

Design and Development of a Flutter Mobile Application for General Affairs Maintenance and Operational Services

Ilyas Kalamullah^{1*}, Raden Budiraharjo¹

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

*Email Korespondensi: ilyas.kalamullah@mhs.itenas.ac.id

Naskah Masuk 30 November 2025 | Direvisi 30 Januari 2026 | Diterima 15 Februari 2026

ABSTRAK

Meningkatnya kompleksitas layanan maintenance dan operasional pada unit General Affairs menuntut adanya sistem informasi yang mampu mendukung pengelolaan layanan secara terintegrasi dan efisien, karena proses konvensional yang masih mengandalkan pelaporan manual dan komunikasi terpisah sering menyebabkan keterlambatan penanganan, inkonsistensi data, serta keterbatasan dalam pemantauan status layanan. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi layanan General Affairs berbasis mobile yang dapat memfasilitasi pengelolaan permintaan maintenance dan operasional menggunakan framework Flutter dengan PT PLN Indonesia Power UBP Banten 3 Lontar sebagai objek penelitiannya. Metode penelitian yang digunakan adalah pendekatan Agile dalam pengembangan perangkat lunak yang meliputi tahapan perencanaan kebutuhan, perancangan sistem, pengembangan iteratif, serta pengujian fungsional menggunakan metode black-box testing untuk memastikan kesesuaian sistem dengan kebutuhan pengguna. Aplikasi yang dikembangkan memiliki fitur utama berupa pengajuan permintaan layanan, pelaporan aktivitas operasional, pemantauan tugas, serta pelacakan status layanan secara real-time. Sebagai indikator keberhasilan sistem, hasil pengujian fungsional dengan metode black-box testing terhadap 13 test case menunjukkan persentase tingkat keberhasilan mencapai 100% (berstatus pass), yang membuktikan bahwa aplikasi telah memenuhi seluruh kebutuhan penggunaannya secara akurat. Hasil penelitian menunjukkan bahwa aplikasi mampu meningkatkan efektivitas pengelolaan layanan, mempercepat akses informasi, serta mendukung alur kerja operasional secara lebih terstruktur dan terdokumentasi, sehingga sistem ini dapat dijadikan sebagai solusi praktis dalam mendukung proses manajemen layanan maintenance dan operasional pada lingkungan organisasi.

Kata kunci: Aplikasi Mobile, General Affairs, Layanan Maintenance, Layanan Operasional, Agile, Flutter

ABSTRACT

The increasing complexity of maintenance and operational services in General Affairs units requires an integrated and efficient information system, as conventional processes that rely on manual reporting and fragmented communication often lead to service delays, data inconsistency, and limited monitoring capabilities. This study aims to design and develop a Flutter-based mobile application to support the management of General Affairs maintenance and operational services, with PT PLN Indonesia Power UBP Banten 3 Lontar as the object of research. The research methodology adopts an Agile approach in software development, which includes requirement planning, system design, iterative development, and functional testing using the black-box testing method to ensure system reliability and compliance with user requirements. The developed application provides core features such as service request submission, operational activity reporting, task monitoring, and real-time service status tracking. As a measurable indicator of success, functional testing using the black-box method on 13 test cases resulted in a 100% success rate (pass status), proving that the application fully complies with user requirements. The results indicate that the application improves service management effectiveness, enhances information accessibility, and supports operational workflows in a more structured and documented manner. Therefore, the proposed system can be considered a practical solution for improving the management of maintenance and operational services in organizational environments.

Keywords: Mobile Application, General Affairs, Maintenance Service, Operational Service, Agile, Flutter.

1. PENDAHULUAN

Dalam era transformasi digital saat ini, efisiensi operasional menjadi faktor kritis dalam menjaga daya saing organisasi. Tuntutan untuk menyediakan layanan yang cepat, transparan, dan terukur tidak hanya berlaku pada layanan eksternal kepada pelanggan, tetapi juga pada proses internal yang mendukung kelancaran operasional perusahaan. Departemen General Affairs (GA) memiliki peran strategis dalam hal ini, karena bertanggung jawab atas pengelolaan fasilitas, pemeliharaan aset, dan berbagai layanan pendukung lainnya yang secara langsung memengaruhi produktivitas dan kenyamanan karyawan [1]. Implementasi teknologi digital dalam manajemen fasilitas terbukti dapat meningkatkan efisiensi operasional dan mempercepat waktu respons terhadap permintaan layanan secara signifikan [2].

PT PLN Indonesia Power UBP Banten 3 Lontar, sebagai salah satu unit pembangkit listrik strategis di Indonesia, memiliki kompleksitas operasional yang tinggi dengan aset tersebar di berbagai lokasi. Kelancaran fungsi GA sangat penting untuk mendukung kegiatan operasional inti perusahaan. Namun, proses layanan GA yang masih berjalan secara manual menimbulkan berbagai tantangan. Proses seperti pelaporan kerusakan fasilitas, permintaan layanan perbaikan, atau pengajuan kebutuhan operasional lainnya seringkali masih mengandalkan formulir fisik, komunikasi melalui email, atau komunikasi lisan yang tidak terpusat [3]. Kondisi ini sejalan dengan temuan bahwa proses manual dalam manajemen fasilitas dapat menyebabkan kehilangan data dan peningkatan biaya operasional karena duplikasi pekerjaan [4].

Ketergantungan pada proses manual menimbulkan berbagai permasalahan signifikan. Pertama, dari segi efisiensi, alur kerja menjadi lambat karena waktu terbuang untuk administrasi manual, rekapitulasi data, dan pelacakan status permintaan yang sulit dilakukan [5]. Kedua, masalah transparansi menjadi kendala utama di mana karyawan yang mengajukan permintaan seringkali tidak memiliki visibilitas yang jelas mengenai status tindak lanjut dari laporannya. Ketiga, kualitas dokumentasi menjadi lemah karena data historis perbaikan dan permintaan layanan tidak tercatat dengan baik, tersebar, dan sulit dianalisis, yang pada akhirnya menghambat pengambilan keputusan strategis terkait pemeliharaan aset dan alokasi sumber daya.

Aplikasi mobile telah terbukti menjadi solusi efektif untuk mengatasi tantangan dalam manajemen layanan. Penggunaan aplikasi mobile dalam facility management dapat meningkatkan produktivitas tim dan mengurangi waktu penyelesaian tugas secara substansial [6]. Framework Flutter, yang dikembangkan oleh Google, menawarkan keunggulan dalam pengembangan aplikasi cross-platform dengan performa mendekati aplikasi native. Flutter memungkinkan pengembangan aplikasi untuk platform iOS dan Android dari satu codebase, sehingga mengurangi waktu dan biaya pengembangan secara signifikan dibandingkan pengembangan native terpisah [7], [8]. Penggunaan Flutter juga memungkinkan fitur hot reload yang mempercepat proses pengembangan dan iterasi aplikasi [9].

Firebase sebagai platform Backend-as-a-Service (BaaS) menyediakan infrastruktur backend yang lengkap tanpa memerlukan pengelolaan server secara manual [10]. Firebase menawarkan berbagai layanan terintegrasi seperti Authentication untuk manajemen pengguna, Firestore Database untuk penyimpanan data real-time, Cloud Functions untuk logika bisnis sisi server, dan Cloud Storage untuk penyimpanan file [11], [12]. Integrasi Flutter dengan Firebase memungkinkan pengembangan aplikasi yang scalable dengan sinkronisasi data real-time antar perangkat, yang sangat penting untuk koordinasi tim dalam manajemen fasilitas [13].

Penelitian terkait aplikasi mobile untuk manajemen fasilitas telah menunjukkan berbagai manfaat dalam meningkatkan efisiensi operasional. Implementasi teknologi Building Information Modeling (BIM) yang terintegrasi dengan aplikasi mobile terbukti efektif dalam memastikan pemeliharaan preventif dilakukan tepat waktu dan membantu mencegah kegagalan peralatan [14]. Studi lain menunjukkan bahwa sistem manajemen pemeliharaan berbasis mobile dapat meningkatkan efisiensi pengelolaan data pemeliharaan dan mengurangi biaya pemeliharaan secara keseluruhan [15].

Meskipun berbagai penelitian sebelumnya telah berhasil menerapkan aplikasi mobile untuk manajemen fasilitas, sebagian besar studi tersebut masih berfokus hanya pada aspek pemeliharaan (maintenance) secara

terpisah. Kesenjangan penelitian (research gap) terlihat dari belum banyaknya sistem yang mengintegrasikan layanan pemeliharaan fasilitas dan operasional transportasi secara bersamaan di dalam satu platform terpusat. Selain itu, belum ada penelitian serupa yang menggabungkan pelacakan status secara real-time dengan mekanisme evaluasi kinerja teknisi maupun driver secara langsung (menggunakan sistem rating) untuk mengakomodasi kompleksitas General Affairs spesifik di lingkungan industri energi. Penelitian ini hadir secara eksplisit untuk mengisi kesenjangan tersebut.

Berdasarkan permasalahan yang telah diidentifikasi, penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi mobile terintegrasi bernama MasBro menggunakan framework Flutter dan Firebase sebagai solusi digital untuk meningkatkan efisiensi dan transparansi layanan GA di PT PLN Indonesia Power UBP Banten 3 Lontar. Fokus utama pengembangan aplikasi ini adalah pada dua layanan yang dapat mendukung operasional harian, yaitu Maintenance App untuk pelaporan dan perbaikan kerusakan fasilitas, serta Operational App untuk permintaan layanan transportasi operasional. Selain kedua layanan utama tersebut, aplikasi ini juga dilengkapi dengan modul dashboard admin yang berisi fitur pengelolaan akun pengguna secara penuh (CRUD). Sistem dirancang untuk memfasilitasi interaksi antar berbagai peran pengguna seperti Karyawan, Officer, Teknisi, dan Driver, dengan fitur pemantauan status secara real-time dan notifikasi otomatis.

Kontribusi utama dari penelitian ini meliputi perancangan dan implementasi sistem informasi yang mendigitalisasi proses manual pada bidang perawatan fasilitas dan dukungan transportasi operasional, penerapan arsitektur aplikasi mobile berbasis Flutter-Firebase yang dapat diadaptasi untuk organisasi sejenis, serta pengembangan mekanisme evaluasi kinerja melalui fitur penilaian teknisi dan fasilitas yang memungkinkan perbaikan berkelanjutan. Hasil penelitian ini diharapkan dapat menjadi rujukan bagi organisasi lain yang menghadapi tantangan dalam transformasi digital layanan pendukung operasional.

2. METODOLOGI

Penelitian ini menggunakan pendekatan Agile Scrum dalam pengembangan aplikasi mobile layanan General Affairs di PT PLN Indonesia Power UBP Banten 3 Lontar. Metodologi Scrum dipilih karena kemampuannya dalam mengelola pengembangan melalui siklus iteratif yang disebut sprint, yang memungkinkan evaluasi dan adaptasi berkala berdasarkan feedback pengguna [16]. Penelitian dilaksanakan di PT PLN Indonesia Power UBP Banten 3 Lontar selama dua bulan (1 November - 31 Desember 2024) di Bagian Efisiensi, Kinerja, dan Sistem Informasi (EKSIS).

2.1 Alat dan Teknologi

Penelitian ini menggunakan beragam alat dan teknologi yang dipilih secara khusus untuk mendukung seluruh tahapan proyek mulai dari analisis, perancangan, hingga pengembangan serta memfasilitasi kolaborasi tim. Pemilihannya didasarkan pada kebutuhan spesifik proyek dan mengikuti standar industri pengembangan aplikasi mobile modern.

2.1.1 Flutter dan Dart

Teknologi frontend aplikasi dikembangkan menggunakan Flutter SDK, sebuah framework open-source yang dikembangkan oleh Google untuk membangun aplikasi cross-platform dengan single codebase [7]. Flutter dipilih karena kemampuannya menghasilkan aplikasi dengan performa mendekati aplikasi native, fitur hot reload yang mempercepat proses pengembangan, serta ekosistem package yang lengkap melalui pub.dev [8]. Bahasa pemrograman yang digunakan adalah Dart, yang merupakan bahasa pemrograman object-oriented yang dioptimalkan untuk pengembangan user interface.

2.1.2 Firebase

Untuk backend, penelitian ini memanfaatkan Firebase sebagai platform Backend-as-a-Service (BaaS) yang menyediakan infrastruktur lengkap tanpa memerlukan pengelolaan server manual [10]. Firebase menyediakan beberapa layanan yang diintegrasikan dalam aplikasi, yaitu Firebase Authentication untuk manajemen autentikasi pengguna dengan metode email/password, Cloud Firestore sebagai basis data

NoSQL real-time yang memungkinkan sinkronisasi data secara otomatis antar perangkat, Cloud Functions untuk menjalankan logika backend berbasis serverless yang dipicu oleh event tertentu, dan Cloud Storage untuk penyimpanan file seperti gambar bukti kerusakan dan dokumen terkait.

2.1.3 Draw.io

Draw.io (diagrams.net) digunakan sebagai tools pembuatan diagram untuk memodelkan sistem menggunakan Unified Modeling Language (UML). Diagram yang dibuat meliputi use case diagram untuk menggambarkan interaksi antara aktor dengan sistem, activity diagram untuk memodelkan alur proses bisnis, dan Entity-Relationship Diagram (ERD) untuk menggambarkan struktur basis data beserta relasi antar entitas.

2.1.4 Visual Studio Code

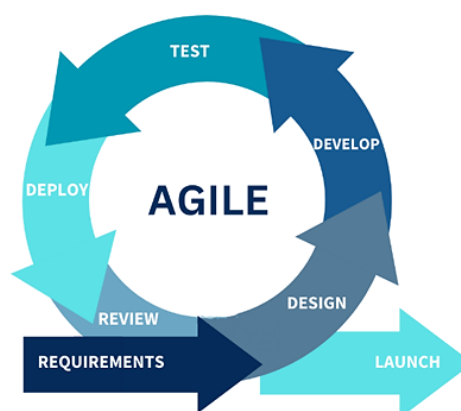
Visual Studio Code digunakan sebagai Integrated Development Environment (IDE) utama untuk proses pengembangan aplikasi. Visual Studio Code dipilih karena dukungannya yang kuat terhadap ekstensi Flutter dan Dart, fitur code completion, debugging yang terintegrasi, serta kemampuan hot reload yang memudahkan pengembangan iteratif [8]. IDE ini juga menyediakan terminal terintegrasi yang memfasilitasi eksekusi perintah Git dan manajemen package dependencies.

2.1.5 Git dan GitHub

Git digunakan sebagai sistem version control untuk mengelola kode sumber aplikasi, sedangkan GitHub dimanfaatkan sebagai platform hosting repository dan kolaborasi tim [17]. Git memungkinkan tim untuk melacak perubahan kode, mengelola branching untuk pengembangan fitur paralel, dan melakukan merging kode dari berbagai kontributor [18]. GitHub menyediakan fitur tambahan seperti pull request untuk code review, issue tracking untuk manajemen tugas, dan project board untuk memvisualisasikan progres pengembangan dalam konteks metodologi Scrum.

2.2 Metodologi Pengembangan Sistem

Pengembangan aplikasi mengadopsi framework Scrum yang merupakan metodologi Agile paling banyak digunakan dalam pengembangan aplikasi mobile [16], [19]. Scrum membagi proyek ke dalam siklus pengembangan pendek yang disebut sprint, yang biasanya berlangsung 1-4 minggu [19]. Setiap sprint bertujuan menghasilkan increment produk yang dapat diuji dan dievaluasi secara berkala berdasarkan feedback pengguna [20]. Metodologi penelitian ini mencakup dua aspek utama, yaitu alat dan teknologi yang digunakan dalam proses pengembangan aplikasi, serta penerapan metodologi Agile secara iteratif sebagai pendekatan pengembangan sistem. Gambar 1 menyajikan visualisasi alur metodologi penelitian, memperlihatkan tahapan pengembangan aplikasi dan peran masing-masing fase dalam mendukung keberhasilan proyek.



Gambar 1. Diagram Alur Metodologi Penelitian

2.2.1 Requirements

Tahap analisis kebutuhan dilakukan melalui wawancara dengan stakeholder untuk mengidentifikasi proses bisnis layanan General Affairs yang sedang berjalan. Observasi langsung terhadap alur kerja manual dilakukan untuk memahami kebutuhan setiap peran pengguna (Karyawan, Officer, Teknisi, Driver, dan Admin). Hasil analisis kemudian didokumentasikan dalam Product Backlog yang berisi daftar kebutuhan fungsional dan non-fungsional sistem [19]. Product Backlog mencakup empat layanan utama yaitu Maintenance App untuk pelaporan dan perbaikan kerusakan fasilitas, Resource/Item App untuk permintaan sumber daya, Operational App untuk transportasi, dan Booking Room App untuk pemesanan ruangan, serta fitur pendukung seperti autentikasi, pelacakan status real-time, notifikasi, dan sistem rating.

2.2.2 Design

Tahap perancangan sistem menggunakan diagram Unified Modeling Language (UML) untuk memodelkan struktur dan perilaku sistem. Use case diagram dibuat untuk setiap layanan utama guna menggambarkan interaksi antara aktor dengan sistem. Activity diagram digunakan untuk memodelkan alur proses bisnis dari pengajuan permintaan hingga penyelesaian. Entity-Relationship Diagram (ERD) dirancang untuk menggambarkan struktur basis data yang mencakup entitas utama seperti buildings, rooms, reports, tasks, drivers, vehicles, dan ride_requests beserta relasi antar entitas. Perancangan antarmuka pengguna (UI) dilakukan untuk membuat wireframe dan prototipe interaktif dengan mengikuti prinsip Material Design untuk memastikan konsistensi visual dan kemudahan penggunaan.

2.2.3 Development

Tahap pengembangan dilakukan secara iteratif dalam sprint berdasarkan Sprint Backlog yang telah ditetapkan [19]. Implementasi dimulai dengan pengembangan modul autentikasi menggunakan Firebase Authentication, dilanjutkan dengan pengembangan layanan-layanan utama secara paralel. Integrasi dengan Cloud Firestore dilakukan menggunakan package cloud_firestore yang menyediakan API reactive untuk sinkronisasi data real-time. Cloud Functions diimplementasikan menggunakan Node.js untuk mengirimkan notifikasi push melalui Firebase Cloud Messaging (FCM). Seluruh kode sumber dikelola menggunakan Git untuk version control dan disimpan di GitHub untuk kolaborasi tim [17], [18].

2.2.4 Testing

Pengujian sistem dilakukan menggunakan metode black-box testing untuk memverifikasi fungsionalitas aplikasi dari perspektif pengguna akhir [21]. Pengujian dilakukan terhadap setiap fitur utama aplikasi yang mencakup Maintenance App, Operational App, Resource App, dan Booking Room App, autentikasi pengguna, dashboard admin, serta integrasi antar modul. Setiap kategori pengujian didokumentasikan dalam test case yang mencakup deskripsi skenario, data input, langkah pengujian, hasil yang diharapkan, dan hasil aktual. Pengujian dilakukan pada emulator Android dan perangkat fisik untuk memastikan kompatibilitas lintas platform.

2.2.5 Deployment

Tahap penerapan dilakukan setelah aplikasi melewati seluruh pengujian dan mendapatkan persetujuan dari pihak Mentor Lapangan. Aplikasi di-build dalam format APK (Android Package Kit) untuk distribusi kepada pengguna di lingkungan PT PLN Indonesia Power UBP Banten 3 Lontar. Proses deployment mencakup konfigurasi Firebase project untuk lingkungan production, pengaturan security rules pada Cloud Firestore, dan distribusi aplikasi melalui instalasi langsung pada perangkat pengguna.

2.2.6 Review

Tahap evaluasi dilakukan melalui Sprint Review Meeting pada akhir setiap sprint untuk mempresentasikan increment produk kepada Mentor Lapangan dan stakeholder [19]. Fitur-fitur yang telah dikembangkan didemonstrasikan untuk mendapatkan feedback langsung. Selain itu, dilakukan juga Sprint Retrospective Meeting untuk mengevaluasi proses kerja dan kolaborasi tim [19], [20]. User Acceptance Testing (UAT) dilakukan dengan melibatkan pengguna dari berbagai peran untuk mendapatkan feedback mengenai fungsionalitas dan pengalaman pengguna. Hasil evaluasi digunakan untuk perbaikan pada sprint berikutnya dalam prinsip continuous improvement.

2.2.7 Metode Evaluasi

Evaluasi sistem dilakukan melalui dua metode utama, yaitu pengujian fungsional berbasis test case dan User Acceptance Testing (UAT). Pengujian fungsional menggunakan metode black-box testing dengan total 16 test case yang mencakup empat modul utama aplikasi (Layanan Maintenance, Layanan Operasional, Dashboard Admin, dan fitur Login). Setiap test case disusun berdasarkan skenario penggunaan nyata dengan kriteria keberhasilan yang terukur.

User Acceptance Testing (UAT) melibatkan 8 responden yang merepresentasikan tiga peran utama dalam sistem, yaitu 5 responden sebagai Karyawan yang bertugas mengajukan permintaan layanan, 1 responden sebagai Officer yang bertugas memverifikasi dan mengelola permintaan, serta 2 responden sebagai Teknisi yang menangani penugasan perbaikan. Kedelapan responden merupakan personel yang terlibat langsung dalam proses layanan General Affairs di unit tersebut, sehingga dapat memberikan evaluasi yang relevan terhadap fungsionalitas dan kegunaan aplikasi.

Instrumen evaluasi UAT mencakup tiga aspek utama. Pertama, kelengkapan fitur yang mengevaluasi apakah seluruh fungsi aplikasi (pengajuan permintaan, pemantauan status, notifikasi, dan sistem rating) dapat digunakan dengan baik. Kedua, kemudahan penggunaan yang mengukur apakah antarmuka aplikasi intuitif dan dapat dipahami pengguna tanpa pelatihan mendalam. Ketiga, kenyamanan antarmuka yang menilai tampilan visual, responsivitas aplikasi, dan pengalaman penggunaan secara keseluruhan.

Metrik keberhasilan yang digunakan meliputi tingkat keberhasilan test case (target 100% pass), tingkat kepuasan pengguna dalam UAT (diukur melalui feedback kualitatif terhadap tiga aspek evaluasi), dan kemampuan sistem dalam menangani skenario penggunaan real-world tanpa kendala signifikan. Kombinasi pengujian fungsional dan UAT memastikan bahwa aplikasi tidak hanya memenuhi spesifikasi teknis, tetapi juga dapat diterima dan digunakan dengan baik oleh pengguna akhir dalam lingkungan operasional sebenarnya.

3. HASIL DAN PEMBAHASAN

3.1 Analisis dan Perancangan Sistem

3.1.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan melalui wawancara terstruktur dengan pemangku kepentingan dan observasi langsung terhadap proses bisnis. Hasil analisis mengidentifikasi tiga permasalahan utama dalam sistem manual yang menjadi dasar pengembangan solusi digital. Pertama, ketidakefisienan waktu proses dimana rata-rata waktu penyelesaian satu permintaan maintenance dari pelaporan hingga verifikasi membutuhkan 2 hingga 3 hari kerja akibat proses administrasi manual dan komunikasi antar unit yang tidak tersinkronisasi. Kedua, minimnya transparansi status dimana pemohon tidak dapat mengetahui progres penyelesaian permintaan tanpa menghubungi bagian General Affairs secara langsung, menyebabkan penumpukan pertanyaan dan redundansi komunikasi. Ketiga, lemahnya sistem dokumentasi melalui pesan personal yang rentan hilang atau tidak tersimpan, mempersulit evaluasi kinerja dan analisis historis untuk pengambilan keputusan strategis.

Berdasarkan analisis tersebut, ditetapkan empat modul layanan utama yang terintegrasi dalam satu platform terpusat. Layanan Maintenance dirancang untuk mengelola pelaporan kerusakan fasilitas dengan fitur unggah foto bukti, sistem penugasan Teknisi berbasis kinerja, dan mekanisme rating untuk evaluasi berkelanjutan. Layanan Operasional dirancang untuk mengelola permintaan transportasi dengan validasi ketersediaan Driver dan kendaraan secara otomatis, dilengkapi fitur rating terpisah untuk Driver dan kendaraan guna mendukung penjadwalan pemeliharaan preventif. Dashboard Admin dirancang untuk pengelolaan data pengguna dengan kontrol akses berbasis peran yang ketat. Fitur Login mengimplementasikan autentikasi Firebase dengan routing otomatis berdasarkan peran pengguna untuk memastikan keamanan dan pengalaman pengguna yang disesuaikan.

Kebutuhan fungsional dan non-fungsional sistem dirangkum dalam Tabel 1 dan Tabel 2. Kebutuhan fungsional mencakup kemampuan sistem dalam menangani seluruh alur proses bisnis dari pengajuan hingga evaluasi, sedangkan kebutuhan non-fungsional menetapkan standar kinerja yang harus dipenuhi untuk memastikan sistem dapat beroperasi secara optimal di lingkungan production. Penetapan kebutuhan non-fungsional ini penting untuk memastikan bahwa sistem tidak hanya berfungsi dengan benar, tetapi juga memberikan pengalaman pengguna yang memuaskan dan dapat diandalkan dalam operasional sehari-hari.

Tabel 1. Ringkasan Kebutuhan Fungsional Sistem

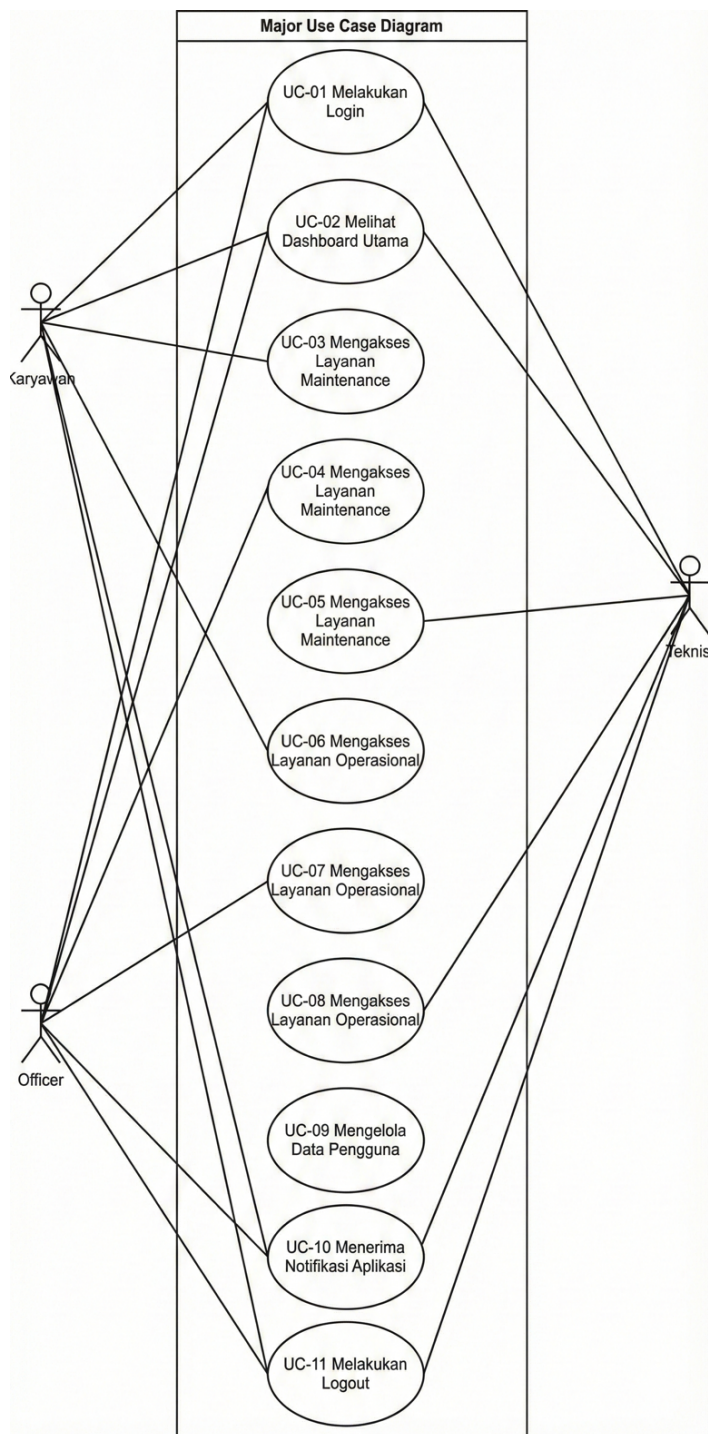
Modul	Deskripsi
Autentikasi Pengguna	Sistem menyediakan form login untuk autentikasi dan secara otomatis mengarahkan pengguna ke dashboard sesuai perannya.
Layanan Maintenance (Karyawan)	Sistem memungkinkan karyawan melapor kerusakan fasilitas, memantau status, dan memberi penilaian kinerja teknisi setelah selesai.
Layanan Maintenance (Officer)	Sistem memungkinkan officer memverifikasi laporan, menugaskan teknisi (in progress), dan memantau hasil penilaian teknisi.
Layanan Maintenance (Teknisi)	Sistem memungkinkan teknisi melihat tugas, memperbarui status, dan mengunggah foto bukti serta catatan saat pekerjaan selesai (done).
Layanan Operasional (Karyawan)	Sistem memungkinkan karyawan mengajukan permintaan kendaraan operasional, memantau status, dan menilai layanan pengemudi serta kendaraan.
Layanan Operasional (Officer)	Sistem memungkinkan officer memverifikasi permintaan, menugaskan armada dan pengemudi (assigned), serta memantau data penilaian layanan.
Layanan Operasional (Pengemudi)	Sistem memungkinkan pengemudi melihat detail tugas, memulai (on trip), dan menyelesaikan perjalanan (completed) disertai catatan.
Fitur Dashboard Admin	Sistem memungkinkan admin mengelola penuh data pengguna (tambah, ubah, hapus akun), termasuk mengatur peran dan kredensial login.

Tabel 2. Ringkasan Kebutuhan Non-Fungsional Sistem

No	Kebutuhan Non-Fungsional
1	Sistem dilengkapi autentikasi menggunakan Firebase Authentication.
2	Sistem mampu menampilkan data secara real-time.
3	Sistem dapat digunakan secara stabil pada perangkat Android.
4	Antarmuka sistem mudah dipahami oleh pengguna.
5	Sistem mudah dikembangkan dan diperbaiki.

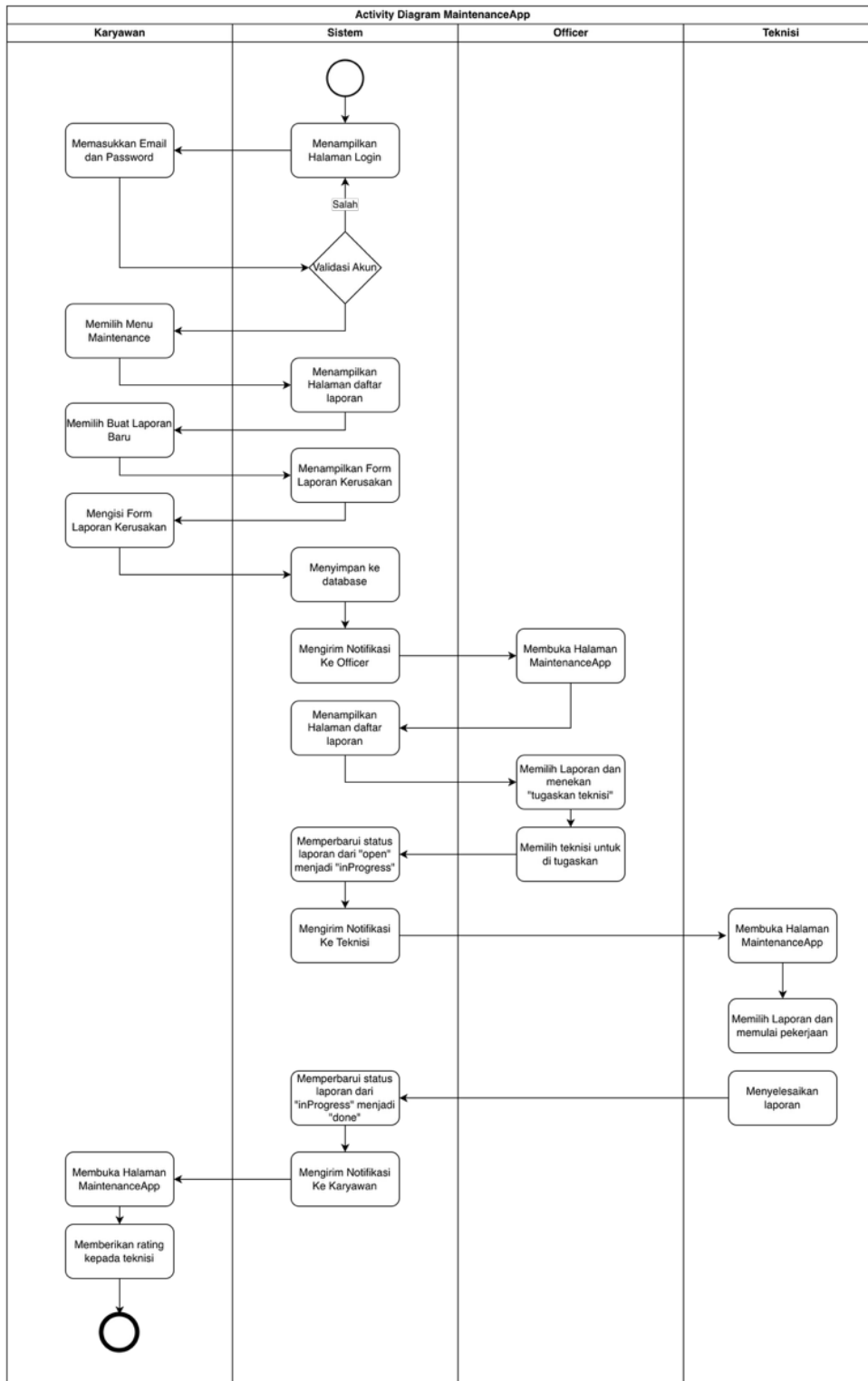
Sistem dirancang dengan lima aktor yaitu Karyawan, Officer, Teknisi, Driver, dan Admin. Use Case Diagram menggambarkan interaksi fungsional antara aktor dan sistem, mencakup proses autentikasi, pengelolaan layanan maintenance, layanan operasional, manajemen pengguna, dan login aplikasi. Selain itu, Activity Diagram digunakan untuk memvisualisasikan alur proses bisnis atau langkah-langkah secara kronologis yang terlibat dalam setiap layanan sistem, seperti proses pelaporan hingga penyelesaian permintaan. Entity Relationship Diagram memodelkan struktur basis data dengan entitas utama Users, Buildings, Rooms, Reports, Tasks untuk maintenance, serta Vehicles, Drivers, dan Ride_Requests untuk layanan operasional.

Gambar 2 menggambarkan Major Use Case Diagram yang memberikan gambaran keseluruhan fungsi sistem secara menyeluruh. Diagram ini mencakup proses autentikasi pengguna, akses ke layanan General Affairs, seperti layanan Maintenance App dan Operational App, serta fungsi manajemen pengguna oleh Admin.



Gambar 2. Major Use Case Diagram

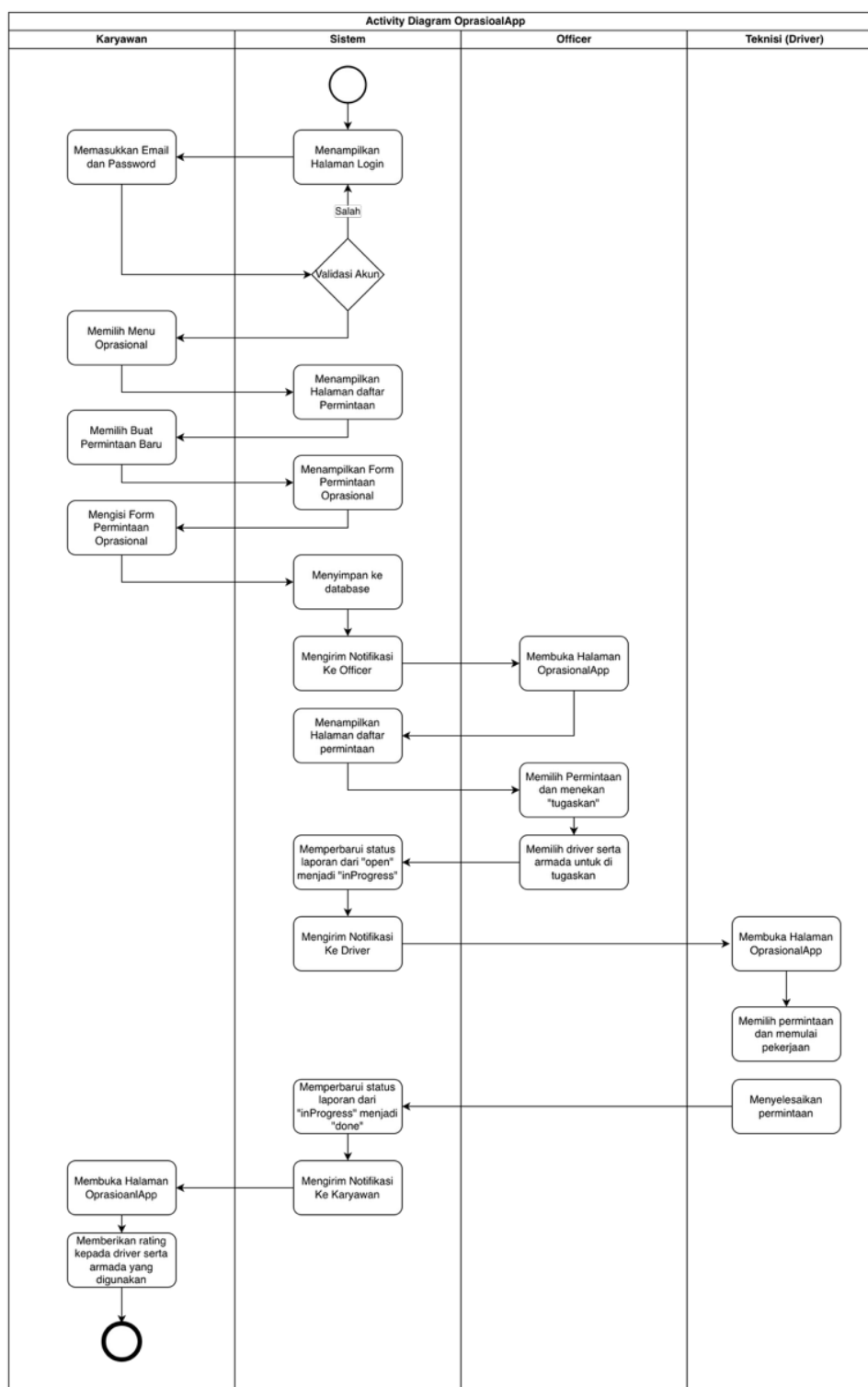
Gambar 3 menunjukkan Activity Diagram Layanan MaintenanceApp, yang mendokumentasikan alur kegiatan operasional terkait pelaporan kerusakan fasilitas dan proses perbaikannya. Diagram ini mencakup seluruh tahapan proses secara kronologis antar aktor dan sistem, mulai dari Karyawan yang melakukan login dan mengajukan laporan kerusakan baru, proses penyimpanan data dan pengiriman notifikasi oleh Sistem, langkah verifikasi serta penugasan Teknisi oleh Officer, eksekusi penyelesaian perbaikan oleh Teknisi, hingga diakhiri dengan pemberian penilaian (rating) kinerja Teknisi oleh Karyawan setelah Sistem memperbarui status menjadi selesai.



Gambar 3. Activiry diagram Aplikasi Maintenance

Gambar 4 menunjukkan Activity Diagram Layanan OperasionalApp, yang mendokumentasikan alur kegiatan terkait permintaan kendaraan operasional untuk layanan antar-jemput karyawan. Diagram ini mencakup seluruh proses, mulai dari pengajuan permohonan perjalanan oleh Karyawan, verifikasi serta penugasan armada kendaraan dan pengemudi (Driver) oleh Officer, pelaksanaan dan penyelesaian tugas

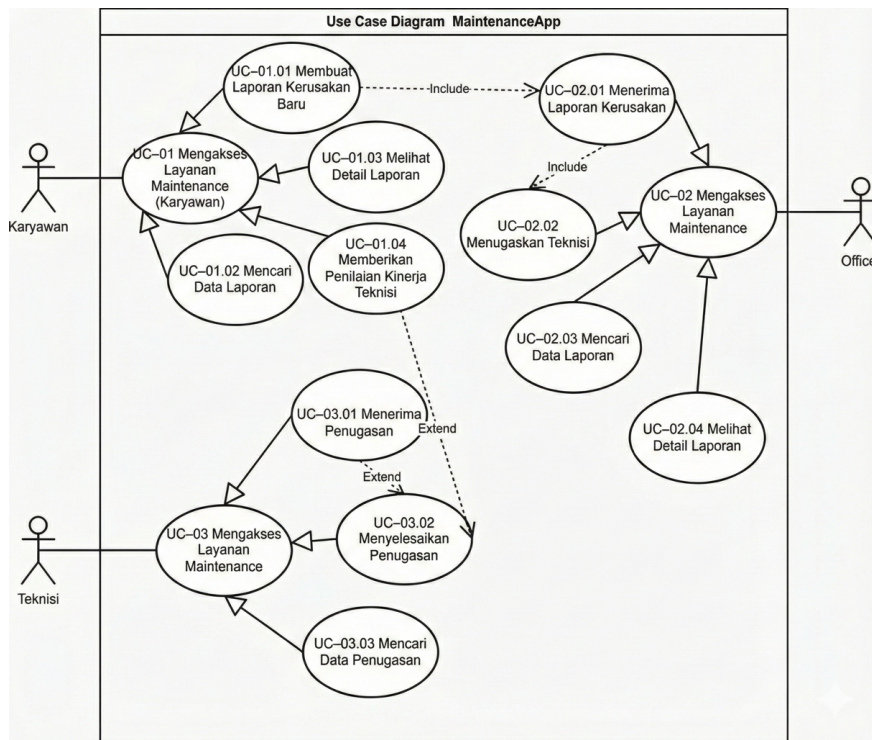
pengantaran oleh Driver, hingga diakhiri dengan pemberian penilaian (rating) terhadap pelayanan pengemudi dan kondisi kendaraan oleh Karyawan setelah perjalanan selesai.



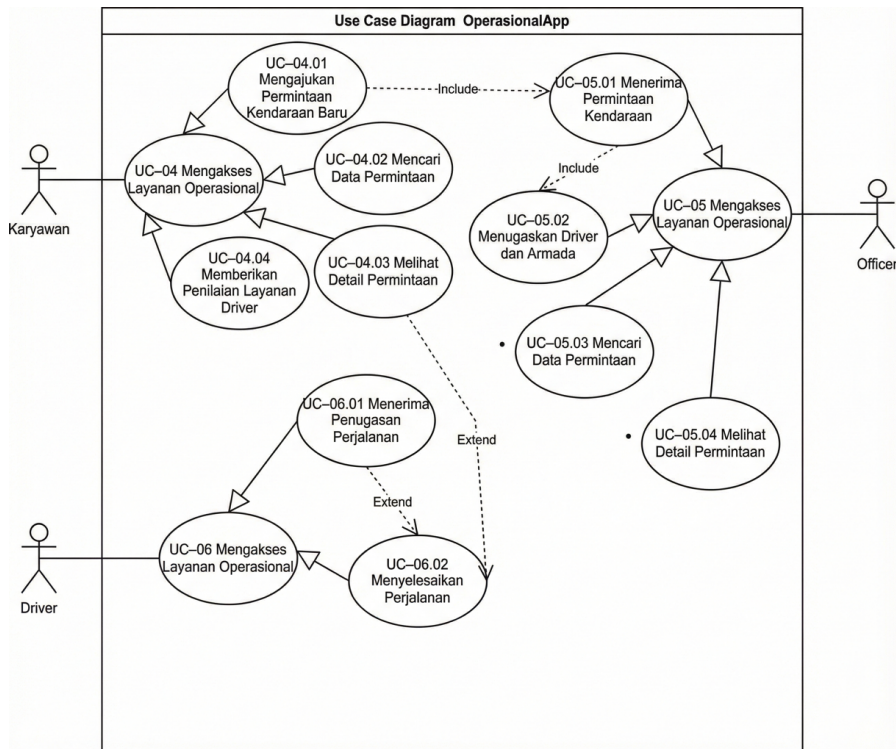
Gambar 4. Activity Diagram Aplikasi Oprasional

Gambar 5 menunjukkan Use Case Diagram Layanan Maintenance, yang mendokumentasikan fungsi-fungsi terkait pengelolaan pelaporan kerusakan fasilitas dan perbaikannya. Diagram ini mencakup seluruh proses,

mulai dari pengajuan laporan kerusakan oleh Karyawan, verifikasi dan penugasan Teknisi oleh Officer, penanganan perbaikan oleh Teknisi, hingga pemberian penilaian kinerja Teknisi oleh Karyawan setelah perbaikan selesai.



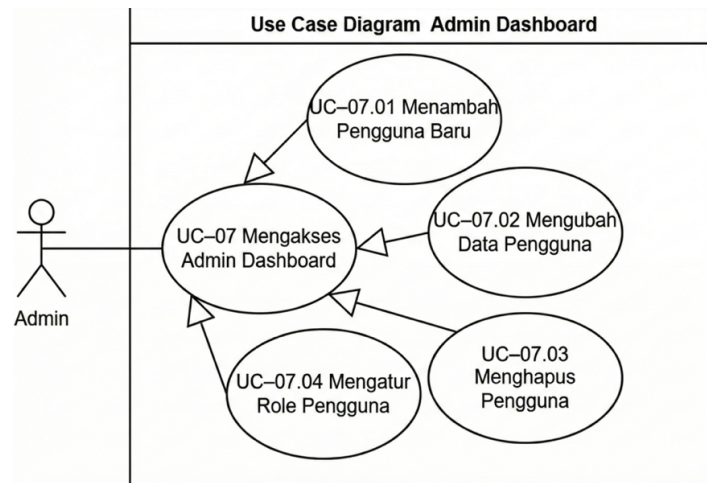
Gambar 5. Use Case Diagram Layanan Maintenance



Gambar 6. Use Case Diagram Layanan Operasional

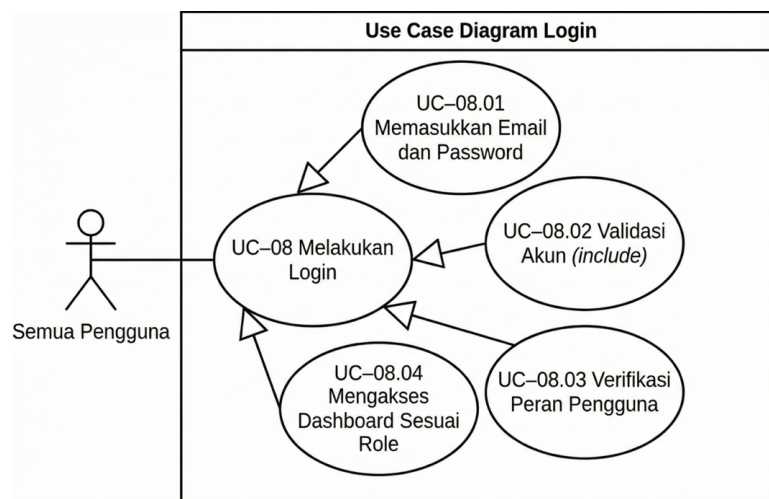
Gambar 6 menggambarkan Use Case Diagram Layanan Operasional, yang menunjukkan interaksi antara Karyawan, Officer, dan Driver dalam proses permintaan transportasi operasional. Diagram ini mencakup pengajuan permintaan kendaraan oleh Karyawan, penugasan Driver dan kendaraan oleh Officer, pelaksanaan tugas transportasi oleh Driver, serta pemberian penilaian terhadap kinerja Driver dan kondisi kendaraan oleh Karyawan.

Gambar 7 menampilkan Use Case Diagram Dashboard Admin, yang menggambarkan fungsi-fungsi pengelolaan data pengguna oleh Admin. Diagram ini mencakup proses pembuatan akun pengguna baru, pengeditan data pengguna yang sudah ada, penghapusan akun pengguna.



Gambar 7. Use Case Diagram Dashboard Admin

Gambar 8 menampilkan Use Case Diagram Login Aplikasi, yang menggambarkan serangkaian langkah yang dilakukan pengguna untuk mengakses sistem aplikasi MasBro. Diagram ini mencakup proses memasukkan email dan password, validasi akun yang dilakukan secara otomatis oleh sistem, verifikasi peran pengguna untuk menentukan hak akses yang sesuai, serta mengarahkan pengguna ke dashboard utama berdasarkan perannya masing-masing.

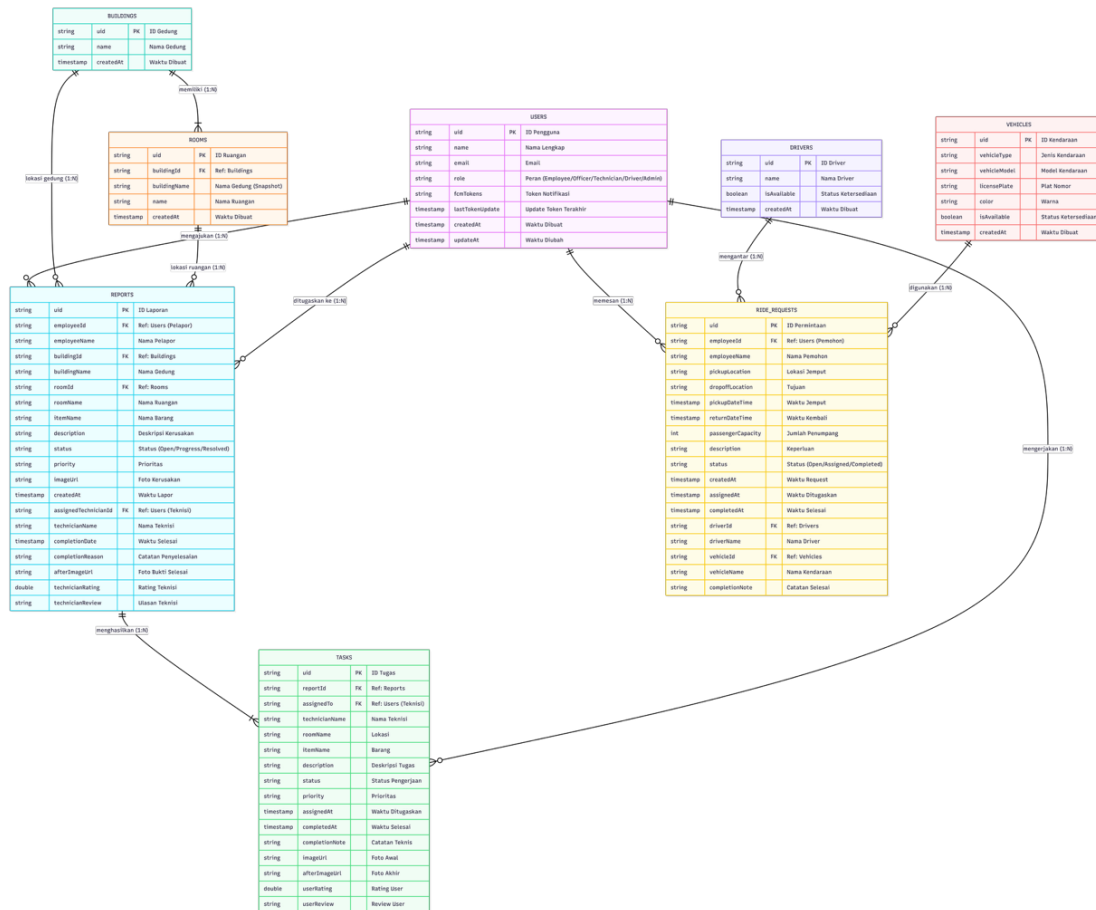


Gambar 8. Use Case Diagram Login Aplikasi

3.1.2 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) dirancang untuk memodelkan hubungan antar entitas dalam sistem dan menjadi panduan dalam perancangan struktur basis data pada Firebase Firestore. Diagram ini menggambarkan entitas-entitas utama dalam aplikasi MasBro beserta atribut, relasi, dan kardinalitasnya.

Gambar 9 menunjukkan ERD aplikasi MasBro yang mencakup beberapa entitas utama. Entitas Users menyimpan data identitas dan peran pengguna, termasuk informasi untuk autentikasi dan notifikasi. Entitas Buildings merepresentasikan gedung-gedung yang ada di perusahaan, sedangkan entitas Rooms menyimpan data ruangan yang tersedia beserta kapasitasnya. Entitas Reports menghimpun laporan kerusakan fasilitas yang diajukan oleh Karyawan pada layanan maintenance, yang kemudian terhubung dengan entitas Tasks yang berisi detail penugasan untuk Teknisi.



Gambar 9. E-R Diagram Layanan Maintenance Dan Layanan Oprasioanal

Untuk layanan operasional, terdapat entitas Vehicles yang menyimpan data kendaraan operasional perusahaan, entitas Drivers yang berisi informasi driver yang tersedia, dan entitas Ride_Requests yang merepresentasikan permintaan transportasi dari Karyawan. Setiap entitas memiliki relasi yang jelas dengan kardinalitas tertentu, seperti relasi one-to-many antara Buildings dengan Rooms, relasi antara Reports dengan Tasks, serta relasi antara Users dengan berbagai entitas lainnya.

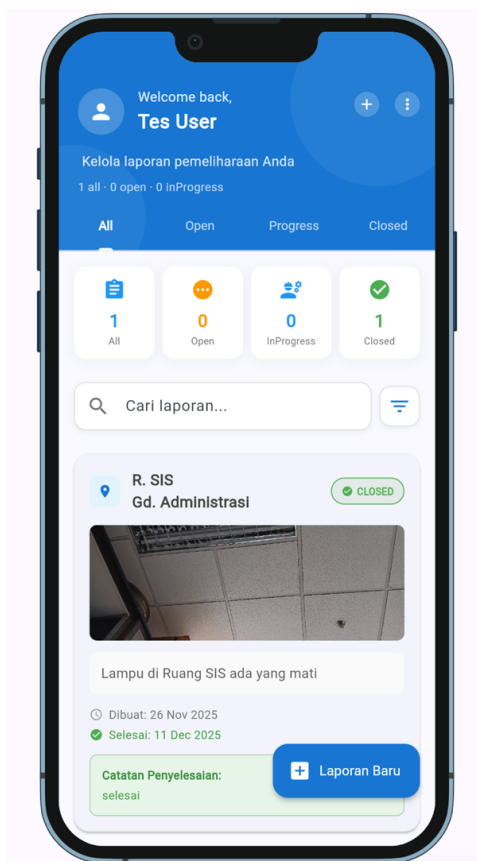
3.2 Implementasi Aplikasi

Implementasi aplikasi MasBro dilakukan secara modular mengikuti fitur-fitur utama yang telah dirancang pada tahap pemodelan. Setiap modul dikembangkan menggunakan Flutter dan Dart di sisi frontend, dengan integrasi Firebase sebagai backend untuk penyimpanan data, autentikasi, dan logika bisnis. Implementasi mencakup empat modul utama yaitu Layanan Maintenance, Layanan Operasional, Dashboard Admin, dan Fitur Login. Hasil implementasi ditampilkan melalui antarmuka pengguna yang telah berhasil dikembangkan untuk setiap modul.

3.2.1 Layanan Maintenance

Layanan Maintenance merupakan modul utama yang mengelola seluruh proses pelaporan kerusakan fasilitas dan penanganan perbaikannya. Modul ini mencakup alur lengkap dari pelaporan kerusakan oleh Karyawan, verifikasi dan penugasan Teknisi oleh Officer, pelaksanaan perbaikan oleh Teknisi, hingga pemberian penilaian kinerja Teknisi setelah perbaikan selesai.

Gambar 10 menunjukkan Dashboard Karyawan Layanan Maintenance yang menampilkan daftar seluruh laporan kerusakan yang telah diajukan. Dashboard ini dilengkapi dengan fitur pencarian untuk memudahkan Karyawan menemukan laporan tertentu, serta tombol untuk menambahkan laporan kerusakan baru. Setiap laporan ditampilkan dengan informasi status terkini, seperti Open, In Progress, atau Done.

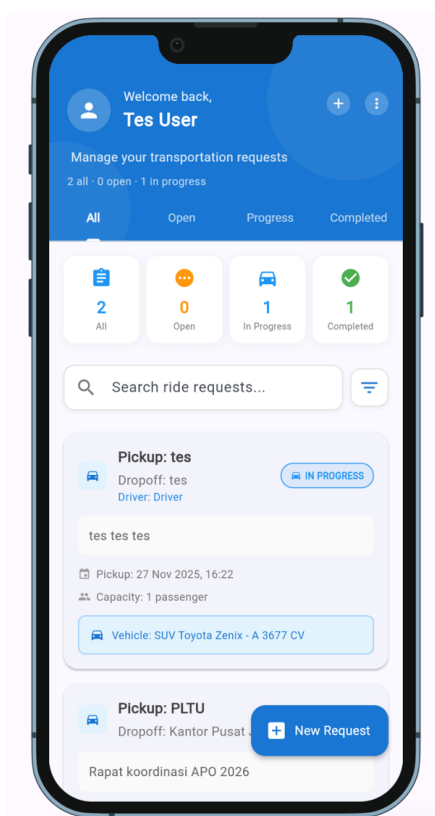


Gambar 10. Tampilan Dashboard Karyawan Layanan Maintenance

3.2.2 Layanan Operasional

Layanan Operasional merupakan modul yang mengelola proses permintaan transportasi operasional perusahaan. Modul ini dirancang untuk memfasilitasi pengajuan kebutuhan kendaraan oleh Karyawan, penugasan Driver dan kendaraan oleh Officer, pelaksanaan tugas transportasi, serta evaluasi kinerja Driver dan kondisi kendaraan.

Gambar 11 menunjukkan Dashboard Karyawan Layanan Operasional yang menampilkan daftar permintaan transportasi yang telah diajukan. Dashboard ini dilengkapi dengan fitur pencarian dan tombol untuk mengajukan permintaan transportasi baru. Status setiap permintaan ditampilkan secara jelas untuk memudahkan pemantauan, seperti Pending, Approved, In Progress, atau Completed.

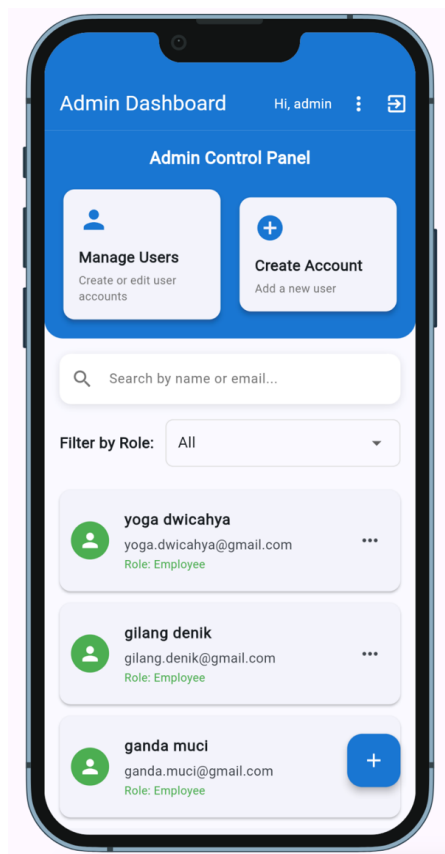


Gambar 11. Tampilan Dashboard Karyawan Layanan Operasional

3.2.3 Fitur Dashboard Admin

Dashboard Admin merupakan modul yang menyediakan antarmuka untuk mengelola data pengguna sistem serta data pendukung operasional aplikasi. Fitur ini hanya dapat diakses oleh pengguna dengan peran Admin dan mencakup fungsi-fungsi untuk membuat, mengedit, dan menghapus data pengguna.

Gambar 12 menunjukkan Tampilan Dashboard Admin yang menampilkan daftar seluruh pengguna terdaftar dalam sistem. Dashboard ini dilengkapi dengan fitur pencarian berdasarkan nama atau peran pengguna, serta tombol aksi untuk menambah pengguna baru, mengedit data pengguna yang sudah ada, atau menghapus akun pengguna. Informasi yang ditampilkan mencakup nama, peran (role), dan status akun masing-masing pengguna.

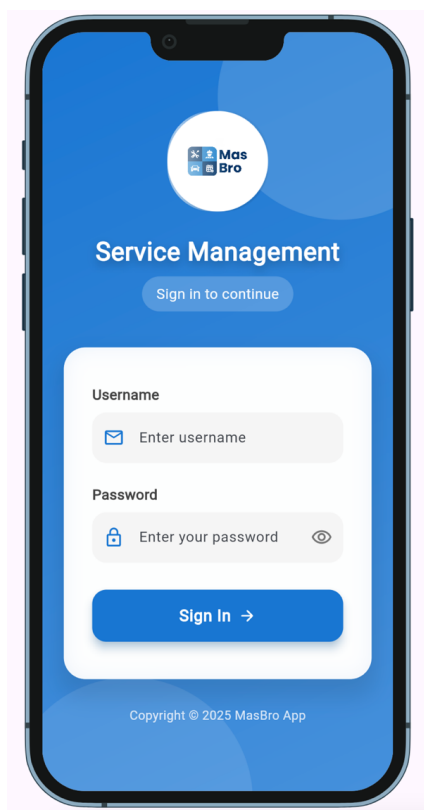


Gambar 12. Tampilan Dashboard Admin

3.2.4 Fitur Login

Fitur Login merupakan pintu masuk utama ke dalam aplikasi MasBro yang mengimplementasikan autentikasi pengguna menggunakan Firebase Authentication. Fitur ini memastikan bahwa hanya pengguna yang terdaftar dan memiliki kredensial valid yang dapat mengakses sistem, serta mengarahkan pengguna ke halaman yang sesuai berdasarkan peran mereka.

Gambar 13 menunjukkan Tampilan Login Aplikasi MasBro yang menyediakan input field untuk email dan password. Tampilan ini dirancang dengan antarmuka yang sederhana dan intuitif, dilengkapi dengan tombol login untuk memulai proses autentikasi. Selain itu, terdapat opsi untuk menampilkan atau menyembunyikan password guna meningkatkan kenyamanan pengguna saat memasukkan kredensial.



Gambar 13. Tampilan Login Aplikasi MasBro

3.3 Hasil Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur yang telah dikembangkan berfungsi sesuai dengan spesifikasi dan kebutuhan yang telah ditetapkan. Pengujian dilaksanakan melalui dua tahapan utama, yaitu pengujian fungsional berbasis test case dan User Acceptance Testing (UAT) untuk memvalidasi kelayakan aplikasi dalam lingkungan operasional.

3.3.1 Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan metode black-box testing berbasis test case. Setiap test case disusun berdasarkan alur kerja yang telah dirancang pada tahap pemodelan sistem, mencakup skenario positif dan negatif untuk memastikan sistem merespons dengan benar terhadap berbagai input dari pengguna. Pengujian mencakup seluruh modul aplikasi dengan total 13 test case.

Tabel 1 berikut menyajikan ringkasan hasil pengujian fungsional dari seluruh modul aplikasi MasBro.

Tabel 1. Ringkasan Hasil Pengujian Fungsional

No	Modul yang Diuji	Jumlah Test Case	Skor
1	Layanan Maintenance	4	Pass
2	Layanan Operasional	4	Pass
3	Fitur Dashboard Admin	4	Pass
4	Fitur Login Aplikasi	1	Pass
Total		13	Pass (100%)

Pengujian fungsional aplikasi MasBro dilakukan melalui 13 skenario uji (test case) yang mencakup empat fitur utama. Pada Layanan Maintenance dan Operasional, masing-masing dijalankan 4 pengujian untuk memastikan kelancaran alur sistem, mulai dari pembuatan pengajuan, validasi pengisian form, penugasan oleh Officer, hingga proses penilaian kinerja Teknisi dan Driver. Selanjutnya, 4 pengujian pada Dashboard Admin difokuskan untuk menguji kelancaran pengelolaan data akun dan hak akses. Terakhir, 1 pengujian pada fitur Login dilakukan guna memastikan keamanan autentikasi dan ketepatan arah halaman (redirect) sesuai peran pengguna. Secara keseluruhan, ke-13 pengujian tersebut berhasil lolos dengan status pass. Hasil ini membuktikan bahwa sistem aplikasi MasBro telah berfungsi dengan baik, mampu merespons setiap input secara tepat, dan memenuhi seluruh kebutuhan fungsional yang ditargetkan.

Sentralisasi proses layanan General Affair (GA) ke dalam aplikasi MasBro tidak hanya berhasil merubah format fisik menjadi digital, tetapi secara fundamental mengatasi masalah latensi dan fragmentasi data. Keberhasilan efisiensi ini sejalan dengan penelitian Taufikkurahman dan Yudoko (2025) yang menekankan bahwa implementasi transformasi digital pada layanan General Affairs terbukti efektif dalam meningkatkan efektivitas operasional dan mereduksi hambatan administrasi. Jika sebelumnya riwayat perbaikan aset dan penggunaan kendaraan operasional sulit dilacak karena mengandalkan komunikasi lisan dan grup pesan singkat, kini sistem menyediakan rekam jejak (audit trail) yang terstruktur. Tata kelola fasilitas yang terukur dan transparan melalui otomatisasi alur kerja (workflow) ini mengonfirmasi temuan Yousefli dkk. (2020), di mana penerapan sistem manajemen fasilitas yang terotomatisasi sangat krusial dalam mengatasi celah komunikasi operasional dan mengoptimalkan pemeliharaan aset di PT PLN Indonesia Power UBP Banten 3 Lontar.

Tabel 2 menyajikan analisis komparatif antara proses manual dan sistem digital yang menjadi dasar untuk mengukur dampak implementasi aplikasi MasBro terhadap efisiensi operasional.

Tabel 2. Analisis Komparatif Sistem Manual dengan Sistem Digital

Asepek	Sistem Manual	Sistem Digital
Waktu Proses Pelaporan	5-10 menit (mengisi formulir melalui pesan personal)	2-3 menit (form digital dengan validasi otomatis)
Waktu Verifikasi Officer	1-3 hari kerja (menunggu approval manual)	10-15 menit (verifikasi digital real-time)
Transparansi Status	Tidak ada (harus menghubungi GA secara langsung)	Real-time tracking dengan notifikasi push otomatis
Dokumentasi	Arsip fisik (rentan hilang, sulit dicari)	Database digital terstruktur dengan backup otomatis
Evaluasi Kinerja	Tidak terstruktur, subyektif	Rating system terukur dengan agregasi data otomatis
Koordinasi Antar Unit	Email/telepon (tidak terintegrasi)	Notifikasi push terintegrasi dalam satu platform

3.3.2 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan tahap pengujian akhir yang bertujuan untuk memvalidasi fungsionalitas aplikasi di lingkungan operasional nyata pada PT PLN Indonesia Power UBP Banten 3 Lontar. Pengujian ini melibatkan 8 responden yang merupakan personel aktif dalam alur layanan General Affairs. Secara rincian, responden terdiri dari 5 orang Karyawan yang bertugas menguji kelancaran alur pengajuan permintaan layanan maintenance maupun operasional serta pemberian penilaian (rating). Selain itu, terdapat 1 orang Officer yang menguji proses verifikasi permintaan, pengelolaan data, dan penugasan, serta 2 orang Teknisi/Driver yang berfokus menguji penerimaan tugas, pembaruan status pengerjaan, hingga penyelesaian tugas.

Setiap responden menjalankan skenario penggunaan secara langsung untuk memvalidasi tiga dimensi utama aplikasi. Dimensi pertama adalah kelengkapan fitur, yang memastikan seluruh fungsi utama seperti

pengajuan, pengelolaan, penugasan, penyelesaian, dan notifikasi real-time dapat beroperasi dengan baik. Dimensi kedua berfokus pada kemudahan penggunaan, guna menilai apakah antarmuka aplikasi tergolong intuitif dan alur sistemnya mudah dipahami tanpa memerlukan pelatihan mendalam. Sementara itu, dimensi ketiga adalah kenyamanan antarmuka, yang mengevaluasi kualitas visual, responsivitas sistem, dan pengalaman pengguna secara keseluruhan.

Berdasarkan hasil UAT yang dilaksanakan, aplikasi MasBro secara keseluruhan memenuhi ekspektasi pengguna dalam hal kelengkapan fitur dan kemudahan penggunaan. Pengguna berhasil melakukan seluruh alur utama dengan baik, mulai dari login berdasarkan peran, pengajuan dan pemantauan permintaan layanan maintenance dan operasional, pengelolaan penugasan, hingga penerimaan notifikasi status secara real-time. Tidak ditemukan kendala signifikan yang menghalangi pengguna untuk menyelesaikan tugas mereka pada saat pengujian. Hasil UAT mengonfirmasi bahwa aplikasi MasBro telah siap untuk digunakan dalam lingkungan operasional sesungguhnya di PT PLN Indonesia Power UBP Banten 3 Lontar.

4. KESIMPULAN

Penelitian ini telah berhasil merancang dan mengimplementasikan aplikasi layanan General Affairs bernama MasBro menggunakan framework Flutter, yang secara fundamental mengubah paradigma tata kelola pemeliharaan aset dan operasional kendaraan di PT PLN Indonesia Power UBP Banten 3 Lontar. Kontribusi ilmiah utama dari penelitian ini terletak pada pembuktian empiris bahwa penerapan arsitektur Backend as a Service (BaaS) bersinkronisasi real-time terbukti efektif dalam mengeliminasi asimetri informasi dan latensi komunikasi dalam manajemen fasilitas industri.

Jika pendekatan konvensional sebelumnya menghasilkan data yang terfragmentasi pada berbagai saluran komunikasi tidak resmi, arsitektur sistem ini menawarkan model pemusatan data (sentralisasi) yang menyediakan rekam jejak operasional (audit trail) yang utuh dan transparan. Interaksi antar aktor mulai dari Karyawan, Officer, hingga eksekutor teknis (Teknisi dan Driver) dapat diakomodasi secara instan dalam satu ekosistem yang seragam tanpa kendala lintas platform.

Hasil validasi yang ditunjukkan oleh pencapaian 100% pada pengujian black-box dan tingkat keberterimaan yang komprehensif pada User Acceptance Testing (UAT) menegaskan bahwa rancangan antarmuka yang konsisten berhasil menekan kurva pembelajaran (learning curve) pengguna. Secara keilmuan dan praktis, arsitektur penyelesaian masalah (problem-solving architecture) yang ditawarkan dalam penelitian ini dapat dijadikan blueprint rujukan bagi pengembangan sistem informasi layanan tata kelola korporat (Corporate Shared Services) di berbagai institusi atau perusahaan skala besar lainnya yang membutuhkan tingkat responsivitas operasional yang tinggi.

DAFTAR PUSTAKA

- [1] Taufikkurahman and G. Yudoko, "Increasing the Effectiveness and Customer Satisfaction of General Affairs Through the Implementation of Digital Transformation; Case Study : Indobara Smart Services at PT Borneo Indobara," *Asian Journal of Engineering, Social and Health*, vol. 4, no. 5, Jun. 2025, doi: 10.46799/ajesh.v4i5.614.
- [2] O. O. AHAMIOJE and A. R. ADEDOKUN, "Assessment of Factors Influencing Digital Transformation in Hotels' Facility Management in Abuja Metropolis, Nigeria," *IJRSI Vol 12 Issue 8*, vol. 12, no. 8, pp. 1627–1633, Sep. 2025, doi: 10.51244/ijrsi.2025.120800142.
- [3] A. Adozuka Aliu, N. Rima Muhamad Ariff, S. Ali Azzran Sh Said, D. Senanu Ametefe, and D. John, "Barriers to Digital Transformation in Facilities Management: Insights from a Developing Country," *Journal of Construction in Developing Countries*, vol. 30, no. 2, pp. 137–168, 2025, doi: 10.21315/jcdc.2025.30.2.6.
- [4] M. Ensafi and W. Thabet, "Challenges and Gaps in Facility Maintenance Practices," *EPiC Series in Built Environment*, 2021.

- [5] Z. Yousefli, F. Nasiri, and O. Moselhi, "Maintenance workflow management in hospitals: An automated multi-agent facility management system," *Journal of Building Engineering*, vol. 32, p. 101431, Nov. 2020, doi: 10.1016/j.jobbe.2020.101431.
- [6] J. Ruby, L. Gustavo, and N. Patricia, "Implementation of a mobile application for the management of maintenance tasks and documentation in automotive workshops Implementación de una aplicación móvil para la gestión de tareas y documentación de mantenimiento en talleres automotrices," *Journal Applied Computing*, vol. 8, 2009, doi: 10.35429/JCA.2024.8.22.5.8.
- [7] A. Sopandi, A. R. Hannan, H. Khotimah, C. Responden, and K. Kunci, "PERANCANGAN APLIKASI MOBILE MENGGUNAKAN FRAMEWORK FLUTTER PADA SISTEM INFORMASI AKADEMIK," *JIKA (Jurnal Informatika)*, vol. 8, no. 3, pp. 304–310, Jul. 2024, doi: 10.31000/jika.v8i3.11402.
- [8] V. R. Goli, "Cross-Platform Mobile Development: Comparing React Native and Flutter, and Accessibility in React Native," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 11, no. 3, 2023, doi: 10.15680/IJRCCE.2023.1103002.
- [9] M. A. Izzuddin, A. Andri, and H. Hardiyansyah, "Leveraging Prototype Method for Designing Tajweed Mobile Based Learning," *Journal of Information Systems and Informatics*, vol. 5, no. 2, pp. 615–629, May 2023, doi: 10.51519/journalisi.v5i2.488.
- [10] P. R. Saraf, S. M. Jadhao, S. J. Wanjari, S. G. Kolwate, P. Ankush, and D. Patil, "A Review on Firebase (Backend as A Service) for Mobile Application Development," vol. 10, 2022, doi: 10.22214/ijraset.2022.39958.
- [11] I. Kaur, "International Journal of Computer Science and Mobile Computing Mobile Cloud Computing: Using Firebase Auth," *International Journal of Computer Science and Mobile Computing*, vol. 11, pp. 61–68, 2022, doi: 10.47760/ijcsmc.2022.v11i04.008.
- [12] V. Kumar and T. Kaur, "Cloud Functions and Serverless Computing," vol. 10, 2022, doi: 10.22214/ijraset.2022.43163.
- [13] F. D. Astuti and Y. Hermawan, "PEMANFAATAN FIREBASE REALTIME DATABASE PADA APLIKASI PEMBELAJARAN AGAMA ISLAM MENGGUNAKAN FRAMEWORK FLUTTER," *Jurnal Informatika Komputer, Bisnis dan Manajemen*, vol. 18, no. 2, pp. 1–11, Jun. 2020, doi: 10.61805/fahma.v18i2.65.
- [14] A. Ehab, M. A. Mahdi, and A. El-Helloty, "Enhancing facility maintenance using knowledge-based BIM systems," *Scientific Reports 2025 15:1*, vol. 15, no. 1, pp. 43392-, Dec. 2025, doi: 10.1038/s41598-025-30057-7.
- [15] A. Ehab, M. A. Mahdi, and A. El-Helloty, "BIM Maintenance System with IoT Integration: Enhancing Building Performance and Facility Management," *Civil Engineering Journal*, vol. 10, no. 6, pp. 1953–1973, Jun. 2024, doi: 10.28991/CEJ-2024-010-06-015.
- [16] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile Software Development: Methodologies and Trends," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 11, pp. 246–270, Jul. 2020, doi: 10.3991/ijim.v14i11.13269.
- [17] M. D. Beckman, M. Çetinkaya-Rundel, N. J. Horton, C. W. Rundel, A. J. Sullivan, and M. Tackett, "Implementing Version Control With Git and GitHub as a Learning Objective in Statistics and Data Science Courses," *Journal of Statistics and Data Science Education*, vol. 29, no. S1, pp. S132–S144, 2021, doi: 10.1080/10691898.2020.1848485.
- [18] J. D. Blischak, E. R. Davenport, and G. Wilson, "A Quick Introduction to Version Control with Git and GitHub," *PLoS Comput. Biol.*, vol. 12, no. 1, p. e1004668, 2016, doi: 10.1371/journal.pcbi.1004668.
- [19] W. T. Lee and C. H. Chen, "Agile Software Development and Reuse Approach with Scrum and Software Product Line Engineering," *Electronics 2023, Vol. 12, Page 3291*, vol. 12, no. 15, p. 3291, Jul. 2023, doi: 10.3390/electronics12153291.
- [20] H. K. Flora and Dr. S. V. Chande, "A Review and Analysis on Mobile Application Development Processes using Agile Methodologies," *International Journal of Research in Computer Scienc*, vol. 3, no. 4, pp. 8–18, Jul. 2013, doi: 10.7815/ijorcs.34.2013.068.
- [21] M. E. Khan and F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 6, Jul. 2012, doi: 10.14569/ijacsa.2012.030603.