal publik.	
Manajemen Obat	Penambahan & pembaruan status ketersediaan farmasi	Menginput data obat baru serta mengubah toggle status ketersediaan obat (Tersedia/Habis).	Data obat baru tersimpan dan pembaruan status ketersediaan langsung ter-update secara real-time.	Valid
Manajemen Dokter	Penambahan nama-nama dokter	Mengisi formulir data master dokter (nama lengkap dan spesialisasi) lalu menekan tombol simpan.	Data dokter berhasil ditambahkan ke dalam database dan siap dihubungkan dengan jadwal praktik.	Valid
Manajemen Jadwal Dokter	Pendaftaran jadwal praktik dokter baru	Memilih nama dokter, menentukan hari, jam mulai, dan jam selesai praktik, lalu menyimpan data.	Jadwal praktik dokter berhasil ditambahkan dan langsung muncul pada papan jadwal dokter publik.	Valid
Manajemen Kamar	Penambahan & pembaruan status kamar yang tersedia	Menginput kamar baru serta memperbarui kapasitas/sisa bed kosong dan status BPJS.	Data kamar tersimpan dan informasi sisa bed yang tersedia ter-update secara presisi.	Valid
Knowledge Base	Melatih kapabilitas chatbot	Menginput materi FAQ medis/layanan rumah sakit baru pada modul	Teks tersimpan dan otomatis dikonversi menjadi vektor embedding untuk	Valid

		Knowledge Base dan menyimpannya.	acuan jawaban AI Chatbot RAG.	
Laporan	Melihat dan mencari laporan jadwal dokter dan obat	Membuka modul laporan, lalu menggunakan fitur pencarian/filter untuk menyaring data dokter dan obat.	Sistem menampilkan daftar laporan data jadwal dokter dan stok obat yang relevan sesuai kata kunci pencarian.	Valid

2. Hasil Pengujian pada Sisi Pengunjung Publik (Pasien / Masyarakat)

Tabel hasil pengujian

Modul Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Status
Navigasi Utama	Pemuatan elemen Landing Page	Mengakses domain utama website RS Bhayangkara melalui peramban (<i>browser</i>).	Seluruh antarmuka termuat sempurna, menampilkan akses cepat layanan, jadwal harian, artikel terbaru, beserta widget <i>chatbot</i> di sudut layar.	Valid
Pencarian Jadwal	Pemfilteran jadwal dokter secara dinamis	Mengetikkan nama dokter tertentu atau menyortir berdasarkan kategori poliklinik	Daftar dokter dan waktu praktik yang tampil tersaring secara dinamis dan asinkron tanpa memuat ulang	Valid

		pada kolom pencarian.	(<i>reload</i>) halaman penuh.	
Pencarian Obat	Verifikasi ketersediaan stok farmasi	Mengetikkan nama obat spesifik pada kolom pencarian inventaris obat-obatan.	Sistem merespons dengan menampilkan nama obat yang relevan beserta deskripsi dan indikator status ketersediaannya (Tersedia/Habis).	Valid
Portal Berita	Transisi antarhalaman daftar berita	Menekan tombol nomor halaman (<i>pagination</i>) atau memilih salah satu kartu berita/artikel kesehatan.	Transisi dan perpindahan halaman artikel berjalan sangat responsif dan lancar didukung oleh protokol Inertia.js.	Valid
Kamar Rawat	Melihat kamar rawat yang tersedia	Membuka halaman/bagian informasi kamar rawat inap pada website.	Sistem menampilkan daftar nama ruangan, kelas perawatan, kapasitas bed, sisa bed kosong yang tersedia, serta indikator status dukungan BPJS.	Valid
AI Chatbot	Pengujian kapabilitas penjawab interaktif	Mengirimkan kueri acak terkait informasi rumah sakit (contoh: "apakah ada kamar BPJS kosong?") ke dalam jendela <i>chat</i> .	Sistem melakukan komputasi pencarian data (RAG/Cosine Similarity) dan membalas dengan rangkaian kalimat natural yang akurat dan informatif.	Valid

AI Chatbot	Pencegahan pengiriman pesan kosong	Berusaha menekan tombol kirim atau menekan <i>Enter</i> ketika kolom input <i>chat</i> sama sekali tidak berisi teks.	Tombol kirim tidak aktif (<i>disabled</i>) atau sistem menolak aksi tersebut guna mencegah kesalahan pemrosesan komputasi <i>embedding</i> pada <i>backend</i> .	Valid
-------------------	------------------------------------	---	--	-------

