

Design and Development of Mobile Based General Affairs Service Application for Resource Management and Room Booking using Flutter

Ridho Sami Rabbani^{1*}, Raden Budiraharjo¹

¹ Program Studi Sistem Informasi, Institut Teknologi Nasional, Bandung, Indonesia

*Email Korespondensi: Ridhosamirabbani1209@gmail.com

Naskah Masuk 30 November 2025 | Direvisi 30 Januari 2026 | Diterima 1 Februari 2026

ABSTRAK

Pengelolaan fasilitas operasional atau General Affairs (GA) di PT PLN Indonesia Power UBP Banten 3 Lontar menghadapi kendala efisiensi akibat prosedur permintaan sumber daya dan pemesanan ruangan yang dilakukan secara manual melalui pesan instan, tanpa sistem pencatatan yang terstandarisasi. Kondisi ini menimbulkan permasalahan nyata, yaitu tumpang tindih jadwal pemesanan ruangan (double booking), tidak adanya rekam jejak permintaan yang terstruktur, lambatnya konfirmasi, serta ketidakmampuan pemohon memantau status permintaan secara mandiri. Penelitian ini membangun aplikasi layanan GA berbasis mobile bernama "MasBro" sebagai solusi digitalisasi atas permasalahan tersebut. Aplikasi dikembangkan menggunakan Flutter dan Dart untuk antarmuka multiplatform yang responsif, serta Firebase sebagai backend berbasis cloud. Metodologi pengembangan menggunakan pendekatan Agile iteratif, mencakup tahap analisis kebutuhan, perancangan dengan UML (Use Case Diagram, Activity Diagram, ERD, Component Diagram), implementasi, pengujian, penerapan, dan evaluasi. Pengujian fungsional dilakukan melalui black box testing dengan 24 test case, seluruhnya berstatus pass (100%). User Acceptance Testing (UAT) yang melibatkan pengguna nyata dari tiga peran (Employee, Officer, dan Teknisi) menunjukkan seluruh alur utama aplikasi berjalan tanpa kendala signifikan. Aplikasi MasBro berhasil mendigitalisasi layanan GA, mengeliminasi risiko double booking, mempercepat koordinasi, dan menyediakan sistem dokumentasi permintaan yang terpusat dan transparan.

Kata kunci: Aplikasi Mobile, General Affairs, Digitalisasi, Flutter, Firebase, Agile.

ABSTRACT

The management of operational facilities or General Affairs (GA) at PT PLN Indonesia Power UBP Banten 3 Lontar faces efficiency constraints due to resource request and room booking procedures conducted manually through instant messaging, without a standardized recording system. This results in tangible problems, including overlapping room booking schedules (double booking), the absence of structured service request records, slow confirmation times, and the inability of requesters to independently monitor their request status. This study developed a mobile-based GA service application named "MasBro" as a comprehensive digitalization solution to these problems. The application was developed using Flutter and Dart for a responsive multi-platform interface, and Firebase as a cloud-based backend. The development methodology employs an iterative Agile approach, covering stages of requirements analysis, UML-based design (Use Case Diagram, Activity Diagram, ERD, Component Diagram), implementation, testing, deployment, and evaluation. Functional testing used the black box testing method across 24 test cases, all achieving a pass status (100%). User Acceptance Testing (UAT) involving real users from three roles (Employee, Officer, and Technician) confirmed that all main application workflows ran without significant issues. MasBro successfully digitalizes the GA service process, eliminating double booking risks, accelerating coordination, and providing a centralized and transparent request documentation system.

Keywords: Mobile Application, General Affairs, Digitalization, Flutter, Firebase, Agile.

1. PENDAHULUAN

Perkembangan teknologi informasi telah membawa transformasi signifikan dalam berbagai aspek operasional perusahaan, termasuk dalam meningkatkan efektivitas, efisiensi, dan produktivitas organisasi [1]. Digitalisasi proses bisnis menjadi kebutuhan yang tidak dapat dihindari, terutama dalam menghadapi tuntutan pengelolaan sumber daya yang semakin kompleks. Penerapan sistem informasi manajemen terbukti mampu meningkatkan efisiensi operasional perusahaan dengan mengotomatisasi proses manual yang sebelumnya memakan waktu dan rentan terhadap kesalahan [2].

PT PLN Indonesia Power UBP Banten 3 Lontar sebagai salah satu unit pembangkit listrik yang berperan penting dalam penyediaan energi listrik nasional, menghadapi tantangan dalam pengelolaan fasilitas operasional yang diemban oleh fungsi General Affairs (GA). General Affairs merupakan fungsi internal perusahaan yang bertanggung jawab atas pengelolaan fasilitas dan berbagai layanan pendukung operasional. Sebelum adanya sistem digitalisasi, seluruh proses permintaan layanan GA mulai dari permintaan sumber daya hingga pemesanan ruangan dilakukan sepenuhnya secara manual melalui komunikasi lisan antarkaryawan atau pesan singkat melalui aplikasi seperti WhatsApp, tanpa adanya formulir standar maupun sistem pencatatan yang terstruktur.

Kondisi ini menimbulkan sejumlah permasalahan operasional yang berulang. Tidak adanya sistem sentralisasi kalender menyebabkan sering terjadinya tumpang tindih jadwal pemesanan ruangan yang dikenal dengan istilah double booking. Selain itu, dokumentasi permintaan layanan yang tidak terdokumentasi dengan baik mengakibatkan sulitnya melakukan audit dan pelaporan karena rekam jejak permintaan tidak tersedia. Proses konfirmasi permintaan pun menjadi lambat karena sangat bergantung pada respons pesan pribadi dari petugas GA, sementara para pemohon tidak dapat memantau status permintaan mereka secara mandiri dan real-time.

Upaya digitalisasi layanan GA di berbagai organisasi telah dilakukan sebelumnya, di antaranya melalui pengembangan aplikasi berbasis web. Implementasi aplikasi General Affairs Service berbasis web telah terbukti efektif dalam meningkatkan komunikasi internal dan mempercepat proses permintaan layanan di berbagai organisasi [3]. Namun, pendekatan berbasis web memiliki keterbatasan yang signifikan dalam konteks operasional lapangan, antara lain: ketergantungan pada perangkat dengan browser dan koneksi stabil, keterbatasan aksesibilitas di lingkungan produksi yang bersifat mobile, serta tidak adanya dukungan notifikasi push secara real-time tanpa bergantung pada email atau pengecekan manual. Kenyataan ini mendorong kebutuhan akan solusi berbasis aplikasi mobile yang dapat diakses kapan saja dan di mana saja oleh karyawan, termasuk di area operasional lapangan.

Pendekatan mobile-first menawarkan sejumlah keunggulan dibandingkan sistem berbasis web, yaitu aksesibilitas yang lebih tinggi melalui perangkat genggam, kemampuan notifikasi push yang proaktif, pengalaman antarmuka yang lebih intuitif dan responsif, serta kemampuan integrasi dengan fitur perangkat seperti kamera untuk dokumentasi lapangan [4]. Framework Flutter telah menjadi pilihan populer dalam pengembangan aplikasi mobile karena kemampuannya menghasilkan aplikasi cross-platform (Android dan iOS) dengan performa tinggi dan antarmuka yang responsif hanya dengan satu basis kode [5]. Hal ini menjadikan Flutter lebih efisien dari sisi biaya dan waktu pengembangan dibandingkan membangun aplikasi secara terpisah untuk setiap platform. Kombinasi Flutter dengan Firebase sebagai backend memberikan solusi yang kuat untuk aplikasi yang membutuhkan penyimpanan data real-time dan sistem notifikasi yang andal [6].

Dalam merancang sistem informasi, penggunaan Unified Modeling Language (UML) menjadi penting untuk mendokumentasikan dan memvisualisasikan struktur sistem secara komprehensif. Teknik Entity-Relationship Diagram (ERD) membantu dalam perancangan database yang terstruktur dan efisien [7]. Use Case Diagram berperan penting dalam mendokumentasikan interaksi antara pengguna dan sistem sehingga seluruh kebutuhan fungsional dapat teridentifikasi secara komprehensif sejak tahap analisis [8]. Pemodelan UML memungkinkan pemangku kepentingan untuk memahami alur sistem dan memastikan bahwa semua kebutuhan fungsional terdokumentasi dengan baik sebelum tahap implementasi.

Penelitian ini bertujuan untuk merancang dan membangun aplikasi mobile berbasis Flutter dengan nama "MasBro" sebagai solusi digitalisasi layanan General Affairs di PT PLN Indonesia Power UBP Banten 3 Lontar. Aplikasi ini dirancang secara khusus untuk mengatasi kendala sistem manual yang selama ini berlangsung, dengan menyediakan fitur yang memungkinkan karyawan mengajukan permintaan layanan dan melakukan pemesanan fasilitas secara digital, termonevitas, dan terdokumentasi. Melalui integrasi dengan Firebase, aplikasi ini menyediakan sistem notifikasi real-time yang memastikan setiap permintaan dapat ditangani dengan cepat dan tepat.

Pengembangan aplikasi ini mengikuti metodologi Agile yang iteratif, dimulai dari analisis kebutuhan pengguna, perancangan sistem menggunakan UML, implementasi dengan Flutter dan Firebase, hingga pengujian melalui black box testing dan User Acceptance Testing (UAT). Dengan adanya aplikasi MasBro, diharapkan dapat diperoleh peningkatan nyata dalam efisiensi pengelolaan layanan GA: eliminasi double booking melalui sistem kalender terpusat, percepatan proses konfirmasi melalui notifikasi otomatis, peningkatan transparansi koordinasi melalui status tracking real-time, dan penyediaan sistem dokumentasi permintaan yang terorganisir dan mudah diakses.

2. METODOLOGI

Penelitian ini menggunakan pendekatan yang terstruktur dalam mengembangkan aplikasi MasBro sebagai solusi digitalisasi layanan General Affairs di PT PLN Indonesia Power UBP Banten 3 Lontar. Metodologi yang digunakan meliputi alat dan teknologi yang digunakan dalam pengembangan aplikasi, serta metodologi pengembangan sistem berbasis Agile yang diterapkan secara iteratif. Penerapan teknologi informasi yang tepat menjadi fondasi utama dalam meningkatkan efektivitas dan efisiensi operasional perusahaan [1].

2.1 Alat dan Teknologi

Berbagai alat dan teknologi dipilih untuk mendukung proses analisis, perancangan, pengembangan, dan kolaborasi dalam proyek ini. Pemilihan alat dan teknologi disesuaikan dengan kebutuhan proyek serta standar pengembangan perangkat lunak modern berbasis mobile.

2.1.1 Flutter dan Dart

Aplikasi MasBro dikembangkan menggunakan framework Flutter dengan bahasa pemrograman Dart. Flutter dipilih karena mampu membangun aplikasi multiplatform (Android dan iOS) hanya dengan satu basis kode, sehingga lebih efisien dari segi waktu dan biaya pengembangan [5]. Flutter memiliki dukungan widget yang lengkap dan komunitas yang terus berkembang, sehingga memudahkan proses debugging maupun penambahan fitur. Bahasa Dart mendukung paradigma pemrograman berorientasi objek dengan sintaks yang sederhana dan terstruktur, sehingga cocok digunakan dalam membangun aplikasi skala menengah hingga besar [5].

2.1.2 Firebase

Aplikasi ini menggunakan layanan Firebase sebagai sistem backend karena sifatnya yang berbasis cloud dan mudah diintegrasikan dengan Flutter [9]. Firestore Database digunakan sebagai basis data NoSQL untuk menyimpan data permintaan dan riwayat aktivitas secara real-time [10]. Firebase Authentication dipakai untuk mengelola akun pengguna sesuai peran (User, Officer, Teknisi, dan Admin), Firebase Storage untuk menyimpan file seperti foto kerusakan atau bukti penyelesaian tugas, serta Firebase Functions untuk menangani logika bisnis di sisi server seperti pengiriman notifikasi otomatis. Sistem notifikasi real-time merupakan komponen kritis dalam aplikasi yang membutuhkan komunikasi instan antar pengguna, dan Firebase Cloud Messaging (FCM) telah terbukti menjadi solusi yang efektif untuk mengirimkan pesan notifikasi secara lintas platform tanpa memerlukan infrastruktur server tersendiri [11]. Pemanfaatan Firebase secara menyeluruh ini membuat aplikasi MasBro lebih efisien, skalabel, dan cukup mudah untuk dikelola.

2.1.3 Visual Studio Code

Visual Studio Code (VS Code) digunakan sebagai code editor dalam pengembangan aplikasi. IDE modern seperti VS Code meningkatkan efisiensi pengembangan perangkat lunak melalui fitur-fitur pendukung seperti debugging tools, manajemen task, dan ekosistem ekstensi yang beragam [7]. VS Code dipilih karena

ringan, cepat, dan memiliki banyak ekstensi yang mendukung Flutter dan Dart, termasuk fitur IntelliSense dan integrasi terminal bawaan yang membantu meningkatkan produktivitas pengembangan.

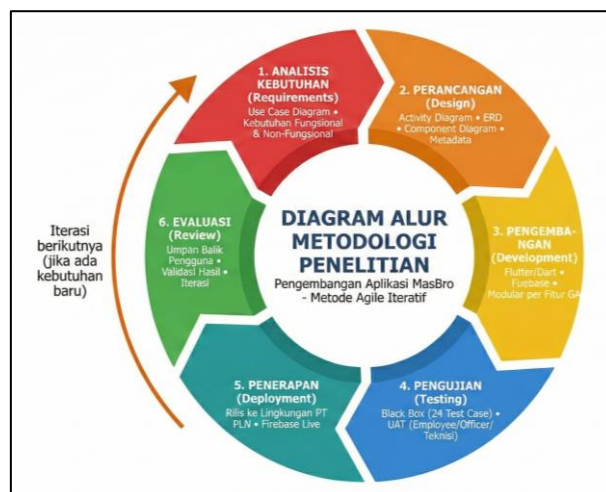
2.1.4 Git dan GitHub

Sistem version control Git dan platform repositori GitHub digunakan untuk mendukung kolaborasi tim dalam pengembangan. Git memungkinkan pencatatan setiap perubahan kode, rollback jika terjadi kesalahan, dan pengelolaan versi aplikasi secara efektif [12]. GitHub digunakan sebagai repositori pusat yang dapat diakses oleh seluruh anggota tim, memungkinkan proses review kode dan integrasi antar modul dilakukan dengan lebih transparan dan terorganisasi [12].

2.2 Metodologi Pengembangan Sistem

Pengembangan aplikasi MasBro menggunakan metodologi Agile, yang dipilih karena sifatnya yang fleksibel terhadap perubahan dan memungkinkan pengembangan dilakukan secara bertahap dan iteratif [4]. Penelitian terdahulu menunjukkan bahwa penerapan metodologi Agile pada pengembangan sistem informasi dapat mengurangi tingkat kegagalan proyek hingga signifikan dibandingkan dengan pendekatan tradisional seperti waterfall, sekaligus meningkatkan kolaborasi antar anggota tim pengembang [13]. Pemilihan pendekatan Agile didasarkan pada kondisi nyata di PT PLN Indonesia Power UBP Banten 3 Lontar, di mana kebutuhan pengguna belum seluruhnya terdokumentasi secara formal sebelum pengembangan dimulai, sehingga diperlukan pendekatan yang fleksibel terhadap perubahan dan memungkinkan validasi bertahap bersama pengguna nyata untuk memastikan solusi yang dikembangkan tetap sesuai dengan kebutuhan operasional yang berkembang.

Gambar 1 menggambarkan alur metodologi penelitian ini, memberikan visualisasi yang jelas tentang langkah-langkah yang diambil dalam pengembangan aplikasi dan bagaimana masing-masing fase berkontribusi terhadap keberhasilan proyek.



Gambar 1. Diagram Alur Metodologi Penelitian

2.2.1 Analisis Kebutuhan

Tahap awal dimulai dengan pengumpulan kebutuhan fungsional dan non-fungsional dari pengguna untuk menentukan fitur-fitur prioritas seperti manajemen pemesanan ruangan dan pelaporan kerusakan fasilitas. Penerapan sistem informasi manajemen yang tepat terbukti dapat meningkatkan efisiensi operasional perusahaan melalui otomatisasi proses yang sebelumnya dilakukan secara manual [2]. Dari proses ini teridentifikasi empat kebutuhan layanan utama yang harus dipenuhi oleh aplikasi, yaitu pengelolaan permintaan sumber daya atau item oleh karyawan yang ditangani oleh officer dan teknisi, pemesanan ruangan rapat yang terpusat dan terhindar dari double booking, dashboard monitoring yang disesuaikan per peran pengguna (Employee, Officer, Teknisi, Admin), dan sistem notifikasi real-time agar setiap perubahan status dapat diketahui segera oleh pihak yang bersangkutan. Kebutuhan-kebutuhan tersebut kemudian

didokumentasikan melalui Use Case Diagram untuk mendefinisikan batasan sistem dan interaksi antar aktor secara grafis [8].

2.2.2 Perancangan

Pada tahap perancangan, kebutuhan yang telah dikumpulkan divisualisasikan secara teknis menggunakan Activity Diagram untuk memetakan alur kerja proses bisnis seperti alur pengajuan permintaan resource dari karyawan, verifikasi officer, hingga penugasan teknisi dan penilaian kinerja, dan Component Diagram untuk menggambarkan arsitektur fisik sistem, yaitu hubungan antara aplikasi Flutter sebagai frontend dengan layanan Firebase (Firestore, Authentication, Storage, Cloud Functions, dan FCM) sebagai backend. Selain itu, perancangan struktur data dilakukan melalui Entity Relationship Diagram (ERD) untuk memetakan hubungan antar entitas seperti Pengguna dan Permintaan [6]. Hasil perancangan kemudian disusun ke dalam Tabel Metadata sebagai kamus data teknis bagi pengembang selama tahap implementasi, sehingga proses implementasi dapat dilakukan secara lebih terstruktur dan konsisten.

2.2.3 Pengembangan

Tahap pengembangan dilaksanakan dengan mengimplementasikan seluruh rancangan ke dalam kode program. Aplikasi MasBro dikembangkan menggunakan Flutter dengan bahasa Dart di sisi frontend, memanfaatkan fitur Hot Reload yang memungkinkan perubahan antarmuka terlihat secara instan tanpa perlu memulai ulang proses kompilasi, hal ini sangat mendukung pendekatan Agile karena mempercepat siklus iterasi pengembangan fitur [5].

Pengembangan dilakukan secara modular, mengikuti empat modul utama yang telah dirancang, yaitu Modul Layanan Resource/Item, yang mengimplementasikan alur lengkap dari pengajuan permintaan oleh Employee, verifikasi dan penugasan oleh Officer, penyelesaian tugas oleh Teknisi, hingga penilaian kinerja, Modul Layanan BookingRoom, yang mengimplementasikan pemesanan ruangan terpusat dengan mekanisme persetujuan Officer untuk mencegah double booking, Modul Dashboard Utama, yang disesuaikan tampilannya per peran mulai dari karyawan melihat menu layanan dan status permintaan, officer melihat pusat kendali permintaan dan tab rating, hingga teknisi melihat daftar penugasan personal, dan Modul Notifikasi, yang diimplementasikan menggunakan Firebase Cloud Messaging (FCM) melalui Cloud Functions sehingga setiap perubahan status permintaan memicu notifikasi push secara otomatis kepada pengguna yang relevan [11].

Integrasi backend dilakukan sepenuhnya menggunakan Firebase, diantaranya Firestore Database untuk penyimpanan data permintaan dan riwayat aktivitas secara real-time [10], Firebase Authentication untuk manajemen akun dan kontrol akses berbasis peran, Firebase Storage untuk penyimpanan file pendukung seperti foto kerusakan atau bukti penyelesaian tugas, serta Firebase Functions untuk menangani logika bisnis sisi server. Selama pengembangan, version control dikelola menggunakan Git dan GitHub untuk memastikan setiap perubahan kode terdokumentasi dan dapat di-rollback jika diperlukan [12].

2.2.4 Pengujian

Pengujian dilakukan dalam dua tahap. Tahap pertama adalah pengujian fungsional menggunakan metode black box testing, di mana setiap fitur divalidasi berdasarkan input dan output yang diharapkan tanpa memperhatikan struktur internal kode [14]. Pengujian fungsional mencakup 24 test case yang disusun berdasarkan alur kerja pada Activity Diagram, mencakup skenario positif dan negatif untuk seluruh empat modul aplikasi (Layanan Resource/Item: 7 test case, Layanan BookingRoom: 9 test case, Dashboard Utama: 4 test case, Notifikasi: 4 test case). Seluruh 24 test case menghasilkan status pass.

Tahap kedua adalah User Acceptance Testing (UAT) yang melibatkan pengguna nyata dari tiga peran: Employee yang bertugas mengajukan permintaan layanan dan pemesanan ruangan, Officer yang bertugas memverifikasi dan mengelola permintaan, serta Teknisi yang menangani penugasan teknis. Setiap pengguna menjalankan skenario penggunaan yang mencakup seluruh fitur utama sesuai perannya di lingkungan testing yang menyerupai kondisi operasional PT PLN Indonesia Power UBP Banten 3 Lontar. Validasi UAT mencakup tiga dimensi, yaitu kelengkapan fitur, kemudahan penggunaan, dan kenyamanan antarmuka [15].

2.2.5 Penerapan

Setelah dinyatakan lolos pengujian, aplikasi MasBro dirilis ke lingkungan terbatas yang melibatkan pengguna representatif dari PT PLN Indonesia Power UBP Banten 3 Lontar. Penerapan memanfaatkan infrastruktur Firebase yang telah dikonfigurasi selama pengembangan, sehingga seluruh layanan backend mulai dari penyimpanan data, autentikasi, hingga pengiriman notifikasi langsung dapat beroperasi secara live tanpa memerlukan konfigurasi server tambahan [9]. Pembaruan fitur selanjutnya dapat didistribusikan dengan cepat karena perubahan pada Cloud Functions dan Firestore langsung terrefleksi pada aplikasi yang berjalan di perangkat pengguna.

2.2.6 Evaluasi

Tahap terakhir dalam siklus Agile ini adalah evaluasi hasil penerapan bersama pemangku kepentingan di PT PLN Indonesia Power. Evaluasi bertujuan untuk memvalidasi apakah aplikasi yang telah di-deploy sudah memenuhi seluruh kebutuhan operasional yang diidentifikasi pada tahap awal. Umpan balik yang diperoleh dari pengguna selama UAT dan tahap penerapan menjadi bahan evaluasi utama. Berdasarkan hasil evaluasi ini, seluruh fitur yang dirancang telah berfungsi sesuai spesifikasi, dan pengguna dari ketiga peran dapat menyelesaikan task utama mereka tanpa kendala signifikan. Apabila di masa mendatang terdapat kebutuhan baru, misalnya penambahan fitur pelaporan atau integrasi dengan sistem internal PLN proses dapat kembali ke tahap analisis kebutuhan untuk memulai siklus pengembangan berikutnya [4]. Pendekatan iteratif ini memastikan bahwa aplikasi terus berkembang sesuai dengan kebutuhan operasional yang dinamis.

3. HASIL DAN PEMBAHASAN

3.1 Analisis Kebutuhan dan Perancangan Sistem

3.1.1 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan melalui diskusi langsung dengan pemangku kepentingan di PT PLN Indonesia Power UBP Banten 3 Lontar untuk mengidentifikasi kendala pada alur kerja manual yang berjalan. Kendala utama yang teridentifikasi meliputi tidak adanya sistem pencatatan permintaan yang terpusat, terjadinya double booking pada pemesanan ruangan, lambatnya proses konfirmasi yang bergantung pada respons pesan pribadi, serta ketidakmampuan pemohon untuk memantau status permintaan secara mandiri.

Berdasarkan analisis tersebut, penulis menetapkan empat layanan utama yang harus disediakan oleh sistem, yaitu Layanan Resource/Item untuk mengelola permintaan sumber daya berupa tenaga kerja atau barang dari karyawan yang ditangani officer dan teknisi, Layanan BookingRoom untuk mengelola pemesanan ruangan rapat secara terpusat dengan mekanisme persetujuan officer, Dashboard Utama sebagai pusat navigasi dan monitoring yang disesuaikan per peran pengguna, dan Fitur Notifikasi untuk memberikan pembaruan status secara real-time kepada seluruh pihak yang terlibat. Untuk menggambarkan alur proses bisnis secara detail, penulis merancang Activity Diagram yang memodelkan urutan aktivitas dan keputusan dalam setiap modul sistem.

Dari analisis ini, penulis mengidentifikasi kebutuhan fungsional dan non-fungsional sistem. Kebutuhan fungsional mencakup seluruh fungsi dan kemampuan yang harus dimiliki sistem untuk memenuhi kebutuhan pengguna, sementara kebutuhan non-fungsional mencakup aspek keandalan, keamanan, kegunaan, dan performa yang memastikan sistem tidak hanya bekerja sesuai fungsi namun juga memenuhi standar kualitas yang diharapkan. Seluruh kebutuhan ini didokumentasikan menggunakan Use Case Diagram dan Activity Diagram yang menggambarkan interaksi dan alur proses sistem aplikasi MasBro.

Tabel 1 dibawah ini merangkum kebutuhan fungsional per modul, sedangkan Tabel 2 merangkum kebutuhan non-fungsional sistem.

Tabel 1. Ringkasan Kebutuhan Fungsional Sistem

Fitur	Deskripsi
Autentikasi Pengguna	Sistem memungkinkan karyawan, officer, teknisi, dan admin melakukan login menggunakan akun valid.

Dashboard Utama	Sistem menampilkan halaman home dashboard berisi status Open dan InProgress, daftar aplikasi tersedia, notifikasi terbaru, dan akses ke layanan GA.
-----------------	---

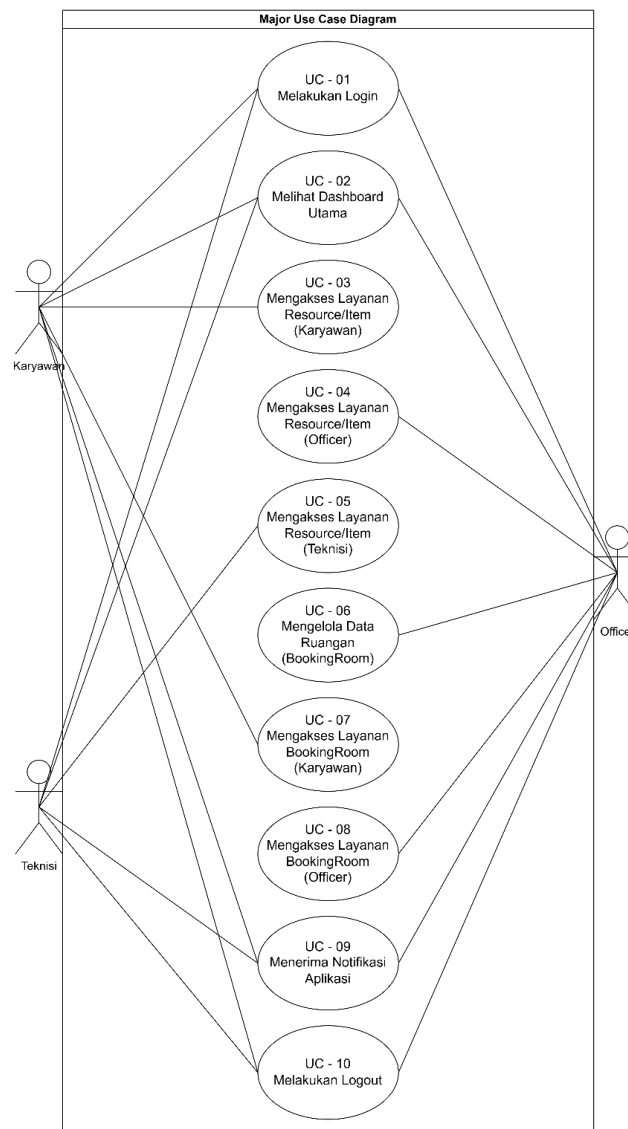
Tabel 1. Ringkasan Kebutuhan Fungsional Sistem (Lanjutan)

Fitur	Deskripsi
Layanan Resource/Item (Karyawan)	Sistem memungkinkan karyawan membuat permintaan baru (resource/item), melihat daftar permintaan, detail permintaan, dan memberikan penilaian kinerja teknisi.
Layanan Resource/Item (Officer)	Sistem memungkinkan officer menerima permintaan, meneruskan ke teknisi (ubah status ke in_progress), atau menyelesaikan langsung (ubah status ke completed), serta melihat data penilaian teknisi.
Layanan Resource/Item (Teknisi)	Sistem memungkinkan teknisi melihat daftar penugasan, menyelesaikan tugas (ubah status ke completed), dan menambahkan catatan penyelesaian.
Layanan BookingRoom (Karyawan)	Sistem memungkinkan karyawan membuat pemesanan ruangan baru, melihat daftar pemesanan, detail pemesanan, dan memberikan penilaian kesiapan ruangan.
Layanan BookingRoom (Officer)	Sistem memungkinkan officer menerima pemesanan, mengelola data ruangan (CRUD), menyetujui (approved) atau menolak (cancelled) pemesanan dengan validasi konflik jadwal, serta melihat data penilaian ruangan.
Notifikasi Real-Time	Sistem mengirimkan notifikasi push kepada karyawan, officer, dan teknisi saat terjadi perubahan status permintaan atau penugasan.

Tabel 2. Kebutuhan Non-Fungsional Sistem

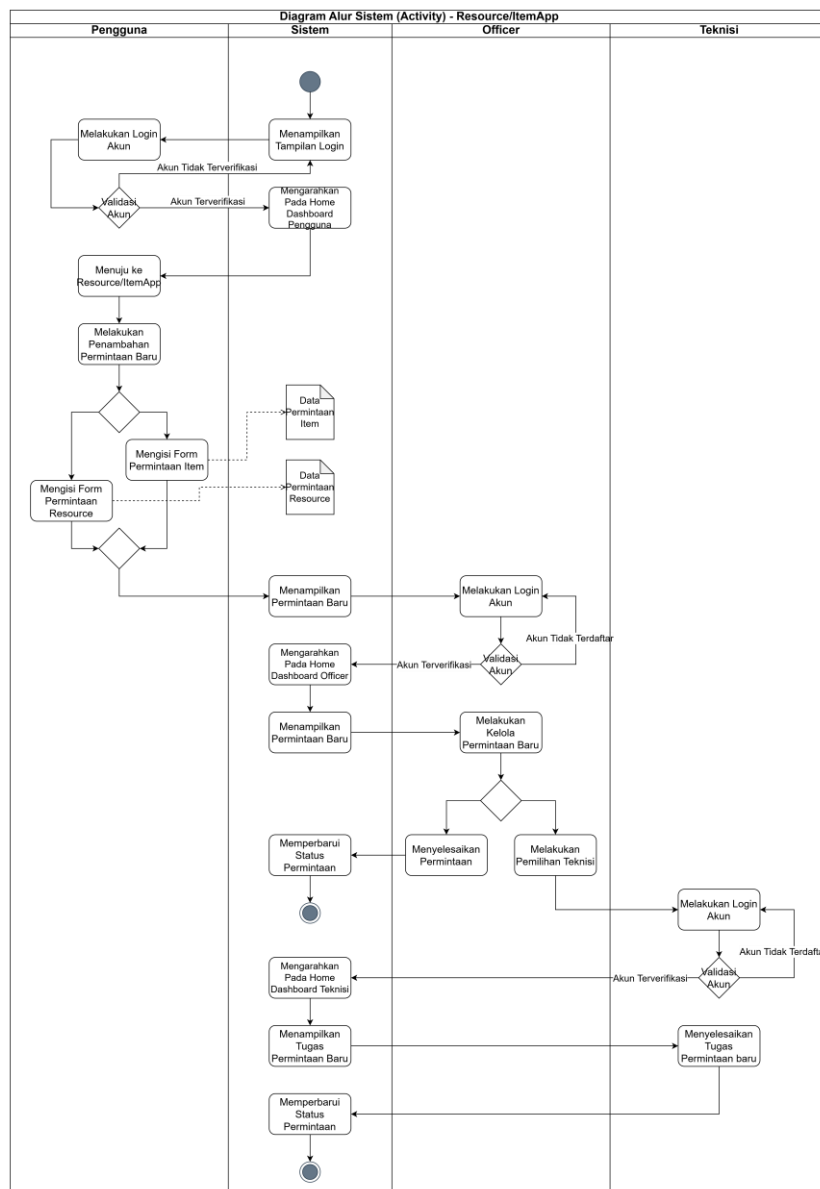
No	Deskripsi
1	Sistem harus memiliki fitur login yang aman dan mengenkripsi data pengguna seperti password dan identitas
2	Antarmuka sistem harus mudah dipahami dan digunakan oleh pengguna tanpa pelatihan teknis khusus
3	Sistem harus dapat diakses melalui berbagai perangkat seperti tablet dan smartphone secara responsif
4	Sistem harus memberikan pesan kesalahan yang jelas dan user-friendly apabila terjadi kesalahan input atau sistem
5	Sistem harus mudah dipelihara dan diperbarui oleh tim pengembang

Gambar 2 menggambarkan Major Use Case Diagram yang memberikan gambaran keseluruhan fungsi sistem secara menyeluruh. Diagram ini mencakup semua proses utama, mulai dari autentikasi pengguna hingga akses ke berbagai layanan General Affairs, sehingga memudahkan pemahaman tentang bagaimana berbagai fungsi berinteraksi satu sama lain dalam aplikasi.



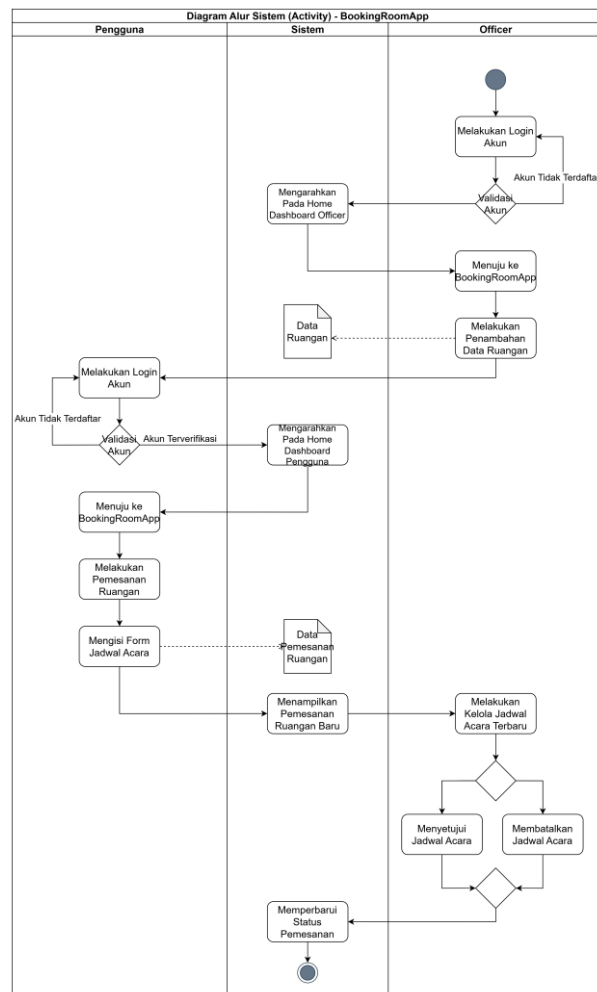
Gambar 2. Major Use Case Diagram

Gambar 2 menampilkan Activity Diagram Layanan Resource/Item yang menggambarkan alur lengkap mulai dari pengguna melakukan login dan validasi akun, menuju ke Resource/Item App untuk melakukan penambahan permintaan baru dengan mengisi form permintaan item atau resource, sistem menampilkan permintaan baru pada dashboard officer, officer melakukan login dan melihat kelola permintaan baru, memperbarui status permintaan dengan menugaskan teknisi yang dipilih, sistem mengirim notifikasi kepada teknisi, teknisi melakukan login dan melihat tugas permintaan baru, menyelesaikan tugas permintaan, sistem memperbarui status permintaan menjadi selesai dan mengirim notifikasi kepada pengguna, hingga pengguna dapat melihat permintaan yang telah selesai dan memberikan penilaian kinerja teknisi.



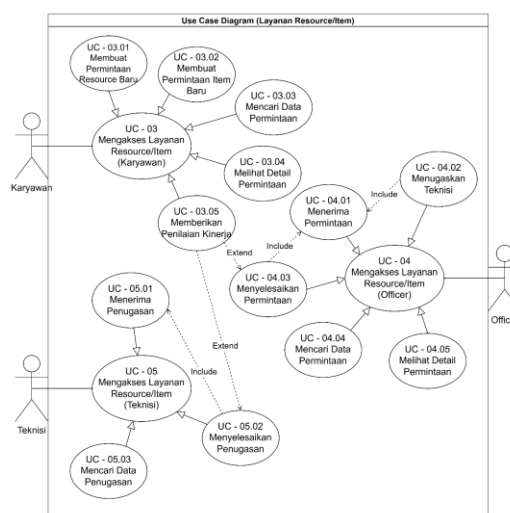
Gambar 2. Activity Diagram Layanan Resource/Item

Gambar 3 menampilkan Activity Diagram Layanan BookingRoom yang menggambarkan alur pemesanan ruangan mulai dari pengguna melakukan login dan validasi akun, menuju ke BookingRoom App untuk melakukan pemesanan ruangan dengan mengisi form jadwal acara, sistem menampilkan pemesanan ruangan baru, officer melakukan login dan menuju ke BookingRoom App untuk melakukan penambahan data ruangan yang tersedia dan melihat kelola jadwal acara terbaru, officer dapat menyetujui jadwal acara jika jadwal tidak bentrok atau membatalkan jadwal acara jika jadwal bentrok dengan acara lain, sistem memperbarui status pemesanan dan mengirimkan notifikasi kepada pengguna.



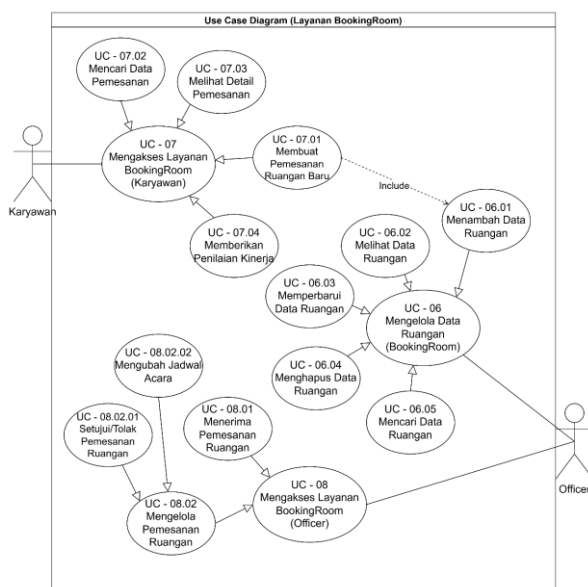
Gambar 3. Activity Diagram Layanan BookingRoom

Gambar 4 menunjukkan Use Case Diagram Layanan Resource/Item yang mendokumentasikan fungsi-fungsi terkait pengelolaan permintaan sumber daya, mencakup seluruh proses mulai dari pengajuan permintaan hingga penugasan kepada teknisi.



Gambar 4. Use Case Diagram Layanan Resource/Item

Gambar 5 menggambarkan Use Case Diagram Layanan BookingRoom yang menunjukkan interaksi antara pengguna dan officer dalam proses pemesanan ruangan, termasuk mekanisme persetujuan dan pengelolaan data ruangan.



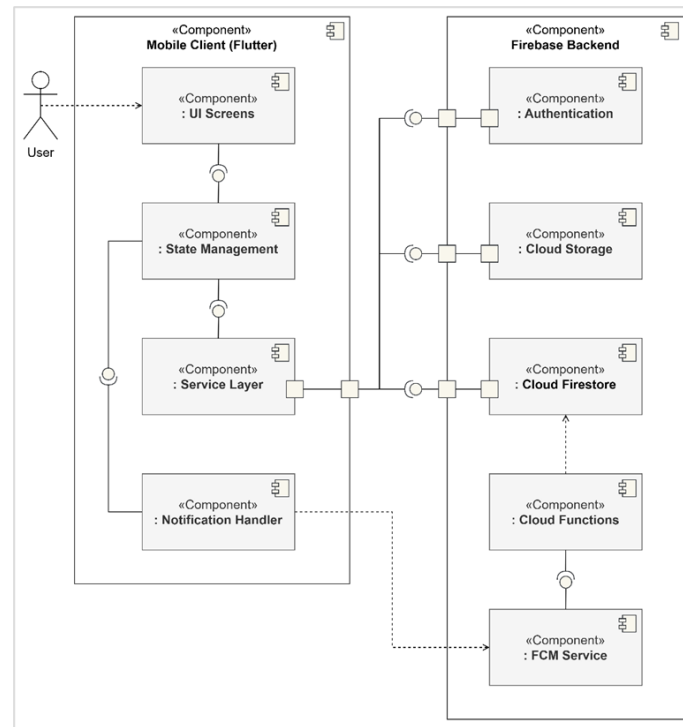
Gambar 5. Use Case Diagram Layanan BookingRoom

3.1.2 Perancangan Arsitektur Sistem

Aplikasi MasBro dirancang menggunakan arsitektur client-server berbasis cloud. Lapisan client adalah aplikasi mobile Flutter yang berjalan di perangkat Android pengguna, sedangkan lapisan server dikelola sepenuhnya oleh layanan Firebase tanpa memerlukan infrastruktur server mandiri. Komunikasi antara client dan server dilakukan melalui Firebase SDK yang menyediakan koneksi langsung dan sinkronisasi data secara real-time dengan Firestore Database melalui protokol WebSocket. Pendekatan ini dipilih karena memungkinkan pembaruan status dan pengiriman notifikasi secara instan tanpa mekanisme polling [9].

Arsitektur sistem terdiri dari tiga lapisan utama. Lapisan pertama adalah Presentation Layer berupa antarmuka Flutter yang merender tampilan berbasis widget dan menangani interaksi pengguna. Lapisan kedua adalah Business Logic Layer yang diimplementasikan sebagian di dalam aplikasi Flutter melalui kelas Service dan sebagian di sisi server melalui Firebase Cloud Functions untuk logika yang membutuhkan eksekusi terpercaya seperti pengiriman notifikasi push otomatis. Lapisan ketiga adalah Data Layer yang sepenuhnya dikelola Firebase, mencakup Firestore sebagai basis data NoSQL berbasis dokumen, Firebase Storage untuk penyimpanan berkas, dan Firebase Authentication untuk manajemen identitas pengguna.

Gambar 6 dibawah ini menampilkan Diagram Komponen yang menggambarkan interaksi antar komponen arsitektur sistem secara lengkap.



Gambar 6. Component Diagram

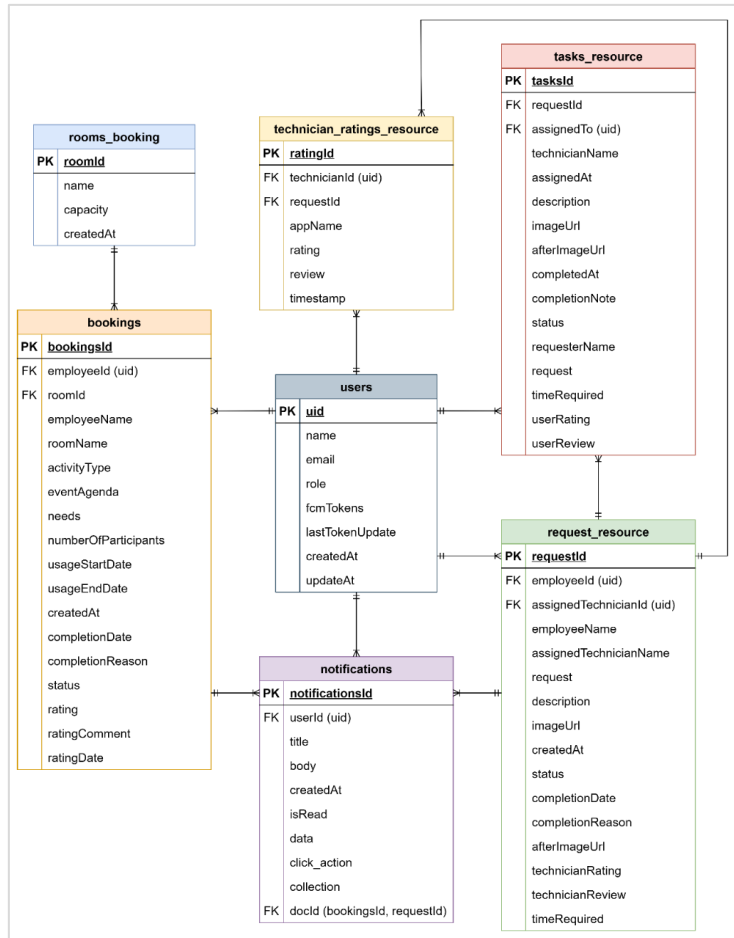
3.1.3 Perancangan Basis Data

Basis data aplikasi MasBro dirancang menggunakan Firebase Cloud Firestore, yaitu basis data NoSQL berbasis dokumen yang menyimpan data dalam bentuk koleksi dan dokumen. Tabel 3 berikut merangkum struktur tujuh koleksi utama yang dirancang untuk mendukung seluruh fungsionalitas sistem.

Tabel 3. Struktur Koleksi Firestore Aplikasi MasBro

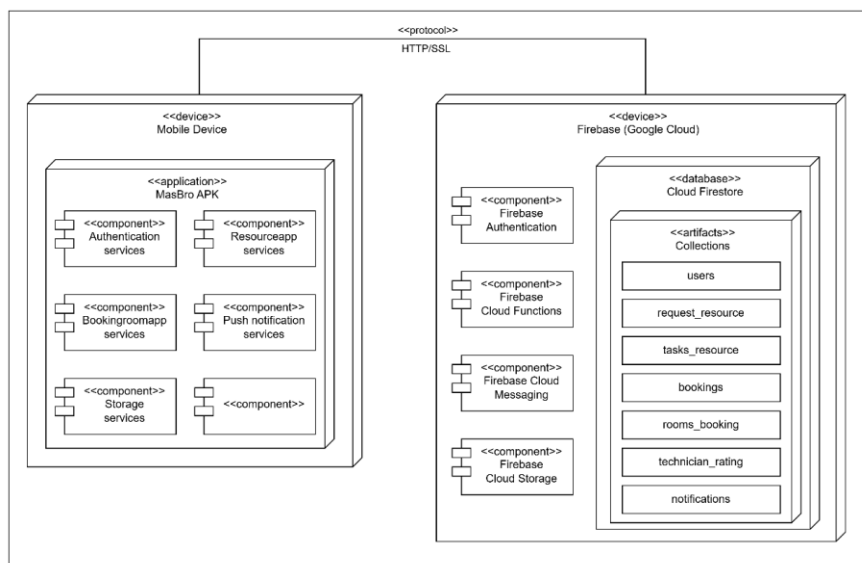
Koleksi	Entitas	Field Kunci dan Keterangan
users	Data akun pengguna	uid (PK), email, name, role, fcmTokens, createdAt
requests_resource	Permintaan sumber daya	requestId (PK), employeeId (FK), request, description, status, assignedTechnicianId (FK), technicianRating
tasks_resource	Penugasan teknisi	taskId (PK), requestId (FK), assignedTo (FK), description, completedAt, completionNote
rooms	Data ruangan	id (PK), name, capacity, createdAt
bookings	Pemesanan ruangan	bookingId (PK), employeeId (FK), roomId (FK), eventAgenda, usageStartDate, usageEndDate, status, rating
notifications	Riwayat notifikasi	notifId (PK), userId (FK), title, body, collection, docId, createdAt
technician_ratings_resource	Penilaian teknisi	ratingId (PK), requestId (FK), technicianId (FK), rating, review

Gambar 7 ini adalah rancangan visualisasi dalam Entity Relationship Diagram yang menampilkan hubungan antar entitas beserta kardinalitasnya.



Gambar 7. Entity Relationship Diagram

Gambar 8 adalah Deployment Diagram yang menggambarkan konfigurasi fisik dan deployment sistem aplikasi MasBro. Diagram ini menunjukkan dua jenis perangkat yaitu perangkat mobile pengguna yang menjalankan aplikasi MasBro APK dan komponen layanan Firebase yang berfungsi sebagai backend.



Gambar 8. Deployment Diagram

3.1.4 Mekanisme Autentikasi dan Kontrol Akses

Sistem autentikasi menggunakan Firebase Authentication dengan metode email dan password. Ketika pengguna memasukkan kredensial pada halaman login, Firebase Authentication memvalidasi kredensial dan mengembalikan token sesi. Sistem kemudian membaca field role pada koleksi users di Firestore untuk menentukan peran pengguna, lalu mengarahkan ke Dashboard yang sesuai.

Kontrol akses berbasis peran (Role-Based Access Control atau RBAC) diimplementasikan secara berlapis untuk memastikan keamanan data [16]. Di sisi client, tampilan dan menu dibatasi berdasarkan nilai role yang dibaca dari Firestore setelah login. Misalnya, tombol "Tugaskan Teknisi" hanya muncul pada antarmuka Officer. Di sisi server, Firebase Security Rules pada Firestore mengatur hak baca dan tulis setiap koleksi berdasarkan uid dan role pengguna yang terautentikasi. Dengan demikian, data tidak dapat diakses atau dimanipulasi oleh peran yang tidak berwenang meskipun pengguna mencoba mengaksesnya secara langsung. Terdapat empat peran dalam sistem yaitu Employee, Officer, Teknisi, dan Admin dengan wewenang masing-masing.

Implementasi keamanan juga mencakup pengelolaan token FCM. Saat pengguna logout, token perangkat dihapus dari array fcmTokens di Firestore agar notifikasi tidak terkirim ke perangkat yang sesinya telah berakhir.

3.1.5 State Management dan Arsitektur Kode

Pengelolaan state pada aplikasi MasBro menggunakan pendekatan setState bawaan Flutter yang dikombinasikan dengan pola Service Layer. Setiap modul memiliki kelas Service tersendiri untuk menangani komunikasi dengan Firestore, seperti FirestoreServiceResource untuk operasi CRUD pada koleksi requests_resource dan FirestoreServiceBooking untuk koleksi bookings. Pemisahan ini menjaga agar logika bisnis tidak tercampur dengan logika tampilan sehingga kode lebih mudah dipelihara.

Struktur folder aplikasi diorganisasi secara modular di bawah direktori lib/ dengan subdirektori screens/ untuk file antarmuka per modul, services/ untuk kelas pengelola komunikasi Firestore, models/ untuk representasi data entitas (RequestModel, BookingModel, RoomModel), dan widgets/ untuk komponen antarmuka yang dapat digunakan ulang. Package Flutter utama yang digunakan mencakup firebase_core dan firebase_auth untuk integrasi Firebase dan autentikasi, cloud_firestore untuk akses basis data, firebase_storage untuk penyimpanan berkas, firebase_messaging untuk push notification, serta intl untuk pemformatan tanggal dan waktu.

3.1.6 Sistem Notifikasi Real-Time

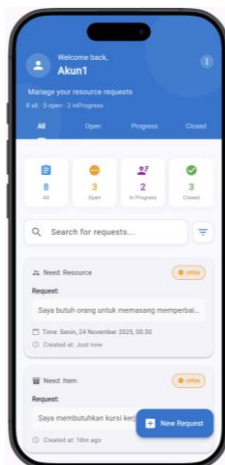
Sistem notifikasi real-time diimplementasikan menggunakan Firebase Cloud Messaging (FCM) yang diintegrasikan dengan Firebase Cloud Functions di sisi server [11]. Ketika terjadi perubahan status pada dokumen Firestore, misalnya status permintaan berubah dari open menjadi in_progress, Cloud Function yang ter-trigger secara otomatis akan membaca token FCM dari array fcmTokens milik pengguna yang bersangkutan, kemudian mengirimkan push notification ke perangkat mobile pengguna tersebut.

Di sisi client Flutter, kelas NotificationService menangani inisialisasi FCM dengan meminta izin notifikasi dari pengguna, menghasilkan token unik per perangkat, dan menyimpan token ke dalam array fcmTokens pada dokumen users di Firestore menggunakan FieldValue.arrayUnion. Pendekatan ini memungkinkan satu akun menerima notifikasi di beberapa perangkat sekaligus. Setiap notifikasi menyertakan payload data berisi collection dan docId untuk mendukung navigasi deep-link langsung ke dokumen terkait saat notifikasi diklik.

3.2 Implementasi Aplikasi

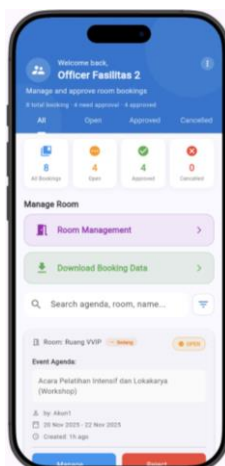
Implementasi aplikasi MasBro dilakukan secara modular mengikuti empat modul utama yang telah dirancang. Setiap modul dikembangkan menggunakan Flutter dan Dart di sisi frontend dengan integrasi Firebase sebagai backend. Implementasi dimulai dengan pembuatan model data untuk merepresentasikan struktur data yang akan disimpan di Firestore, seperti RequestModel, BookingModel, dan RoomModel. Selanjutnya dibuat service class untuk menangani operasi CRUD pada masing-masing koleksi Firestore. Berikut hasil implementasi yang direpresentasikan melalui tampilan antarmuka kunci dari setiap modul.

Gambar 9 menampilkan Dashboard Karyawan pada Layanan Resource/Item yang berisi daftar permintaan yang telah diajukan dilengkapi fitur pencarian dan tombol untuk menambah permintaan baru. Modul ini mengelola seluruh proses permintaan sumber daya, mencakup alur dari pengajuan permintaan oleh Employee, verifikasi dan penugasan oleh Officer, penyelesaian tugas oleh Teknisi, hingga pemberian penilaian kinerja.



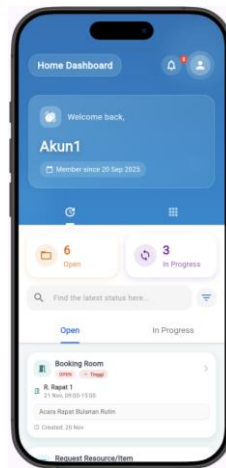
Gambar 9. Tampilan Dashboard Karyawan Aplikasi Resource/Item

Gambar 10 menampilkan Dashboard Officer pada Layanan BookingRoom yang berisi daftar pemesanan ruangan yang telah diajukan. Modul ini mengelola proses pemesanan ruangan rapat dan dirancang untuk mengatasi permasalahan double booking yang sebelumnya sering terjadi melalui mekanisme validasi konflik jadwal otomatis dan persetujuan Officer.



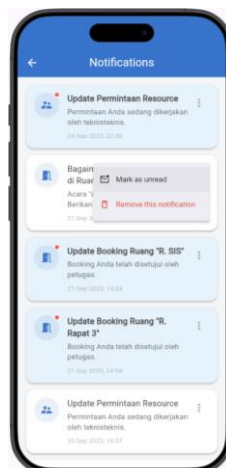
Gambar 10. Tampilan Dashboard Officer Aplikasi Bookingroom

Gambar 11 menampilkan Dashboard Utama untuk Employee yang merupakan halaman pertama setelah login dan berfungsi sebagai pusat navigasi untuk mengakses seluruh layanan GA. Tampilan menampilkan grid menu layanan serta status permintaan yang sedang dalam proses dengan status Open dan InProgress. Dashboard Utama dirancang khusus sesuai peran masing-masing pengguna untuk memudahkan akses dan monitoring.



Gambar 11. Tampilan Dashboard Utama untuk Karyawan

Gambar 12 menampilkan daftar notifikasi terbaru untuk Employee yang berisi notifikasi status permintaan dari seluruh modul layanan di aplikasi. Sistem notifikasi diimplementasikan menggunakan Firebase Cloud Messaging untuk memastikan setiap pengguna mendapatkan pembaruan status secara real-time melalui push notification.



Gambar 12. Tampilan Daftar Notifikasi untuk Karyawan

3.3 Pengujian Sistem

Pengujian sistem dilaksanakan dalam dua tahap untuk memastikan seluruh fitur berfungsi sesuai spesifikasi dan dapat diterima oleh pengguna nyata di lingkungan operasional PT PLN Indonesia Power UBP Banten 3 Lontar.

3.3.1 Pengujian Fungsional (Black Box Testing)

Pengujian fungsional dilakukan menggunakan metode black box testing berbasis manual test case. Setiap test case divalidasi berdasarkan kesesuaian antara input yang diberikan dengan output yang dihasilkan sistem tanpa memperhatikan struktur internal kode [14]. Test case disusun berdasarkan alur kerja pada Activity Diagram, mencakup skenario positif (input valid) dan skenario negatif (input tidak valid, kondisi konflik) [15]. Total 24 test case dijalankan mencakup empat modul aplikasi.

Tabel 4 merangkum hasil pengujian fungsional per modul, sedangkan Tabel 5 menyajikan contoh detail beberapa test case representatif.

Tabel 4. Ringkasan Hasil Pengujian Fungsional per Modul

No	Modul yang Diuji	Jumlah Test Case	Skor
1	Layanan Resource/Item	7	Pass
2	Layanan BookingRoom	9	Pass
3	Fitur Dashboard Utama	3	Pass
4	Fitur Notifikasi Aplikasi	5	Pass
Total		24	Pass (100%)

Tabel 5. Contoh Test Case Representatif per Modul

ID	Test Case	Skenario yang Diuji	Hasil
TC.1	Validasi form kosong (Resource)	Employee submit form tanpa mengisi kolom. Sistem menampilkan pesan validasi dan menolak penyimpanan. (Skenario negatif)	Pass
TC.2	Penugasan teknisi (Resource)	Officer memilih teknisi dan submit. Sistem ubah status ke in_progress dan kirim notifikasi ke Teknisi.	Pass
TC.4	Membuat pemesanan ruangan (Booking)	Employee mengisi form pemesanan dan submit. Sistem menyimpan dengan status open.	Pass
TC.8	Popup peringatan rating (Booking)	Jika pemesanan sebelumnya belum dinilai, sistem tampilkan popup peringatan sebelum boleh membuat pemesanan baru.	Pass
TC.10	Deteksi konflik jadwal (Booking)	Officer coba approve booking pada ruangan dan jadwal yang sudah terisi. Sistem tampilkan pesan error dan tolak persetujuan. (Skenario negatif)	Pass
TC.13	Notifikasi ke Officer (Notifikasi)	Employee submit permintaan baru. Sistem kirim notifikasi push ke perangkat Officer.	Pass
TC.21	Validasi form kosong (Resource)	Employee submit form tanpa mengisi kolom. Sistem menampilkan pesan validasi dan menolak penyimpanan. (Skenario negatif)	Pass

Seluruh 24 test case menghasilkan status pass (100%), membuktikan bahwa aplikasi memenuhi seluruh kebutuhan fungsional yang ditetapkan. Skenario negatif seperti TC.2, TC.10, dan TC.13 juga berhasil divalidasi, menunjukkan sistem mampu menangani kondisi error dengan benar.

3.3.2 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) dilaksanakan untuk memvalidasi bahwa aplikasi dapat digunakan dengan baik oleh pengguna nyata dalam lingkungan operasional PT PLN Indonesia Power UBP Banten 3 Lontar [15]. UAT melibatkan 8 responden yang merepresentasikan tiga peran utama, yaitu 5 responden sebagai Karyawan, 1 responden sebagai Officer, dan 2 responden sebagai Teknisi. Delapan responden merupakan personel yang terlibat langsung dalam proses layanan GA di unit tersebut.

Setiap responden diberikan akses ke aplikasi dalam lingkungan testing dan diminta menjalankan skenario penggunaan yang mencakup seluruh alur utama sesuai perannya. Evaluasi UAT mencakup tiga dimensi yaitu Kelengkapan Fitur (validasi apakah seluruh fungsi yang dibutuhkan tersedia dan dapat dijalankan), Kemudahan Penggunaan (penilaian apakah antarmuka intuitif sesuai), dan Kenyamanan Antarmuka (evaluasi tampilan visual dan responsivitas). Tabel 6 merangkum hasil UAT per responden beserta evaluasi pada ketiga dimensi tersebut.

Tabel 6. Hasil User Acceptance Testing (UAT)

Peran (Responden)	Kelengkapan Fitur	Kemudahan Penggunaan	Kenyamanan Antarmuka	Kesimpulan
Employee (5 responden)	Seluruh alur pengajuan, pemesanan, monitoring status, dan notifikasi berhasil dijalankan	Antarmuka dan form dinilai mudah dipahami tanpa pelatihan teknis	Dashboard intuitif, notifikasi diterima tepat waktu	Memenuhi ekspektasi
Officer (1 responden)	Seluruh fungsi verifikasi, approve/reject, penugasan, pengelolaan ruangan berhasil	Mekanisme approve/reject dan penugasan dinilai jelas dan efisien	Dashboard dengan tab rating informatif untuk monitoring kualitas	Memenuhi ekspektasi
Teknisi (2 responden)	Daftar penugasan tampil benar, pembaruan status dan catatan penyelesaian berhasil	Alur penyelesaian tugas dinilai sederhana	Tampilan daftar penugasan personal ringkas dan mudah dipantau	Memenuhi ekspektasi

Berdasarkan hasil UAT, aplikasi MasBro memenuhi ekspektasi seluruh responden pada ketiga dimensi yang dievaluasi. Seluruh responden berhasil menyelesaikan skenario penggunaan utama tanpa kendala signifikan dan tidak ditemukan bug fungsional selama sesi UAT.

3.4 Evaluasi Efektivitas Sistem

Evaluasi efektivitas dilakukan dengan membandingkan alur kerja layanan GA sebelum dan sesudah implementasi aplikasi MasBro. Perbandingan ini didasarkan pada kondisi manual yang teridentifikasi selama analisis kebutuhan serta kapabilitas sistem yang telah diverifikasi melalui pengujian fungsional dan UAT. Tabel 7 menyajikan perbandingan enam aspek operasional utama yang mengalami perbaikan setelah implementasi sistem.

Tabel 7. Perbandingan Alur Kerja Sebelum dan Sesudah Implementasi

Aspek Operasional	Sebelum	Sesudah
Pengajuan Permintaan	Dilakukan melalui pesan WhatsApp atau lisan tanpa formulir standar dan tidak terdokumentasi	Employee mengisi form digital terstruktur, data langsung tercatat di Firestore dengan status open dan notifikasi otomatis terkirim ke Officer (terverifikasi TC.1)
Pencatatan & Dokumentasi	Tidak ada rekam jejak terdokumentasi, riwayat hanya tersimpan di chat pribadi yang tidak dapat diaudit	Setiap permintaan tersimpan permanen di Firestore dengan riwayat status lengkap dan dapat diunduh sebagai CSV oleh Officer (terverifikasi TC.7)
Pemesanan Ruangan (Double Booking)	Dilakukan secara lisan tanpa sistem sentralisasi, tidak ada validasi konflik jadwal sehingga double booking rutin terjadi	Pemesanan terpusat dalam satu sistem, konflik jadwal dideteksi otomatis dan persetujuan Officer diperlukan sebelum booking valid (terverifikasi TC.13)
Proses Konfirmasi	Bergantung pada respons pesan pribadi petugas GA dengan waktu konfirmasi tidak terprediksi	Officer menerima notifikasi push instan saat permintaan baru masuk dan dapat melakukan konfirmasi kapan saja dari aplikasi (terverifikasi TC.21)
Status Tracking	Pemohon tidak dapat memantau perkembangan permintaan secara mandiri dan harus menghubungi petugas kembali	Pemohon memantau status secara real-time di dashboard (Open, InProgress, Completed/Approved) dan menerima notifikasi setiap perubahan status
Evaluasi Kualitas Layanan	Tidak ada mekanisme formal untuk mengumpulkan umpan balik atas kualitas layanan teknisi maupun fasilitas ruangan	Sistem menyediakan fitur rating dan ulasan oleh Employee setelah layanan selesai, Officer memantau rekapitulasi penilaian melalui tab Rating

Berdasarkan perbandingan pada Tabel 7, implementasi aplikasi MasBro menghasilkan perbaikan pada seluruh aspek operasional layanan GA. Digitalisasi proses permintaan mengeliminasi ketergantungan pada komunikasi tidak terstruktur dan menyediakan audit trail yang komprehensif. Mekanisme deteksi konflik jadwal otomatis yang terverifikasi melalui TC.13 secara langsung mengatasi permasalahan double booking. Sistem notifikasi real-time berbasis FCM memastikan setiap pihak mendapatkan informasi status yang relevan secara proaktif. Fitur penilaian layanan memberikan mekanisme umpan balik yang sebelumnya tidak tersedia sebagai dasar evaluasi kinerja berkelanjutan oleh Officer.

4. KESIMPULAN

Penelitian ini telah berhasil mengembangkan aplikasi MasBro sebagai solusi digitalisasi layanan General Affairs di PT PLN Indonesia Power UBP Banten 3 Lontar menggunakan metodologi Agile dengan teknologi Flutter dan Firebase. Aplikasi ini mengintegrasikan empat modul utama dalam satu platform terpusat, yaitu Layanan Resource/Item untuk pengelolaan permintaan sumber daya, Layanan BookingRoom untuk pemesanan ruangan dengan validasi konflik jadwal otomatis, Dashboard Utama sebagai pusat navigasi per peran, serta sistem notifikasi real-time berbasis Firebase Cloud Messaging.

Dari sisi pencapaian teknis, penelitian ini berhasil mengimplementasikan arsitektur client-server berbasis Firebase SDK yang memanfaatkan komunikasi real-time melalui WebSocket tanpa memerlukan server mandiri. Sistem autentikasi dan kontrol akses berbasis peran (RBAC) diimplementasikan secara berlapis baik di sisi client maupun server melalui Firebase Security Rules, memastikan keamanan data sesuai wewenang masing-masing peran. Integrasi Firebase Cloud Functions dengan FCM menghasilkan mekanisme notifikasi push otomatis yang ter-trigger setiap terjadi perubahan status dokumen di Firestore. Penggunaan pola Service Layer dalam pengelolaan state Flutter menghasilkan kode yang modular dan mudah dipelihara.

Implementasi sistem menghasilkan peningkatan efisiensi operasional yang signifikan pada layanan GA. Digitalisasi proses permintaan mengeliminasi ketergantungan pada komunikasi manual tidak terstruktur dan menyediakan audit trail yang komprehensif. Mekanisme deteksi konflik jadwal otomatis secara langsung mengatasi permasalahan double booking yang sebelumnya rutin terjadi. Sistem notifikasi real-time memastikan setiap pihak mendapatkan informasi status secara proaktif tanpa perlu pengecekan manual berulang. Fitur penilaian layanan memberikan mekanisme umpan balik yang sebelumnya tidak tersedia untuk evaluasi kinerja berkelanjutan.

Kontribusi desain sistem terletak pada arsitektur serverless berbasis Firebase yang menawarkan skalabilitas dan efisiensi biaya operasional, serta pola Service Layer yang memisahkan logika bisnis dari logika tampilan untuk meningkatkan maintainability kode. Hasil pengujian menunjukkan seluruh 24 test case berhasil dengan status pass 100%, membuktikan aplikasi memenuhi kebutuhan fungsional yang ditetapkan. User Acceptance Testing dengan 8 responden dari tiga peran menunjukkan aplikasi memenuhi ekspektasi pengguna pada aspek kelengkapan fitur, kemudahan penggunaan, dan kenyamanan antarmuka.

Keterbatasan sistem saat ini meliputi ketergantungan penuh pada koneksi internet untuk mengakses seluruh fungsi aplikasi, belum tersedianya mode offline untuk fitur dasar seperti melihat riwayat permintaan, serta belum adanya integrasi dengan sistem kalender eksternal yang dapat menyinkronkan jadwal pemesanan ruangan dengan kalender pribadi pengguna. Untuk pengembangan selanjutnya, disarankan untuk mengimplementasikan integrasi dengan Google Calendar API agar jadwal pemesanan ruangan dapat tersinkronisasi otomatis dengan kalender pengguna, melakukan optimasi mekanisme pengiriman notifikasi push untuk mengurangi konsumsi baterai dan bandwidth, serta mengeksplorasi penerapan AI-based scheduling untuk memberikan rekomendasi waktu dan ruangan optimal berdasarkan pola penggunaan historis.

DAFTAR PUSTAKA

- [1] H. Indrayani, “Penerapan Teknologi Informasi Dalam Peningkatan Efektivitas, Efisiensi Dan Produktivitas Perusahaan,” *Jurnal EL-RIYASAH*, vol. 3, no. 1, pp. 48–56, Dec. 2012, doi: 10.24014/JEL.V3I1.664.
- [2] J. F. Ramadhan, M. N. Arifin, W. R. Safei, and Z. P. A. Setiawan, “Penerapan Sistem Informasi Manajemen dalam Meningkatkan Efisiensi Operasional Perusahaan,” *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis*, pp. 660–663, Jul. 2024, Accessed: Sep. 17, 2025. [Online]. Available: <https://ojs.udb.ac.id/Senatib/article/view/4669>
- [3] Z. A. Ramadhani, C. Hendriyani, T. S. Rochmah, A. Sekretari, and M. Taruna Bakti, “Implementasi Aplikasi General Affair Service dalam Meningkatkan Komunikasi Internal,” *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 7, no. 3, 2023, doi: 10.33395/remik.v7i3.12685.
- [4] F. Leonardy, N. Indah Lestari, dan Zafira Ilmi Najwati, P. Studi Bisnis Digital, J. Matematika dan Teknologi Informasi, and I. Teknologi Kalimantan, “Perancangan Sistem Informasi Pendaftaran Pasien Berbasis Web,” *EQUIVA Journal of Mathematics & Information Technology*, vol. 2, no. 2, 2024.
- [5] A. Sopandi, A. R. Hannan, and H. Khotimah, “Perancangan Aplikasi Mobile Menggunakan Framework Flutter Pada Sistem Informasi Akademik,” *JIKA (Jurnal Informatika)*, vol. 8, no. 3, p. 304, Jul. 2024, doi: 10.31000/jika.v8i3.11402.
- [6] K. 'Afiifah, Z. F. Azzahra, and A. D. Anggoro, “Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review,” *INTECH*, vol. 3, no. 2, pp. 70–74, Nov. 2022, doi: 10.54895/INTECH.V3I2.1682.
- [7] N. Wilyanto, J. Firnando, B. Franko, S. P. Tanzil, H. C. Tan, and E. Hartati, “Pembuatan Website Menggunakan Visual Studio Code di SMA Xaverius 3 Palembang,” *FORDICATE*, vol. 3, no. 1, pp. 1–8, Nov. 2023, doi: 10.35957/FORDICATE.V3I1.5057.
- [8] Siska Narulita, Ahmad Nugroho, and M. Zakki Abdillah, “Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS),” *Bridge : Jurnal publikasi Sistem Informasi dan Telekomunikasi*, vol. 2, no. 3, pp. 244–256, Aug. 2024, doi: 10.62951/bridge.v2i3.174.
- [9] H. Nursobah, S. Kom, M. Kom, M. I. Sa'ad, J. Andrew, and J. Kansil, “Implementasi Framework Flutter Untuk Membangun Aplikasi E-commerce,” *TEPIAN*, Dec. 2024, doi: <https://doi.org/10.51967/tepi.v5i4.3110>.
- [10] R. Kesavan, D. Gay, D. Thevessen, J. Shah, and C. Mohan, “Firestore: The NoSQL Serverless Database for the Application Developer,” *Proc. Int. Conf. Data Eng.*, vol. 2023-April, pp. 3376–3388, 2023, doi: 10.1109/ICDE55515.2023.00259.
- [11] A. Faisol and F. Rahmadianto, “Realtime Notification Pada Aplikasi Berbasis Web Menggunakan Firebase Cloud Messaging (FCM),” 2018.
- [12] P. Keshavrao Mudholkar, M. Mudholkar, and P. Mudholkar, “A Beginner’s Guide to Git and GitHub,” *IJSRD-International Journal for Scientific Research & Development*, vol. 5, pp. 2321–0613, 2017, doi: 10.13140/RG.2.2.20126.89927.
- [13] L. Khoza and C. Marnewick, “Waterfall and agile information system project success rates-a South African perspective,” *South African Computer Journal*, vol. 32, no. 1, pp. 43–73, Jul. 2020, doi: 10.18489/sacj.v32i1.683.
- [14] P. Ayuningtyas, P. K. Ayuningtyas, D. A. WP, and P. Rachmadi, “Performance And Functional Testing With The Black Box Testing Method,” *International Journal of Progressive Sciences and Technologies*, vol. 39, no. 2, pp. 212–218, Jul. 2023, doi: 10.52155/ijpsat.v39.2.5471.
- [15] N. S. R. Pillai and R. R. Hemamalini, “Hybrid User Acceptance Test Procedure to Improve the Software Quality,” *International Arab Journal of Information Technology*, vol. 19, no. 6, pp. 956–964, Nov. 2022, doi: 10.34028/iajit/19/6/14.
- [16] M. Sahyudi and E. R. Susanto, “Analisis Implementasi Sistem Keamanan Basis Data Berbasis Role-Based Access Control (RBAC) pada Aplikasi Enterprise Resource Planning,” *SATESI: Jurnal Sains Teknologi dan Sistem Informasi*, vol. 5, no. 1, pp. 105–116, Apr. 2025, doi: 10.54259/satesi.v5i1.3997.
- [17] Y. F. Putri, A. B. P. Irianto, and S. Sharma, “Comparison of Automatic and Manual Regression Testing on Mobile Application Health Technology with Black Box Testing Method,” 2025.