



PERANCANGAN DAN IMPLEMENTASI SISTEM RESERVASI KLINIK BERBASIS WEB DENGAN *FRAMEWORK* REACT JS: STUDI KASUS KLINIK PRATAMA AGHA DEPOK

Ammar Farras Hudzaifah¹, Maulana Fakhri Latief²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
amma21222ti@student.nurulfikri.ac.id, maulana.latief@nurulfikri.ac.id

Abstract

Digital transformation in the healthcare sector presents opportunities to improve service efficiency, especially in patient reservation processes. This study aims to develop a web-based doctor consultation reservation information system using the React JS framework at Klinik Pratama Agha Depok. The research method applies the Software Development Life Cycle (SDLC) approach through stages of requirement analysis, system design, implementation, and testing. Data were collected through observation, interviews with clinic staff, and documentation. System testing was conducted using Black Box Testing to assess system functions and User Acceptance Testing (UAT) involving five user respondents. The test results showed that all system features functioned as expected, with an average UAT score of 96%, indicating a high level of user satisfaction. The system enables patients to register independently, select consultation schedules, and receive automatic notifications. The study concludes that implementing an online reservation system improves service efficiency and reduces physical queues in clinics. This research recommends integrating a WhatsApp-based notification feature to enhance patient-system connectivity.

Keywords: Black Box, Reservation System, SDLC, UAT, Website

Abstrak

Transformasi digital dalam bidang kesehatan membuka peluang untuk meningkatkan efisiensi pelayanan, khususnya dalam proses reservasi pasien. Penelitian ini bertujuan mengembangkan sistem reservasi konsultasi dokter berbasis *website* menggunakan *framework* React JS pada Klinik Pratama Agha Depok. Metode penelitian menggunakan pendekatan *Software Development Life Cycle* (SDLC) dengan tahapan analisis kebutuhan, perancangan sistem, implementasi, serta pengujian. Data dikumpulkan melalui observasi, wawancara dengan pihak klinik, dan dokumentasi. Pengujian sistem dilakukan dengan *Black Box Testing* untuk menguji fungsi sistem dan *User Acceptance Testing* (UAT) yang melibatkan lima responden pengguna sistem. Hasil pengujian menunjukkan bahwa seluruh fungsi sistem berjalan sesuai harapan dan memperoleh rata-rata skor UAT sebesar 96%, yang mengindikasikan tingkat kepuasan pengguna sangat tinggi. Sistem yang dibangun memungkinkan pasien melakukan pendaftaran secara mandiri, memilih jadwal konsultasi, serta menerima notifikasi otomatis. Kesimpulan dari penelitian ini adalah bahwa implementasi sistem reservasi *online* dapat meningkatkan efisiensi pelayanan dan mengurangi antrean fisik di klinik. Rekomendasi dari penelitian ini adalah pengembangan fitur notifikasi berbasis WhatsApp untuk meningkatkan keterhubungan pasien dengan sistem.

Kata kunci: Black Box, SDLC, Sistem Reservasi, UAT, Website

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah mendorong transformasi digital di berbagai sektor, termasuk sektor kesehatan. Salah satu tantangan utama dalam pelayanan kesehatan adalah pengelolaan antrean pasien yang efektif, yang secara langsung berpengaruh terhadap tingkat kepuasan pasien dan kualitas layanan yang diberikan [1]. Lamanya waktu tunggu pasien sering kali

disebabkan oleh berbagai faktor, seperti jumlah pasien yang melebihi kapasitas, ketersediaan tenaga medis, proses administrasi yang belum optimal, serta sistem pendaftaran yang masih manual [2]. Permasalahan ini juga dialami oleh Klinik Pratama Agha Depok yang hingga saat ini masih menerapkan sistem reservasi konvensional.

Seiring dengan meningkatnya jumlah pasien, kebutuhan akan sistem reservasi *online* menjadi semakin penting untuk mengurangi antrean fisik, mempercepat proses pendaftaran, serta meningkatkan efisiensi pelayanan. Sistem reservasi *online* memungkinkan pasien melakukan pendaftaran secara mandiri, memilih jadwal konsultasi, dan menerima notifikasi secara otomatis terkait waktu konsultasi [3]. Dengan demikian, sistem ini diharapkan mampu meningkatkan kenyamanan pasien sekaligus mempermudah pengelolaan administrasi oleh pihak klinik.

Dalam pengembangan sistem reservasi *online*, pemilihan teknologi yang tepat menjadi faktor kunci keberhasilan implementasi. React JS, sebagai salah satu *framework frontend* modern, menawarkan berbagai keunggulan, seperti arsitektur berbasis komponen, performa aplikasi yang responsif, serta dukungan ekosistem yang luas [4].

Penelitian ini bertujuan untuk merancang sistem pendaftaran konsultasi dokter berbasis web dengan React JS, mengevaluasi fungsionalitasnya, serta menganalisis efektivitas sistem dalam meningkatkan efisiensi layanan. Pengembangan sistem dilakukan dengan pendekatan *Software Development Life Cycle* (SDLC), serta pengujian fungsionalitas sistem menggunakan metode *Black Box Testing* dan *User Acceptance Testing* (UAT). Diharapkan, implementasi sistem ini dapat meningkatkan efisiensi pelayanan klinik, meminimalkan waktu tunggu, serta memberikan pengalaman pelayanan kesehatan yang lebih optimal bagi pasien.

React JS

React JS dalam bahasa Indonesia disebut React JS juga. Ini adalah *library* JavaScript yang digunakan untuk mengembangkan antarmuka pengguna (UI) yang interaktif dan responsif [5]. React JS membuat proses pembuatan UI interaktif menjadi lebih mudah dengan menggunakan konsep komponen. Komponen adalah unit terkecil dari UI yang dapat di-*render* dan dimanipulasi secara independen. React JS juga menggunakan konsep *state* untuk mengelola data yang digunakan untuk *render* UI. *State* adalah data yang dapat berubah selama *runtime website*. React JS melacak perubahan *state* dan secara otomatis me-*render* UI saat *state* berubah. React JS adalah *library* yang sangat populer dan banyak digunakan untuk mengembangkan situs web yang kompleks dan interaktif [6]. Beberapa contoh *website* yang dibangun menggunakan React JS adalah Facebook, Instagram, dan Netflix.

Mock API

Mock API dalam bahasa Indonesia disebut Mock API juga. Ini adalah server API palsu yang dapat digunakan untuk menguji *website* web [7]. Mock API memungkinkan Anda untuk memanipulasi data yang dikembalikan oleh API, sehingga Anda dapat menguji *website* Anda dengan data yang berbeda-beda.

SDLC (*System Development Life Cycle*)

Dalam bahasa Inggris, "metodologi siklus hidup pengembangan perangkat lunak" (SDLC) adalah singkatan dari "metodologi siklus hidup pengembangan perangkat lunak". Ini adalah istilah yang mengacu pada metode atau pendekatan yang digunakan untuk membuat dan mengubah sistem, baik sistem komputer maupun sistem informasi, untuk memastikan bahwa mereka efektif, efisien, dan terus beroperasi [8].

Black Box Testing

Black Box Testing adalah teknik pengujian perangkat lunak yang tidak memeriksa struktur internal atau algoritma yang digunakan, tetapi memeriksa fungsionalitas sistem berdasarkan *input* dan *output*-nya. Metode ini dilakukan dengan memberikan *input* ke dalam sistem dan kemudian memeriksa *output*-nya untuk memastikan bahwa sesuai dengan yang diharapkan [9].

User Acceptance Testing

Pengujian perangkat lunak untuk penerimaan pengguna (UAT) dilakukan oleh pengguna akhir atau perwakilan bisnis untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi persyaratan yang telah ditetapkan. UAT dilakukan setelah pengujian fungsional, integrasi, dan sistem, serta menjadi tahap akhir sebelum perangkat lunak dipindahkan ke lingkungan produksi atau dirilis ke publik [10].

Beberapa penelitian sebelumnya telah dilakukan terkait pengembangan sistem informasi di bidang kesehatan. Maolana mengembangkan aplikasi rekam medis di Imran Medical Center menggunakan *framework* React JS dengan metode *Prototype*. Aplikasi ini mempermudah pengelolaan data pasien, riwayat pemeriksaan, serta proses pendaftaran pengobatan, sehingga menggantikan sistem pencatatan manual berbasis kertas yang sebelumnya digunakan [11].

Penelitian lainnya oleh Anggrainy merancang sistem informasi pelanggan pasien di Rumah Sakit Jiwa Aceh. Sistem ini memanfaatkan *framework* CodeIgniter dan React JS untuk meningkatkan efisiensi pengelolaan data pasien, sehingga mempercepat proses administrasi dan pelayanan [12].

Selanjutnya, Putri menganalisis efektivitas implementasi sistem reservasi *online* pasien rawat jalan di Poli Jantung Rumah Sakit Tk II Udayana. Hasil uji *usability* menunjukkan bahwa sistem ini memperoleh skor 78,90%, yang dikategorikan layak untuk digunakan (*eligible*) [13].

Molly juga melakukan penelitian mengenai pemanfaatan Sistem Informasi Manajemen Rumah Sakit (SIMRS) di RSUD DOK II Jayapura. Hasilnya menunjukkan bahwa penerapan SIMRS mampu meningkatkan efisiensi administrasi serta mengintegrasikan layanan rumah sakit secara lebih efektif [14].

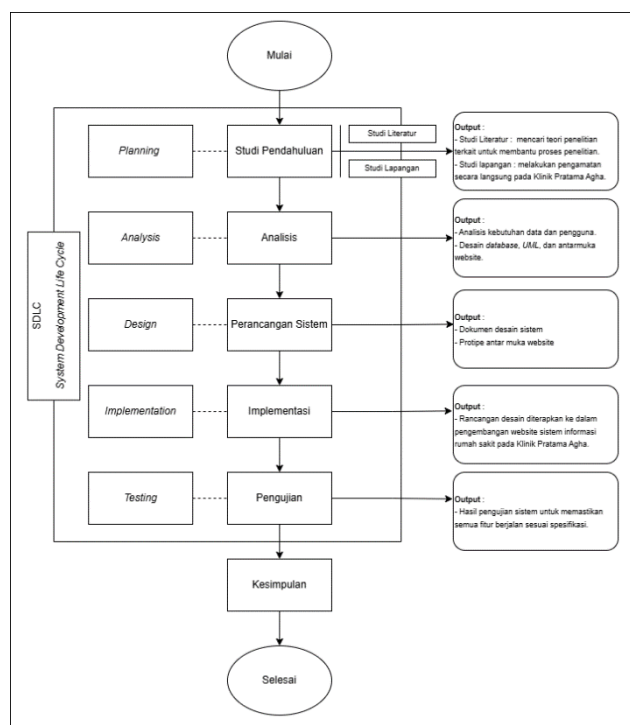
Sementara itu, Sekarang mengembangkan aplikasi informasi kunjungan harian pasien di salah satu rumah sakit swasta di Kota Denpasar dengan menggunakan *framework* Laravel. Sistem ini berhasil mempermudah proses *monitoring* kunjungan pasien, meningkatkan efisiensi pencatatan data, meminimalkan kesalahan rekap data, serta mendukung pengambilan keputusan manajemen rumah sakit [15].

Dari sejumlah penelitian tersebut, dapat disimpulkan bahwa pengembangan sistem informasi berbasis *web* untuk pelayanan kesehatan telah banyak memberikan dampak positif, terutama dalam peningkatan efisiensi pelayanan administrasi, pengelolaan data pasien, serta pengurangan waktu tunggu. Namun, penelitian-penelitian sebelumnya sebagian besar masih menggunakan *framework* berbasis *server-side* seperti Laravel dan CodeIgniter, serta belum secara optimal memanfaatkan *framework modern* berbasis komponen seperti React JS yang memiliki keunggulan dalam pengembangan antarmuka pengguna yang dinamis dan responsif. Oleh karena itu, penelitian ini berupaya mengisi celah tersebut dengan mengembangkan sistem reservasi konsultasi dokter berbasis *web* menggunakan *framework* React JS, serta mengintegrasikan pengujian *Black Box Testing* dan *User Acceptance Testing* untuk memastikan kualitas sistem.

2. METODE PENELITIAN

2.1. Tahapan Penelitian

Pada Gambar 1 merupakan rangkaian dari tahapan penelitian yang berawal dari studi pendahuluan, analisis, implementasi, pengujian, hingga kesimpulan.



Gambar 1. Tahapan Penelitian

2.1.1. Studi Pendahuluan

Tahap pertama dalam penelitian ini adalah studi pendahuluan yang dilakukan untuk memahami permasalahan dan ruang lingkup penelitian. Studi pendahuluan meliputi pengumpulan informasi terkait Klinik Pratama Agha Depok dan teknologi yang akan digunakan, seperti React JS, Mock API, serta metode pengembangan sistem. Studi pendahuluan ini dilakukan melalui berbagai metode, seperti studi literatur untuk memahami teori dan konsep yang relevan, observasi langsung di lokasi klinik untuk mengetahui permasalahan yang dihadapi dalam sistem administrasi saat ini, serta diskusi awal dengan pemilik klinik terkait kebutuhan dan ekspektasi terhadap sistem informasi yang akan dikembangkan.

2.1.2. Analisis

Tahap analisis bertujuan untuk mengidentifikasi kebutuhan sistem pada situs *web* Klinik Pratama Agha Depok. Proses ini melibatkan analisis kebutuhan fungsional dan non-fungsional, pembuatan desain *database*, perancangan arsitektur sistem, serta diagram UML seperti *use case diagram*, *class diagram*, *sequence diagram*, dan *activity diagram*. Analisis kebutuhan dilakukan secara menyeluruh untuk memastikan bahwa setiap fitur yang dikembangkan benar-benar relevan dan sesuai dengan kebutuhan klinik. Kebutuhan fungsional meliputi fitur utama seperti sistem registrasi pasien, jadwal konsultasi dokter, dan pelacakan status pendaftaran, sedangkan kebutuhan non-fungsional meliputi keamanan data, kecepatan akses, dan kemudahan penggunaan.

2.1.3. Perancangan Sistem

Perancangan sistem bertujuan untuk menerjemahkan hasil analisis ke dalam desain teknis yang akan digunakan dalam pengembangan sistem reservasi. Proses ini mencakup beberapa tahapan, yaitu perancangan arsitektur sistem dengan menggunakan React JS untuk *frontend* dan Mock API sebagai simulasi *backend*, perancangan *database* untuk mengatur alur data dan hubungan antar entitas, pembuatan prototipe antarmuka pengguna (*UI*) yang meliputi halaman *login*, registrasi, konsultasi, dan pengecekan status pendaftaran, serta pembuatan *diagram UML* seperti *use case diagram* dan *activity diagram* untuk memvisualisasikan proses yang terjadi dalam sistem. Perancangan sistem ini menjadi fondasi penting agar proses implementasi dapat berjalan sesuai dengan rencana dan target yang telah ditetapkan.

2.1.4. Implementasi

Implementasi meliputi pengembangan situs *web* menggunakan React JS untuk *frontend* dan Mock API sebagai simulasi *backend*. Pada tahap ini, dilakukan pembuatan fitur-fitur seperti halaman *login*, register, daftar konsultasi, dan pengecekan status pendaftaran. Kode program ditulis secara iteratif dengan mengadopsi prinsip-

prinsip dari SDLC. Proses implementasi juga mencakup konfigurasi perangkat lunak, pembuatan komponen-komponen utama antarmuka pengguna (UI), serta koneksi antara *frontend* dan Mock API untuk memastikan integrasi berjalan lancar. Setiap fitur diuji secara bertahap sebelum masuk ke tahap pengujian menyeluruh.

2.1.5. Pengujian

Proses pengujian bertujuan untuk memastikan bahwa sistem berfungsi sesuai kebutuhan. Pengujian dilakukan menggunakan metode Black Box Testing untuk memeriksa fungsionalitas tanpa melihat kode internal dan memastikan bahwa *input* dan *output* sesuai dengan spesifikasi. Selain itu, dilakukan *User Acceptance Testing* (UAT) dengan melibatkan pengguna klinik untuk mendapatkan umpan balik.

2.1.6. Kesimpulan

Tahapan akhir adalah kesimpulan yang berisi evaluasi dari implementasi sistem, hasil pengujian, dan efektivitas solusi yang diterapkan. Pada tahap ini juga diberikan saran untuk pengembangan lebih lanjut agar sistem informasi di Klinik Pratama Agha Depok terus berkembang sesuai kebutuhan pengguna.

2.2. Rancangan Penelitian

Penelitian ini termasuk dalam kategori *Research and Development* (R&D) yang bertujuan untuk mengembangkan produk berupa sistem informasi berbasis web. Fokus utama penelitian ini adalah membangun situs web Klinik Pratama Agha Depok yang memuat fitur konsultasi *online* dan manajemen data klinik. Proses R&D melibatkan identifikasi masalah, analisis kebutuhan, pengembangan sistem, pengujian, dan evaluasi. R&D dipilih karena metode ini memungkinkan proses pengembangan yang iteratif dan adaptif, di mana setiap tahap pengembangan didasarkan pada data empiris dan umpan balik pengguna. Dengan demikian, sistem yang dihasilkan akan relevan, efektif, dan sesuai dengan kebutuhan Klinik Pratama Agha Depok.

3. HASIL DAN PEMBAHASAN

3.1. Analisis Kebutuhan

Analisis merupakan tahapan awal dalam melakukan perancangan sebuah sistem. Analisis dilakukan dengan tujuan untuk memperoleh solusi dan kebutuhan apa saja yang dibutuhkan dalam membangun sebuah *website*. Analisis ini dibutuhkan sebagai dasar perancangan *website*. Analisis Kebutuhan Sistem Pendaftaran untuk Klinik Pratama Agha Depok:

a) Pelayanan Umum: Sistem pendaftaran harus menyediakan layanan umum untuk pasien. Memastikan adanya formulir registrasi *online* untuk pasien baru dan pembaruan informasi bagi pasien yang sudah terdaftar.

b) Pelayanan Kandungan: Jadwal dokter kandungan harus terintegrasi dalam sistem pendaftaran. Pasien dapat memilih waktu konsultasi kandungan sesuai ketersediaan dokter.

c) Pelayanan Laboratorium: Sistem harus memfasilitasi reservasi tes laboratorium secara *online*. Pasien dapat memilih jadwal pemeriksaan tes laboratorium. Dengan memperhatikan elemen-elemen di atas, implementasi sistem pendaftaran yang komprehensif akan membantu meningkatkan efisiensi, akurasi, dan pengalaman pasien di Klinik Pratama Agha Depok.

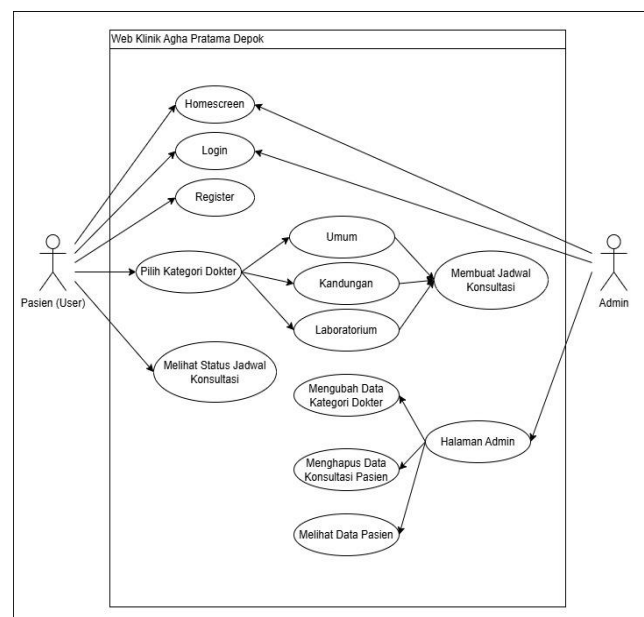
Dengan memperhatikan elemen-elemen di atas, implementasi sistem pendaftaran yang komprehensif akan membantu meningkatkan efisiensi, akurasi, dan pengalaman pasien di Klinik Pratama Agha Depok.

3.2. Perancangan Website

Perancangan *website* ini melibatkan pengembangan tampilan dan alur sistem dengan fokus pada struktur navigasi, *use case diagram*, *activity diagram*, rancangan *database*, dan perancangan halaman *website*. Tujuan utamanya adalah menciptakan pengalaman pengguna yang intuitif dan efisien.

3.2.1. Use Case Diagram

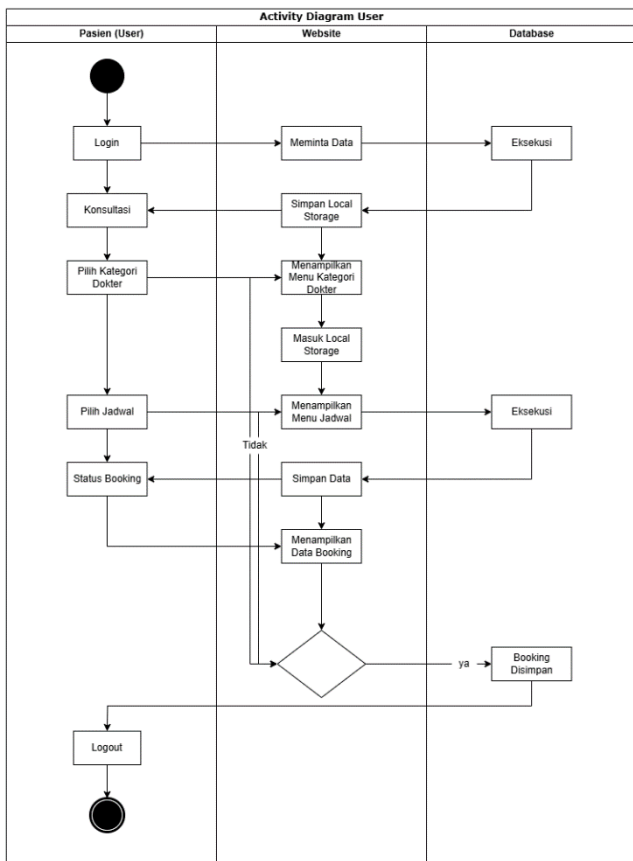
Use Case Diagram adalah hasil analisis perancangan sistem menggunakan pemodelan *use case diagram* yang merepresentasikan interaksi – interaksi antara *user* dan *admin* pada *use case diagram*. Berdasarkan Gambar 2 rancangan *use case diagram*, diagram memiliki 2 *actor* yaitu *User* dan *Admin*. Pada diagram di bawah *User* dan *Admin* dapat mengakses *register*, *login* dan daftar konsultasi dokter antrean pada *Website* Klinik Pratama Agha Depok.



Gambar 2. Use Case Diagram

3.2.2. Activity Diagram

Konsep dasar aplikasi melibatkan langkah-langkah seperti *login*, pemilihan dokter, penjadwalan antrean konsultasi, dan konfirmasi antrean konsultasi. Diagram aktivitas menggambarkan serangkaian proses, dimulai dari aktivitas *login* hingga konfirmasi antrean, serta mencakup interaksi *admin* dengan data kategori dokter, data daftar konsultasi dokter pasien, dan informasi akun pasien. Dengan demikian, bab ini memberikan gambaran komprehensif tentang struktur dan 26 interaksi dalam aplikasi, bertujuan untuk meningkatkan efisiensi dan pengalaman pengguna dalam proses pendaftaran dan antrean konsultasi di Klinik Pratama Agha Depok seperti Gambar 3 di bawah ini.



Gambar 3. Activity Diagram

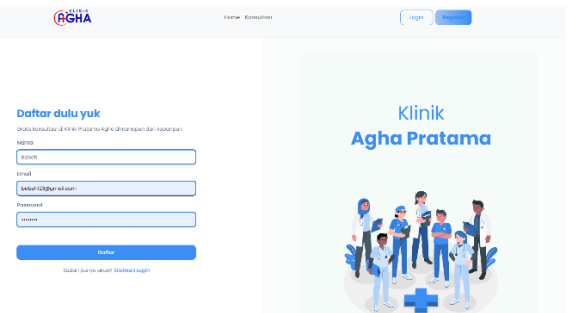
3.2.3. Rancangan Database

Pada tahap ini dilakukan perancangan basis data untuk memastikan seluruh kebutuhan informasi sistem reservasi Klinik Pratama Agha Depok dapat terakomodasi dengan baik. Rancangan basis data terdiri dari beberapa tabel utama. Tabel *User* memiliki *field* *id_user* (Integer, *Primary Key*), *user_name* (String, 30), *user_password* (String, 20), *user_email* (String, 20), dan *user_image_link* (String, 20). Tabel *Admin* terdiri dari *id_admin* (Integer, 3, *Primary Key*), *admin_name* (String, 30), *admin_password* (String, 20), *admin_email* (String, 20), dan *admin_image_link* (String, 20). Tabel *Kategori Dokter* mencakup *id_kategori* (Integer, 3, *Primary Key*), *nama_kategori* (String, 15), dan *image_link* (String, 20). Sementara itu, Tabel *Daftar*

Konsultasi Antrean Dokter mencakup *id_daftar_konsultasi_dokter* (Integer, 20, *Primary Key*), *id_user* (Integer, 3, *Foreign Key*), *id_kategori* (Integer, 3, *Foreign Key*), *waktu_konsultasi* (String, 15), dan *nomor_antrian* (Integer, 3). Perancangan tabel ini bertujuan untuk memastikan relasi antar entitas dalam sistem dapat berjalan dengan efektif dan efisien.

3.3 Implementasi Sistem

Implementasi sistem merupakan tahap di mana rancangan sistem yang telah dibuat sebelumnya mulai diterapkan dalam bentuk nyata, yaitu berupa *website* daftar konsultasi dokter. Pada penelitian ini, implementasi dilakukan menggunakan *framework* React JS sebagai teknologi utama dalam pembangunan antarmuka pengguna (*frontend*). Struktur implementasi sistem dimulai dengan pengembangan beberapa komponen utama dalam bentuk antarmuka pengguna berbasis React JS yang saling terintegrasi. Komponen pertama adalah halaman *login* dan registrasi, yang berfungsi sebagai pintu masuk pengguna ke dalam sistem. Pada halaman ini, pengguna melakukan proses *autentikasi* sebagai pasien. Data *login* yang berhasil akan disimpan dalam *session* dan digunakan sebagai referensi dalam proses pendaftaran konsultasi berikutnya seperti pada Gambar 4 .



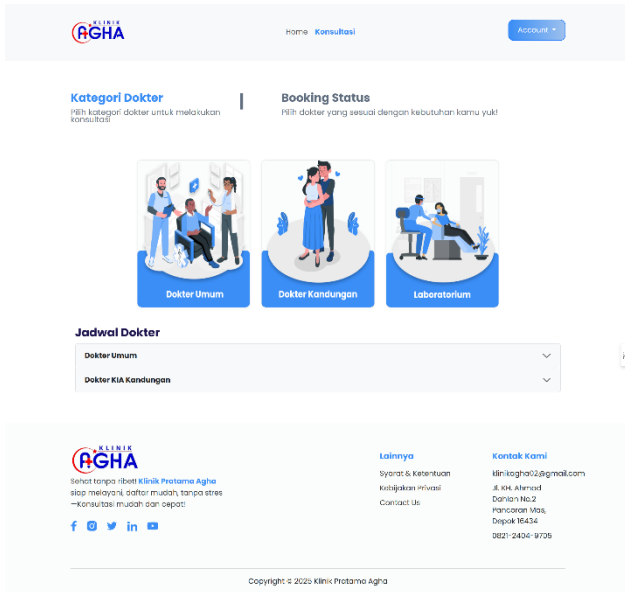
Gambar 4. Halaman Login dan Registrasi

Setelah berhasil masuk, pengguna diarahkan ke halaman beranda (*home*) yang menjadi pusat informasi utama sistem pada Gambar 5. Pada halaman ini, pengguna dapat mengakses fitur-fitur seperti melihat jadwal dokter, memilih kategori layanan, serta melanjutkan ke proses pendaftaran konsultasi.



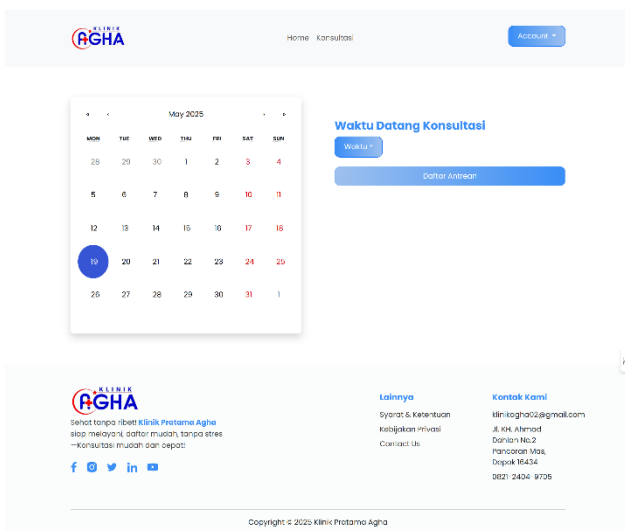
Gambar 5. Halaman Beranda

Langkah berikutnya pada Gambar 6 adalah mengakses halaman konsultasi dokter, di mana pengguna dapat memilih kategori dokter, memilih jadwal konsultasi, serta secara otomatis mendapatkan nomor antrean. Data yang di-input-kan pengguna akan dikirim ke *backend* simulasi menggunakan Mock API yang berfungsi sebagai penyimpanan sementara.



Gambar 6. Halaman Konsultasi Dokter

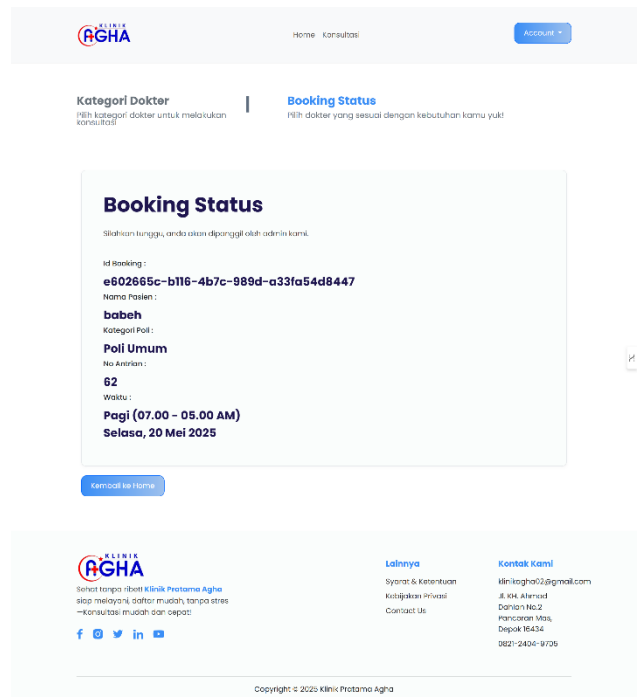
Kemudian, sistem akan menampilkan halaman pilih jadwal konsultasi dokter yang berisi detail informasi antrean pasien, seperti nomor antrean, waktu konsultasi, dan poli yang dipilih pada Gambar 7. Halaman ini memungkinkan pengguna untuk memantau posisi antrean secara *real time* dan memastikan kejelasan jadwal layanan yang dipilih.



Gambar 7. Halaman Pilih Jadwal Konsultasi Dokter

Terakhir, pada Gambar 8 terdapat halaman status registrasi konsultasi dokter yang memungkinkan pasien untuk melihat rekap hasil pendaftaran mereka. Informasi yang tersedia mencakup status antrean, jadwal yang dipilih, dan detail dokter atau layanan yang telah dipilih. Fitur ini membantu

pasien dalam memantau status konsultasi secara mandiri dan terstruktur tanpa harus kembali menghubungi pihak klinik.



Gambar 8. Halaman Status Registrasi Konsultasi Dokter

Struktur direktori React JS disusun secara modular berdasarkan prinsip *separation of concerns*, meliputi *folder components, pages, services*, dan *assets*. Komunikasi antara *frontend* dan *backend* menggunakan Axios untuk melakukan *request* HTTP ke *endpoint* Mock API. Implementasi juga dilengkapi dengan validasi *form* menggunakan *react-hook-form* untuk memastikan setiap *input* pengguna sesuai dengan format yang diharapkan.

3.4 Evaluasi Hasil

Setelah tahap implementasi sistem selesai dilakukan, evaluasi dilakukan guna mengukur sejauh mana sistem yang dibangun sesuai dengan kebutuhan pengguna. Evaluasi dilakukan melalui dua metode, yaitu *Black Box Testing* dan *User Acceptance Testing (UAT)*.

Black Box Testing difokuskan pada pengujian fungsi-fungsi utama sistem dari sisi pengguna, seperti tombol *login*, *daftar*, *navigasi antar halaman*, serta proses pemilihan *jadwal* dan *pendaftaran antrean*. Hasil pengujian menunjukkan bahwa semua fitur berjalan sesuai dengan yang diharapkan, sehingga dapat disimpulkan sistem berfungsi dengan baik dari sisi operasional.

Sedangkan pada pengujian UAT, dilakukan survei kepada 10 pengguna dengan menggunakan kuesioner berbasis skala Likert (skor 1–5). Hasilnya menunjukkan bahwa rata-rata nilai kepuasan pengguna berada pada angka 96%, yang termasuk dalam kategori "Sangat Setuju". Dengan rincian: fitur mudah dipahami (100%), sistem berjalan baik di perangkat (98%), *website* membantu pengguna (96%),

tampilan menarik (96%), dan ketertarikan untuk konsultasi setelah melihat *website* (90%).

Secara keseluruhan, hasil evaluasi membuktikan bahwa sistem reservasi berbasis *website* ini telah diterima dengan baik oleh pengguna dan mampu mendukung proses pelayanan di Klinik Pratama Agha Depok secara efektif.

4. KESIMPULAN

Penelitian ini menghasilkan sebuah sistem reservasi klinik berbasis *website* yang berfungsi sebagai solusi digital untuk mempermudah proses pendaftaran konsultasi di Klinik Pratama Agha Depok. Sistem ini memungkinkan pasien untuk melakukan registrasi, memilih jadwal konsultasi, serta memantau status antrean secara mandiri dan efisien.

Berdasarkan hasil pengujian *Black Box*, seluruh fitur sistem berjalan sesuai dengan fungsionalitas yang dirancang, tanpa ditemukan kesalahan pada interaksi pengguna. Selain itu, pengujian *User Acceptance Testing* (UAT) yang dilakukan terhadap sepuluh responden menunjukkan tingkat kepuasan pengguna yang sangat tinggi, dengan rata-rata nilai sebesar 96%. Hal ini menunjukkan bahwa sistem telah diterima dengan baik oleh pengguna dari sisi kegunaan, kemudahan akses, serta tampilan antarmuka yang *informatif*.

Dengan demikian, dapat disimpulkan bahwa tujuan penelitian telah tercapai, yaitu membangun sistem reservasi yang mampu meningkatkan efisiensi layanan klinik dan memberikan kemudahan bagi pasien dalam melakukan pendaftaran secara *online*.

Sebagai pengembangan ke depan, sistem ini dapat dilengkapi dengan integrasi *database* yang lebih stabil, fitur notifikasi berbasis email atau SMS, serta penyempurnaan UI/UX berdasarkan evaluasi lanjutan. Selain itu, perlu dilakukan pengujian pada skala pengguna yang lebih luas untuk mengetahui performa sistem dalam kondisi trafik tinggi.

DAFTAR PUSTAKA

- [1] P. D. P. Silitonga and D. E. R. Purba, "Implementasi System Development Life Cycle Pada Rancang Bangun Sistem Pendaftaran Pasien Berbasis Web," *Jurnal Sistem Informasi Kaputama (JSIK)*, vol. 5, no. 2, pp. 196–203, Jul. 2021, doi: 10.59697/jsik.v5i2.712.
- [2] D. Triandini and F. Agustina, "Pengaruh Waktu Tunggu Antrian Pasien terhadap Kepuasan Pasien di Klinik Pratama Promedika," *Jurnal Ekonomi, Manajemen, Akuntansi Bisnis Digital, Ekonomi Kreatif, dan Entrepreneur*, vol. 4, no. 2, pp. 389–398, Jun. 2024.
- [3] S. W. Primadhani, Y. Ilyas, and A. I. Atthahirah, "Sistem Pendaftaran Online sebagai suatu Strategi Peningkatan Layanan Rumah Sakit: Literature Review," *Media Publikasi Promosi Kesehatan Indonesia (MPPKI)*, vol. 6, no. 1, pp. 20–26, Jan. 2023, doi: 10.56338/mppki.v6i1.2890.
- [4] I. Bismopotro, F. A. Huda, and A. H. Brata, "Pengembangan Single Page Application Berbasis Reactjs Untuk Usaha Percetakan Online (Studi Kasus: Global Grafika)," *J-PTIHK*, vol. 8, no. 7, Jul 2024, Diakses: 25 Mei 2026. [Daring]. Tersedia pada: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/13985>
- [5] A. Rafi Kusumah and R. Andarsyah, "Generator Structure Library React JS Menggunakan Metode User Centered Design," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 2, pp. 1390–1394, Sep. 2023, doi: 10.36040/jati.v7i2.6860.
- [6] Mgs. M. F. Abdillah, I. L. Sardi, and A. Hadikusuma, "Analisis Performa GetX dan BLoC State Management Library Pada Flutter untuk Perangkat Lunak Berbasis Android," *LOGIC: Jurnal Penelitian Informatika*, vol. 1, no. 1, p. 73, Sep. 2023, doi: 10.25124/logic.v1i1.6479.
- [7] J. M. Rivero, S. Heil, J. Grigera, M. Gaedke, and G. Rossi, "Mock API: An Agile Approach Supporting API-first Web Application Development," *ICWE*, 2013, pp. 7–21. doi: 10.1007/978-3-642-39200-9_4.
- [8] M. Ridwan, I. Fitri, and B. Benrahman, "Rancang Bangun Marketplace Berbasis Website menggunakan Metodologi Systems Development Life Cycle (SDLC) dengan Model Waterfall," *Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 5, no. 2, p. 173, Apr. 2021, doi: 10.35870/jtik.v5i2.209.
- [9] M. Ardiansyah and S. Munir, "Rancang Bangun Web Pelaporan Dakwah menggunakan Metode Unified Process," *Jurnal Informatika Terpadu*, vol. 8, no. 2, pp. 85–92, Sep. 2022, doi: 10.54914/jit.v8i2.463.
- [10] A. A. Surya and I. Haromain, "Rancang Bangun Website Lelang Mobil menggunakan Framework Codeigniter 3 pada PT.ABC," *Jurnal Teknologi Terpadu*, vol. 9, no. 2, pp. 133–142, Dec. 2023, doi: 10.54914/jtt.v9i2.1031.
- [11] Y. Maolana, Asriyanik, and A. Pambudi, "Aplikasi Rekam Medis Imran Medical Center Menggunakan React JS Dengan Metode Prototype," *INFOTECH journal*, vol. 9, no. 2, pp. 626–636, Nov. 2023, doi: 10.31949/infotech.v9i2.7473.
- [12] H. Anggrainy, "Perancangan Sistem Informasi Pelanggan Pasien Pada Rumah Sakit Jiwa (RSJ) Aceh," *Jurnal Indonesia : Manajemen Informatika*

- dan Komunikasi*, vol. 2, no. 1, pp. 41–52, Jun. 2021, doi: 10.35870/jimik.v2i1.49.
- [13] A. A. I. D. Putri, I. W. W. Karsana, and N. M. Diaris, “Analisis Efektivitas Implementasi Sistem Reservasi Online Pasien Rawat Jalan Pada Poli Jantung Berbasis Web Di Rumah Sakit TK II Udayana,” *Bali Health Published Journal*, vol. 4, no. 2, pp. 39–47, Dec. 2022, doi: 10.47859/bhbj.v4i2.261.
- [14] R. Molly and M. Itaar, “Analisis Pemanfaatan Sistem Informasi Manajemen Rumah Sakit (SIMRS) Pada RRSUD DOK II Jayapura,” *Journal of Software Engineering Ampera*, vol. 2, no. 2, pp. 95–101, Jun. 2021, doi: 10.51519/journalsea.v2i2.127.
- [15] I. G. A. Ayu Sekarini, A. A. Sandatya Widhiyanti, and I. G. A. Bagus Santosa, “Aplikasi Informasi Kunjungan Harian Pasien pada Rumah Sakit Swasta Kota Denpasar Menggunakan Framework Laravel,” *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, vol. 22, no. 2, p. 546, Aug. 2023, doi: 10.53513/jis.v22i2.8627.