

**PERANCANGAN APLIKASI MOBILE PENDUKUNG MANAJEMEN
ASET BERBASIS FLUTTER**

PRAKTIK KERJA LAPANGAN



**UNIVERSITAS
Ma CHUNG**

**BERNARDUS REYNALDI ANANDA PRIASMARA
NIM : 312110001**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG
MALANG
2025**

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga Laporan Praktik Kerja Lapangan berjudul "*Perancangan Aplikasi Mobile Pendukung Manajemen Aset Berbasis Flutter*" dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Teknologi dan Desain, Universitas Ma Chung.

Dalam penyusunan laporan ini, penulis mendapat dukungan dan bantuan dari berbagai pihak. Ucapan terima kasih disampaikan kepada:

1. Bapak Prof. Dr. Eng. Romy Budhi Widodo, selaku Dekan Fakultas Sains dan Teknologi Universitas Ma Chung. Penulis menyampaikan terima kasih atas dukungan dan kepemimpinan beliau yang menginspirasi serta menciptakan lingkungan akademik yang kondusif bagi pengembangan ilmu pengetahuan.
2. Bapak Hendry Setiawan, ST., M.Kom., selaku Kepala Program Studi Teknik Informatika Universitas Ma Chung. Terima kasih atas arahannya selama perkuliahan dan atas bimbingannya yang sangat membantu dalam proses akademik penulis.
3. Bapak Paulus Lucky Tirma I., S.Kom., MT., selaku dosen pembimbing Praktik Kerja Lapangan yang telah memberikan arahan, saran, dan bimbingan selama proses penggeraan Laporan Praktik Kerja Lapangan ini. Terima kasih atas kesabaran dan waktu yang telah diberikan dalam membimbing penulis hingga laporan ini dapat terselesaikan dengan baik.
4. Bapak Yudhi Kurniawan, S.Kom., M.MT., selaku dosen Kepala Unit Sistem Informasi dan Pusat Data (SIPUSDA), atas bimbingan teknis, masukan, dan dukungan data yang sangat membantu dalam Praktik Kerja Lapangan ini.
5. Ibu dan Kakak penulis yang selalu memberikan kasih sayang, doa, dan dukungan moral maupun material secara tanpa henti sepanjang proses studi hingga selesaiya laporan Praktik Kerja Lapangan ini.
6. Seluruh dosen dan staf Universitas Ma Chung yang telah membekali penulis dengan ilmu dan pengalaman berharga.
7. Rekan-rekan Teknik Informatika angkatan 2021 serta sahabat dekat yang telah

memberikan semangat, motivasi, dan kebersamaan selama masa studi.

Penulis menyadari bahwa laporan Praktik Kerja Lapangan ini masih memiliki kekurangan. Oleh karena itu, penulis sangat terbuka terhadap kritik dan saran yang membangun demi perbaikan di masa mendatang. Semoga karya ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya dalam bidang teknologi informasi dan pengelolaan aset berbasis AI.

Malang, 1 Juli 2025



Bernardus Reynaldi Ananda Priasmara
312110001

**LEMBAR PENGESAHAN
LAPORAN PRAKTIK KERJA LAPANGAN**

**PERANCANGAN APLIKASI MOBILE PENDUKUNG MANAJEMEN
ASET BERBASIS FLUTTER**

Oleh:

**BERNARDUS REYNALDI ANANDA PRIASMARA
NIM. 312110001**

dari:

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG**

Telah dinyatakan lulus dalam melaksanakan laporan Praktik Kerja Lapangan sebagai syarat kelulusan dan berhak mendapatkan gelar Sarjana S.Kom.

Dosen Pembimbing,

Dekan Fakultas Sains dan Teknologi



Paulus Lucky T. Irawan, S.Kom., MT. **Prof. Dr. Eng. Romy Budhi Widodo**
NIP. 20100005 **NIP. 20070035**

DAFTAR ISI

LEMBAR PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah	3
1.3 Batasan Masalah.....	4
1.4 Rumusan Masalah	4
1.5 Tujuan	5
1.6 Manfaat	5
BAB II GAMBARAN UMUM PERUSAHAAN	6
2.1 Program Studi Teknik Informatika Universitas Ma Chung	6
2.2 Sistem Pusat Data dan Informasi Universitas Ma Chung	7
BAB III TINJAUAN PUSTAKA	12
3.1 Manajemen Aset dan Proses Bisnis	12
3.2 Android Operating System.....	13
3.3 Android Studio	14
3.4 Flutter	14
3.4.1 Arsitektur Flutter.....	16
3.4.2 Plugin dan Library Tambahan	17
3.5 Dart.....	18
3.6 Proses Bisnis	18

3.7 Evaluasi Kinerja	27
BAB IV HASIL DAN PEMBAHASAN	30
4.1 Alur Penelitian.....	30
4.2 Studi Literatur	30
4.3 Analisis Kebutuhan	32
4.3.1 Analisis Kebutuhan Pengguna.....	32
4.3.2 Kebutuhan Peneliti	33
4.4 Rancangan UI.....	34
4.5 Implementasi Aplikasi.....	42
4.5.1 Alur Kerja Deteksi dan Integrasi API	44
4.6 Evaluasi Kinerja	45
4.6.1 Pengujian Halaman Utama.....	46
4.6.2 Pengujian Halaman Riwayat	53
4.6.3 Pengujian Halaman Informasi	57
4.6.4 Pengujian Error	58
4.7 Evaluasi Black Box Testing	59
BAB V	63
KESIMPULAN DAN SARAN.....	63
5.1 Kesimpulan.....	63
5.2 Saran.....	64
DAFTAR PUSTAKA	66

DAFTAR GAMBAR

Gambar 3. 1 Arsitektur Flutter	17
Gambar 3. 2 Layar Login Aplikasi MAIS.....	20
Gambar 3. 3 Layar Dashboard Aplikasi MAIS	21
Gambar 3. 4 Layar Scan QR Code Aplikasi MAIS.....	22
Gambar 3. 5 Layar Detail Aset Aplikasi MAIS	23
Gambar 3. 6 Formulir Monitoring Aset Aplikasi MAIS	24
Gambar 3. 7 Layar List Data Monitoring Aplikasi MAIS	26
Gambar 3. 8 Black Box Testing	27
Gambar 4. 1 Home Screen	35
Gambar 4. 2 Fungsi Camera.....	36
Gambar 4. 3 Fungsi Gallery	37
Gambar 4. 4 Konfigurasi API.....	38
Gambar 4. 5 Tutorial Screen	39
Gambar 4. 6 Tab Results	40
Gambar 4. 7 Tab Information	41
Gambar 4. 8 Wireframe Alur Aplikasi	43

Gambar 4. 9 Splash Screen Prototype 46

Gambar 4. 10 Onboarding Screen 47

DAFTAR TABEL

Tabel 3. 1 Studi Pustaka Penelitian Terdahulu.....	31
Tabel 4. 1 Hasil Pengujian Black-Box	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Aset kampus mencakup berbagai sumber daya milik institusi pendidikan tinggi, antara lain bangunan perkuliahan, laboratorium, peralatan pembelajaran, kendaraan operasional, dan infrastruktur teknologi. Semua aset ini memiliki peranan penting dalam menunjang kegiatan pendidikan, penelitian, dan operasional kampus. Oleh karena itu, pengelolaan aset kampus yang efektif sangat diperlukan untuk memastikan keberlangsungan operasional dan penggunaan sumber daya secara optimal. Manajemen aset yang baik akan membantu institusi menjaga kondisi aset, merencanakan perawatan atau penggantian, serta mengoptimalkan nilai umur aset tersebut.

Selama ini, pengelolaan aset di banyak institusi masih mengandalkan sistem lama yang bersifat manual. Pencatatan data aset umumnya dilakukan melalui tabel spreadsheet (misalnya Microsoft Excel atau Google Sheets) atau bahkan dokumen fisik. Pendekatan tradisional seperti ini memiliki berbagai keterbatasan, antara lain data yang tercatat dapat tidak akurat, dokumen rawan hilang, dan sulit menelusuri informasi aset secara cepat. Proses pemantauan kondisi aset pun menjadi tidak transparan dan tidak efisien karena minimnya integrasi dan otomasi. Kelemahan sistem manual yang rentan terhadap kesalahan dan kehilangan data tersebut menunjukkan urgensi untuk beralih ke solusi digital. Pencatatan aset kampus secara konvensional sangat rentan terhadap kehilangan kontrol atas data aset, sehingga dibutuhkan digitalisasi sistem manajemen aset untuk meningkatkan akurasi dan keamanan informasi aset.

Seiring perkembangan teknologi informasi, transformasi digital dalam manajemen aset kampus dipandang sebagai solusi strategis untuk mengatasi kendala sistem lama. Digitalisasi manajemen aset bukan lagi sekadar tren, melainkan kebutuhan strategis untuk meningkatkan efisiensi kerja, mempercepat layanan pencatatan, dan menekan biaya operasional. Upaya digitalisasi ini telah mulai diinisiasi di berbagai institusi; misalnya, Riyanto (2019) mengembangkan sistem informasi manajemen aset berbasis web untuk Universitas Pamulang yang

memudahkan pencatatan jumlah aset, pengelompokan, dan pelaporan aset secara terstruktur, dengan hasil implementasi menunjukkan peningkatan efisiensi dalam proses inventarisasi kampus. Kholis dan Huda (2019) juga membangun aplikasi manajemen aset multiplatform (web dan Android) di Universitas Yudharta dan mengevaluasinya menggunakan uji kebergunaan; hasilnya aplikasi tersebut memperoleh skor kelayakan yang tinggi, menandakan dapat diterima dengan baik oleh pengguna kampus. Studi-studi tersebut menunjukkan tren positif transformasi digital dalam pengelolaan aset pendidikan. Melalui digitalisasi, pengelolaan aset dapat dilakukan secara terpusat, real-time, dan terdokumentasi dengan baik, sehingga Bagian Aset dapat memantau kondisi aset secara transparan dan akuntabel. Penelitian terbaru pun melaporkan bahwa penerapan *Asset Management System* berbasis digital mampu meningkatkan efisiensi dan akurasi pengelolaan aset secara signifikan. implementasi SIM Aset berbasis web di instansi pemerintah daerah berhasil mengoptimalkan manajemen aset, dibuktikan dengan proses yang lebih cepat (real-time) dan pengurangan risiko kelalaian pencatatan aset. Temuan ini menggarisbawahi bahwa digitalisasi sistem aset berperan penting dalam meningkatkan efisiensi operasional dan akuntabilitas pengelolaan aset.

Berdasarkan kebutuhan di atas, Unit Sistem Informasi dan Pusat Data (SIPUSDA) Universitas Ma Chung berinisiatif untuk membangun sebuah aplikasi mobile yang dapat mendukung proses manajemen aset kampus secara lebih efisien. Aplikasi mobile dipilih agar petugas Bagian Aset dapat melakukan pendataan dan pendokumentasian kondisi aset langsung di lapangan menggunakan perangkat bergerak. Dengan adanya aplikasi ini, diharapkan pencatatan data aset (termasuk kondisi fisik dan kerusakan) dapat dilakukan lebih cepat, akurat, dan terpadu dalam satu sistem. Pengembangan aplikasi tersebut dilaksanakan oleh mahasiswa selama kegiatan Praktik Kerja Lapangan (PKL) di unit SIPUSDA, sebagai bagian dari kontribusi inovasi digital di lingkungan Universitas Ma Chung.

Dari sisi teknis, teknologi Flutter dipilih sebagai kerangka kerja (*framework*) untuk membangun aplikasi mobile ini. Flutter merupakan framework open-source yang dikembangkan oleh Google, yang memungkinkan pengembangan aplikasi *cross-platform* (Android dan iOS) dengan satu basis kode (*single codebase*). Dengan Flutter, tim pengembang cukup menuliskan satu kali kode program untuk

kemudian di-*compile* menjadi aplikasi *native* di berbagai platform, tanpa perlu membuat dua aplikasi terpisah untuk Android dan iOS. Pendekatan satu basis kode lintas platform ini dapat secara signifikan mengurangi waktu dan usaha pengembangan dibandingkan pendekatan pengembangan terpisah untuk setiap platform. Selain unggul dalam kemampuan lintas platform, Flutter juga menawarkan performa tinggi dan tampilan antarmuka pengguna yang konsisten serta menarik di berbagai ukuran layar perangkat. Beberapa studi menunjukkan bahwa performa aplikasi Flutter tergolong sangat baik dan mendekati aplikasi native, serta mengungguli *framework* lintas platform lainnya dalam berbagai pengujian kinerja. Hal ini disebabkan oleh arsitektur Flutter yang tidak memerlukan *bridge* ke kode native (seperti yang diperlukan pada React Native), melainkan langsung mengompilasi kode Dart ke ARM native, sehingga animasi dan respon aplikasi berjalan mulus. Flutter memiliki fitur Hot Reload yang memudahkan pengembang melihat hasil perubahan kode secara instan, sehingga proses *development* dapat berlangsung lebih cepat dan iteratif. Penggunaan Flutter diharapkan dapat mempercepat waktu pengembangan aplikasi dan memastikan aplikasi yang dihasilkan memiliki UI/UX yang responsif serta pengalaman pengguna mendekati aplikasi native. Dengan pertimbangan-pertimbangan tersebut, Flutter dipilih sebagai solusi teknis yang tepat untuk merealisasikan aplikasi mobile pendukung manajemen aset di Universitas Ma Chung.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, terdapat beberapa permasalahan utama dalam sistem manajemen aset kampus saat ini. Pertama, proses pendataan masih dilakukan secara manual, baik melalui catatan tertulis maupun file Excel. Metode pencatatan ini sangat rentan terhadap kesalahan input, duplikasi data, serta inkonsistensi informasi yang berujung pada ketidakakuratan data. Selain itu, dokumen pendukung yang berbentuk fisik maupun digital terpisah berpotensi hilang atau tidak terdokumentasi dengan baik. Proses pengolahan data aset secara manual juga membutuhkan waktu yang lebih lama, sehingga menurunkan efisiensi kerja petugas yang bertanggung jawab atas pendataan aset.

Kedua, belum tersedia mekanisme digital yang terstruktur untuk pencatatan kondisi kerusakan atau perbaikan aset. Dalam sistem yang lama, laporan kerusakan

umumnya disampaikan secara ad-hoc, misalnya melalui pesan singkat atau komunikasi lisan, tanpa adanya sistem dokumentasi yang tersimpan dalam basis data terpusat. Hal ini menyebabkan riwayat kerusakan atau tindakan perbaikan terhadap aset sulit untuk ditelusuri, informasi sering kali tersebar di berbagai tempat, dan pengambilan keputusan terkait penggantian atau perawatan aset menjadi tidak akurat. Selain itu, ketiadaan rekaman digital yang sistematis juga menyulitkan proses pelaporan kepada pimpinan institusi, karena membutuhkan usaha tambahan dalam menyusun dan menyatukan data yang relevan dari berbagai sumber.

1.3 Batasan Masalah

Adapun batasan masalah dalam penggerjaan proyek ini adalah sebagai berikut:

- a. Sistem hanya difokuskan untuk mendeteksi kerusakan pada aset kursi dan di lingkungan Universitas Ma Chung.
- b. Aplikasi dikembangkan menggunakan framework Flutter dan hanya ditujukan untuk platform Android selama masa PKL.
- c. Fitur yang dikembangkan meliputi pencatatan data aset, dokumentasi kerusakan aset, serta tampilan riwayat data. Fungsi deteksi otomatis berbasis machine learning tidak dibahas dalam laporan ini.
- d. Proyek tidak mencakup pengembangan atau integrasi sistem backend kompleks. Data disimpan secara lokal atau menggunakan metode sederhana yang mendukung keperluan uji coba awal.
- e. Dataset citra aset yang digunakan bersumber dari Universitas Ma Chung dan tidak mencakup aset dari institusi lain.

1.4 Rumusan Masalah

Berdasarkan identifikasi masalah yang telah disampaikan, maka berikut adalah rumusan masalah dari proyek ini.

- a. Bagaimana merancang aplikasi mobile yang dapat digunakan oleh petugas Bagian Aset untuk mendata dan mencatat informasi aset secara praktis di lapangan?
- b. Bagaimana menyusun antarmuka pengguna (UI) aplikasi yang sederhana, intuitif, dan mudah dipahami oleh pengguna non-teknis?

- c. Bagaimana mengevaluasi fungsionalitas aplikasi agar dapat dipastikan sesuai dengan kebutuhan pengguna (Bagian Aset) dan dapat digunakan dalam lingkungan operasional kampus?

1.5 Tujuan

Tujuan dari penggerjaan proyek ini adalah sebagai berikut.

- a. Menghasilkan *prototype* aplikasi mobile yang memungkinkan petugas mencatat data aset secara langsung di lapangan dengan perangkat Android.
- b. Merancang antarmuka pengguna (UI) yang sederhana, fungsional, dan ramah pengguna untuk mendukung operasional staf non-teknis.
- c. Melakukan pengujian dan evaluasi fungsionalitas aplikasi secara langsung untuk memastikan kesesuaian aplikasi dengan kebutuhan pengguna akhir di lingkungan Universitas Ma Chung.

1.6 Manfaat

Manfaat dari penggerjaan proyek ini adalah sebagai berikut.

- a. Bagi penulis, Meningkatkan pemahaman dan keterampilan dalam pengembangan aplikasi mobile berbasis Flutter.
- b. Bagi Universitas, Khususnya bagi Program Studi Teknik Informatika dan SIPUSDA Universitas Ma Chung, hasil proyek ini diharapkan dapat memperoleh solusi awal berupa aplikasi mobile yang dapat dimanfaatkan sebagai dasar digitalisasi manajemen aset kampus.
- c. Bagi Praktisi, Menambah referensi penerapan Flutter dalam proyek perangkat lunak praktis di bidang sistem informasi manajemen aset.

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 Program Studi Teknik Informatika Universitas Ma Chung

Universitas Ma Chung adalah perguruan tinggi swasta Indonesia yang sudah berdiri dan diresmikan sejak 7 Juli 2007. Universitas ini berada dibawah naungan Yayasan Harapan Bangsa Sejahtera dan berlokasi di Villa Puncak Tidar N-01, Karangwidoro, Dau, Malang, Indonesia.

Visi Universitas Ma Chung adalah untuk memuliakan Tuhan melalui penerapan nilai-nilai moral, penguasaan ilmu pengetahuan, serta kontribusi nyata dalam kehidupan akademik dan sosial. Untuk mewujudkan visi tersebut, Universitas Ma Chung mengembangkan sejumlah misi, antara lain:

- a. Menyelenggarakan Tri Dharma Perguruan Tinggi secara terfokus dan berkualitas, yang mencakup pendidikan tinggi, penelitian ilmiah, dan pengabdian kepada masyarakat yang relevan dengan kebutuhan zaman.
- b. Membentuk dan mengembangkan generasi motivator dan pemimpin yang berkarakter mulia, berjiwa kewirausahaan, serta memiliki kepribadian yang unggul, rendah hati, dan berorientasi pada pelayanan.
- c. Mendorong dan mengembangkan pola pikir kritis dan kreatif yang dilandasi oleh kepekaan hati nurani dan prinsip realistik.
- d. Menghasilkan lulusan yang kompeten dan siap bersaing di tingkat nasional maupun internasional.
- e. Berperan aktif dalam pengembangan peradaban global melalui lulusan yang toleran, cinta damai, dan produktif dalam menghasilkan karya yang memanusiakan sesama.
- f. Melaksanakan tata kelola institusi pendidikan tinggi berdasarkan prinsip efisiensi dan akuntabilitas.

Hingga tahun 2024, Universitas Ma Chung mengelola empat fakultas dan sebelas program studi. Empat fakultas tersebut adalah Fakultas Ekonomi dan Bisnis (mencakup program S1 Manajemen, S1 Akuntansi, dan S2 Magister Manajemen Inovasi), Fakultas Bahasa (S1 Sastra Inggris dan S1 Pendidikan Bahasa Mandarin), Fakultas Teknologi dan Desain (S1 Teknik Informatika, S1 Sistem Informasi, S1

Teknik Industri, dan S1 Desain Komunikasi Visual), serta Fakultas Ilmu Kesehatan, yang menaungi program S1 Farmasi dan D3 Optometri serta program profesi Apoteker

Program studi Teknik Informatika memiliki dua jalur peminatan, yaitu peminatan sistem cerdas yang berfokus ke bidang kecerdasan buatan dan sistem komputer yang berfokus ke bidang jaringan komputer. Pada proyek Praktik Kerja Lapangan kelompok riset Precision Agriculture termasuk dalam peminatan sistem cerdas. Di sini penulis menjalankan Praktik Kerja Lapangan sebagai bagian dari kelompok riset tersebut, yang relevan dengan pengembangan aplikasi mobile untuk pendataan aset kampus.

2.2 Sistem Pusat Data dan Informasi Universitas Ma Chung

Unit Sistem Informasi dan Pusat Data (SIPUSDA) Universitas Ma Chung merupakan salah satu unit strategis yang berperan penting dalam pengembangan, pengelolaan, dan pemeliharaan infrastruktur teknologi informasi di lingkungan kampus. Unit ini berada di bawah koordinasi langsung institusi dan bertugas mendukung penyelenggaraan aktivitas akademik maupun non-akademik yang membutuhkan layanan teknologi informasi, seperti pengembangan sistem informasi internal, manajemen jaringan kampus, layanan multimedia, serta administrasi perangkat keras dan lunak.

Secara struktural, SIPUSDA dipimpin oleh Bapak Yudhi Kurniawan, S.Kom., M.MT. selaku Kepala Unit (*Head of IT*), dan dibantu oleh lima staf profesional yang bekerja secara penuh waktu (*full-time*) setiap hari kerja, yaitu Senin hingga Jumat pukul 08.00–17.00 WIB. Total terdapat enam orang yang tergabung dalam unit ini, dengan pembagian tugas sebagai berikut:

a. *Head of IT*

Bapak Yudhi Kurniawan, S.Kom., M.MT. bertanggung jawab memformulasikan arah strategis teknologi informasi universitas dan menerjemahkannya ke dalam rencana kerja tahunan yang terukur melalui *Key Performance Indicator/Service Level Agreement*. Ruang lingkupnya mencakup tata kelola TI (*governance*), manajemen risiko dan kepatuhan (*Governance, Risk, Compliance*), serta penganggaran—mulai dari perencanaan, alokasi, hingga evaluasi pemanfaatan anggaran TI. Selain

memimpin portofolio proyek dan mengoordinasikan lintas tim (*developer*, infrastruktur, service desk), *Head of IT* menyusun dan menegakkan kebijakan keamanan informasi, SOP operasional, serta kerangka manajemen perubahan (*change management*) termasuk persetujuan rilis melalui forum/rapat pengendali perubahan. *Head of IT* juga menjadi penghubung antara unit TI dan pimpinan universitas untuk memastikan inisiatif seperti integrasi sistem kampus terpadu dan *Single Sign-On* berjalan konsisten, melakukan evaluasi tren teknologi (cloud, otomasi, AI), negosiasi vendor/mitra, serta mengawal audit berkala, uji rencana keberlanjutan layanan (*Business Continuity Plan / Disaster Recovery Plan*), dan peningkatan kapasitas SDM.

b. *IT Infrastructure and Network*

Bapak Dwi Endra Krisna, S.Kom. Bertugas merancang, mengimplementasikan, dan memelihara arsitektur jaringan kampus (routing, switching, segmentasi/VLAN, firewall, dan manajemen Wi-Fi) serta layanan pendukungnya seperti DNS, DHCP, NTP, dan otentikasi jaringan. Tim ini melakukan pemantauan ketersediaan dan kinerja server/jaringan, penanganan insiden dan problem root-cause analysis, serta penguatan postur keamanan melalui patch/firmware update, hardening, dan kontrol akses. Di sisi komputasi, tim mengelola server fisik/virtual, penyimpanan (NAS/SAN), backup–restore terjadwal, serta orkestrasi kapasitas agar layanan tetap andal saat beban meningkat. Dokumentasi topologi dan konfigurasi dijaga mutakhir untuk mempercepat troubleshooting, sementara uji coba pemulihan bencana (disaster recovery drill) dilakukan berkala untuk memastikan RTO/RPO tercapai. Fungsi ini juga menilai dan merekomendasikan adopsi teknologi baru yang relevan (misalnya konsolidasi ke lingkungan hybrid/cloud) untuk meningkatkan efisiensi operasional.

c. *Information System Developer*

Bapak Amal Dermawan Udjir, S.Kom. berfokus pada otomasi proses bisnis kampus. Aktivitas dimulai dari analisis kebutuhan (wawancara, observasi, pemodelan proses), perancangan arsitektur aplikasi dan basis data (use case, ERD, flow/UI flow), hingga implementasi aplikasi web. Pengembang menyiapkan dan mendokumentasikan API (RESTful/HTTP)

untuk integrasi antarsistem internal/eksternal, menerapkan praktik pengembangan yang baik (version control, code review, branch strategy), serta menyusun pengujian berjenjang (unit, integrasi, sistem) untuk menjamin kualitas. Pengelolaan *error logging*, audit trail, dan keamanan data (validasi input, kontrol akses, sanitasi) menjadi bagian integral dari siklus pengembangan. Setelah rilis, IS *Developer* menangani perbaikan bug, menyusun changelog, dan melakukan continuous improvement berdasarkan umpan balik pengguna dan data penggunaan sistem, termasuk interoperabilitas dengan sistem kampus lain (akademik, kepegawaian, atau inventaris).

d. Web Full Stack Developer

Bapak Leonardo Steven Wijaya, S.Kom. Bertanggung jawab mengembangkan tampilan antarmuka (*front-end*) yang responsif dan aksesibel, logika bisnis (*back-end*), hingga pengelolaan basis data. Di lapisan frontend, pengembang memastikan konsistensi UI/UX, kompatibilitas lintas peramban, dan kepatuhan prinsip aksesibilitas. Di backend, pengembang merancang API yang efisien dan aman, menerapkan *authentication/authorization*, serta *data validation/sanitization* yang kuat. Optimasi performa dilakukan melalui caching, pagination, dan pemanfaatan CDN/optimasi asset untuk mempercepat *page load*. Siklus rilis menerapkan CI/CD, pengujian otomatis (unit, integrasi, *end-to-end*), dan observabilitas (*logging, metrics, error monitoring*). Untuk reliabilitas, pengembang menyiapkan strategi *rollback*, dokumentasi rilis, dan mekanisme *feature flag* saat diperlukan. Aspek keamanan mengikuti praktik terbaik (mitigasi OWASP Top 10) agar sistem tetap tahan terhadap serangan.

e. Mobile Full Stack Developer

Bapak M. Akif Balya Alim, S.Kom. Memfokuskan *Mobile Developer* merancang dan membangun aplikasi native atau lintas-platform dengan arsitektur yang terstruktur (mis. BLoC/MVVM) agar mudah dirawat dan diuji. Di sisi UI, pengembang memastikan antarmuka intuitif, responsif, dan mengikuti pedoman platform (Material Design/HIG). Integrasi dengan backend dilakukan melalui API yang aman (*token-based auth*), dengan

pengelolaan sesi dan penyimpanan kredensial yang aman (*secure storage*). Untuk keandalan, pengembang menerapkan *offline-first* dan sinkronisasi data, meminimalkan konsumsi memori/baterai, serta memantau crash dan kinerja melalui alat analitik/monitoring. Pengujian dilakukan berlapis (unit, widget/integration, *instrumentation*) sebelum rilis internal. Proses rilis mengelola *app signing*, varian *build* (dev/staging/prod), dan *feature toggles*. Pemeliharaan meliputi kompatibilitas OS/perangkat terbaru, perbaikan bug/performa, serta penanganan umpan balik pengguna untuk iterasi berkelanjutan.

f. IT Service Desk

Bapak Farhan Adriansyah Ekadana, S.Kom. garda terdepan dukungan TI yang menjalankan proses *incident management* dan *service request management* secara terukur. Tugasnya meliputi penerimaan tiket, klasifikasi dan penentuan prioritas, penyelesaian langsung (*first call resolution*) atau eskalasi terstruktur ke tim infrastruktur/developer sesuai matriks eskalasi dan SLA. Service Desk menyusun dan memperbarui dokumentasi teknis, panduan troubleshooting, serta *knowledge base* untuk mendorong *self-service* pengguna dan konsistensi solusi. Aktivitas pemantauan tren insiden, analisis berulang (problem management), dan pelaporan periodik (*SLA attainment*, CSAT) dilakukan untuk memberi masukan perbaikan layanan. Selain dukungan harian (*onsite/remote*), Service Desk juga melakukan edukasi singkat kepada pengguna terkait praktik aman penggunaan akun, kebijakan perangkat, dan etika penggunaan sistem.

Visi SIPUSDA adalah menciptakan ekosistem teknologi informasi yang mendukung pencapaian tujuan akademik, mempermudah akses bagi pengguna, serta mendorong kolaborasi dan inovasi dari seluruh elemen kampus. Fokus kerja utama SIPUSDA pada tahun 2025 adalah integrasi sistem informasi kampus ke dalam satu platform terpadu berbasis cloud, yaitu Ma Chung Integrated Information System (MACIS), serta penerapan sistem otentikasi tunggal (*Single Sign-On*) untuk seluruh layanan digital internal kampus.

Selama pelaksanaan Praktik Kerja Lapangan, penulis ditempatkan dalam tim

Information System Development dan bertanggung jawab dalam perancangan serta pengembangan aplikasi mobile pendukung manajemen aset berbasis Flutter. Kegiatan kerja dilaksanakan secara Work From Office (WFO) dengan presensi harian melalui sistem online, dan laporan perkembangan proyek disampaikan setiap hari Rabu pukul 13.00 WIB kepada pembimbing dari pihak SIPUSDA. Proyek ini menjadi bagian dari kontribusi terhadap pengembangan sistem digital kampus dan selaras dengan arah strategis transformasi teknologi Universitas Ma Chung.

BAB III

TINJAUAN PUSTAKA

3.1 Manajemen Aset dan Proses Bisnis

Manajemen aset merupakan suatu proses sistematis yang mencakup seluruh tahapan siklus hidup aset, mulai dari pengadaan, pendataan, penggunaan, pemeliharaan, hingga penghapusan. Aset dalam konteks institusi pendidikan dapat berupa sarana dan prasarana, peralatan, maupun perangkat teknologi informasi yang digunakan dalam menunjang kegiatan operasional dan akademik. Wahyuni dan Khoirudin (2020) menjelaskan bahwa manajemen aset bertujuan untuk memastikan ketersediaan, keandalan, serta optimalisasi penggunaan aset agar mendukung pencapaian tujuan organisasi secara efisien dan efektif.

Dalam implementasinya, proses bisnis pendataan dan pemantauan kerusakan aset di lingkungan kampus sebelumnya dilakukan secara manual dan sederhana. Bagian Aset sebagai pihak yang bertanggung jawab terhadap pengelolaan inventaris kampus telah menggunakan aplikasi internal sederhana yang hanya memungkinkan pengunggahan gambar kerusakan aset. Namun, sistem tersebut tidak memiliki kemampuan untuk mengklasifikasi atau memverifikasi tingkat kerusakan secara otomatis. Verifikasi masih dilakukan secara manual oleh petugas melalui pengecekan langsung atau interpretasi dari gambar yang dikirimkan. Akibatnya, proses ini memakan waktu, bergantung pada tenaga manusia, dan rentan terhadap kesalahan atau keterlambatan tindak lanjut.

Kondisi ini mendorong perlunya pengembangan aplikasi yang tidak hanya mampu mendokumentasikan aset dan kerusakannya, tetapi juga menyajikan informasi secara lebih sistematis, terstruktur, dan efisien. Sistem baru dirancang agar petugas lapangan dapat secara langsung mengambil gambar kondisi aset melalui perangkat mobile, melengkapi data deskriptif, dan menyimpannya dalam sistem digital yang terintegrasi. Proses bisnis baru ini mengurangi ketergantungan pada proses manual dan memungkinkan penyusunan laporan serta pelacakan riwayat kerusakan dengan lebih baik.

Seiring berkembangnya teknologi informasi, digitalisasi dalam manajemen aset menjadi suatu kebutuhan. Digitalisasi ini mengubah proses manual yang

berisiko terhadap kesalahan pencatatan, duplikasi data, dan kehilangan informasi, menjadi sistem yang terdokumentasi dengan baik, akurat, dan mudah ditelusuri (Riyanto, 2019). Dalam konteks kampus, penerapan sistem informasi manajemen aset digital dinilai mampu meningkatkan transparansi dan akuntabilitas, serta mempermudah monitoring dan pelaporan kondisi aset secara real-time (Ahmad & Maulana, 2020). Manajemen aset merupakan suatu proses sistematis yang mencakup seluruh tahapan siklus hidup aset, mulai dari pengadaan, pendataan, penggunaan, pemeliharaan, hingga penghapusan. Aset dalam konteks institusi pendidikan dapat berupa sarana dan prasarana, peralatan, maupun perangkat teknologi informasi yang digunakan dalam menunjang kegiatan operasional dan akademik. Wahyuni dan Khoirudin (2020) menjelaskan bahwa manajemen aset bertujuan untuk memastikan ketersediaan, keandalan, serta optimalisasi penggunaan aset agar mendukung pencapaian tujuan organisasi secara efisien dan efektif.

Seiring berkembangnya teknologi informasi, digitalisasi dalam manajemen aset menjadi suatu kebutuhan. Digitalisasi ini mengubah proses manual yang berisiko terhadap kesalahan pencatatan, duplikasi data, dan kehilangan informasi, menjadi sistem yang terdokumentasi dengan baik, akurat, dan mudah ditelusuri (Riyanto, 2019). Dalam konteks kampus, penerapan sistem informasi manajemen aset digital dinilai mampu meningkatkan transparansi dan akuntabilitas, serta mempermudah monitoring dan pelaporan kondisi aset secara real-time (Ahmad & Maulana, 2020).

3.2 Android Operating System

Sistem operasi Android merupakan OS berbasis kernel Linux yang dirancang untuk perangkat bergerak seperti smartphone dan tablet. Android dikembangkan oleh Android Inc. dan kemudian diakuisisi oleh Google. OS ini bersifat open source melalui proyek Android Open Source Project (AOSP), sehingga banyak digunakan dan dimodifikasi oleh produsen perangkat di seluruh dunia. Android terdiri atas beberapa lapisan arsitektur, yakni kernel Linux, library dan Android runtime (ART), application framework, dan application layer. Keunggulan utama Android antara lain fleksibilitas dalam pengembangan aplikasi, kompatibilitas dengan berbagai jenis perangkat keras, serta dukungan terhadap berbagai layanan Google

seperti Google Play, Maps, dan Firebase (Safaat, 2012).

Android juga dikenal dengan kemampuannya dalam mendukung pengembangan aplikasi mobile untuk kebutuhan operasional lapangan. Fitur-fitur seperti kamera, GPS, sensor gerak, dan jaringan data sangat mendukung pengembangan aplikasi yang digunakan di lingkungan kerja dinamis seperti pencatatan aset lapangan. Kombinasi antara fleksibilitas sistem, dukungan komunitas pengembang, dan infrastruktur distribusi aplikasi membuat Android menjadi platform pilihan utama dalam proyek pengembangan aplikasi mobile di lingkungan kampus maupun industri.

3.3 Android Studio

Android Studio merupakan Integrated Development Environment (IDE) resmi yang dikembangkan oleh Google untuk membangun aplikasi Android. IDE ini berbasis IntelliJ IDEA dan dilengkapi dengan berbagai fitur yang mendukung proses pengembangan aplikasi secara komprehensif, mulai dari penulisan kode, debugging, hingga proses kompilasi dan distribusi aplikasi ke Google Play Store. Android Studio menyediakan emulator, visual layout editor, integrasi dengan Firebase, serta dukungan penuh untuk Android SDK dan Android Jetpack, yang mempermudah pengembangan aplikasi Android secara modern dan efisien (Google Developers, 2023).

Dalam konteks penelitian ini, meskipun pengembangan utama aplikasi dilakukan menggunakan Flutter yang lintas platform, Android Studio tetap digunakan sebagai lingkungan pengembangan utama untuk keperluan pengujian dan debugging di perangkat Android. Android Studio kompatibel dengan plugin Flutter dan Dart, sehingga memungkinkan pengembang menjalankan simulasi aplikasi langsung di emulator Android atau perangkat fisik. Keunggulan ini menjadikan Android Studio sebagai alat pendukung penting dalam pengembangan aplikasi mobile pendekripsi kerusakan aset yang efisien dan stabil.

3.4 Flutter

Flutter adalah framework open-source yang dikembangkan oleh Google untuk membangun aplikasi mobile secara lintas platform (*cross-platform*). Berbeda dari pendekatan natif tradisional yang mengharuskan pengembangan terpisah untuk

Android (Java/Kotlin) dan iOS (Swift/Objective-C), Flutter memungkinkan penulisan satu basis kode (menggunakan bahasa Dart) yang dapat dikompilasi langsung menjadi aplikasi Android maupun iOS. Flutter menyediakan serangkaian widget UI bawaan yang kaya dan fleksibel, sehingga pengembang dapat merancang antarmuka pengguna yang elegan dengan performa mendekati native. Mesin rendering Flutter (berbasis Skia) menggambar UI langsung ke kanvas tanpa memerlukan bridge ke kode native, yang membuat animasi dan respons aplikasi menjadi halus. Salah satu alasan pemilihan Flutter dalam penelitian ini adalah efisiensi pengembangan lintas platform dan kematangan ekosistemnya. Dengan Flutter, aplikasi pendeteksi kerusakan aset dapat dikembangkan dan diuji dengan cepat pada berbagai perangkat tanpa perlu menulis ulang kode untuk tiap sistem operasi. Selain itu, Flutter dikenal mudah diintegrasikan dengan pustaka-pustaka pihak ketiga, termasuk yang mendukung fungsi machine learning. Menurut dokumentasi resmi Google (2022), Flutter memiliki kompatibilitas yang baik untuk integrasi model pembelajaran mesin.

Flutter juga dilengkapi fitur "hot reload", yang memungkinkan pengembang untuk melihat hasil perubahan kode secara langsung tanpa harus melakukan proses kompilasi ulang secara penuh. Fitur ini mempercepat proses pengembangan dan debugging, sehingga produktivitas pengembang meningkat secara signifikan (Flutter, 2022).

Pemilihan Flutter dalam proyek ini tidak hanya didasarkan pada keunggulan teknisnya, tetapi juga karena aplikasi pendukung manajemen aset sebelumnya telah dikembangkan menggunakan framework Flutter. Dengan demikian, penggunaan Flutter dalam proyek ini memungkinkan kesinambungan pengembangan dan kompatibilitas dengan struktur aplikasi yang telah ada. Hal ini juga memudahkan proses pemeliharaan, penambahan fitur, dan integrasi dengan komponen atau modul yang telah dikembangkan sebelumnya.

Dalam hal antarmuka pengguna, Flutter menyediakan pustaka widget yang kaya dan dapat dikustomisasi, baik untuk gaya Material Design (Android) maupun Cupertino (iOS). Semua elemen UI dibangun menggunakan pendekatan deklaratif, sehingga UI dapat dengan mudah dikelola dan diubah sesuai state aplikasi. Selain itu, karena Flutter menggambar UI-nya sendiri menggunakan engine rendering

Skia, hasil tampilan antar perangkat lebih konsisten.

Dari sisi performa, Flutter tidak menggunakan JavaScript bridge seperti pada framework hybrid lain (misalnya React Native), melainkan mengkompilasi kode Dart langsung ke kode native (ARM). Dengan demikian, aplikasi Flutter mampu menjalankan animasi dan transisi dengan halus serta respon yang cepat.

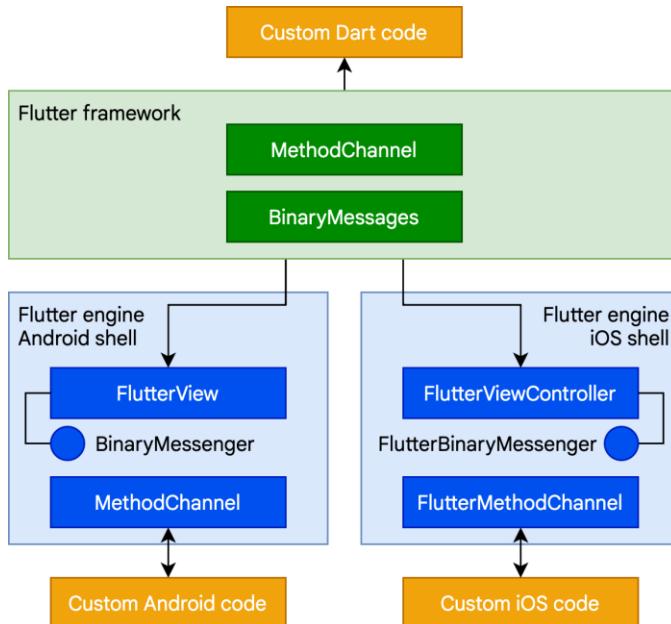
Terakhir, Flutter memiliki dukungan komunitas yang luas dan dokumentasi yang lengkap. Hal ini memberikan keuntungan dalam hal ketersediaan paket pustaka pihak ketiga dan kemudahan mencari solusi ketika menghadapi kendala teknis selama proses pengembangan. Oleh karena itu, Flutter sangat layak dipilih sebagai platform utama dalam pengembangan aplikasi mobile untuk proyek ini.

3.4.1 Arsitektur Flutter

Arsitektur Flutter terdiri atas tiga komponen utama yang saling terintegrasi, yaitu Framework, Engine, dan Embedder. Lapisan pertama adalah Framework, yang dibangun menggunakan bahasa Dart. Lapisan ini merupakan area utama bagi pengembang dalam menyusun antarmuka pengguna dan logika aplikasi. Framework Flutter menyediakan berbagai macam widget dan pustaka antarmuka pengguna (UI), serta tools untuk mengatur navigasi, layout, animasi, dan state management. Penggunaan pendekatan deklaratif berbasis widget membuat pengembangan UI menjadi lebih modular dan efisien.

Lapisan kedua adalah Engine, ditulis dalam bahasa pemrograman C++, yang bertanggung jawab atas rendering antarmuka pengguna ke layar. Engine ini menggunakan pustaka grafis Skia untuk menggambar elemen UI secara langsung ke canvas, sehingga menghasilkan performa tinggi dengan frame rate yang stabil. Selain itu, engine juga mengelola proses kompilasi just-in-time (JIT) dan ahead-of-time (AOT), serta mengatur komunikasi antara kode Dart dengan native code melalui platform channel.

Lapisan ketiga adalah Embedder, yang berfungsi sebagai jembatan antara engine Flutter dengan sistem operasi target (seperti Android atau iOS). Embedder menangani rendering ke layar perangkat, pengaturan input dari pengguna (touch, gesture), akses ke sistem file, kamera, serta integrasi lainnya dengan API native. Dengan arsitektur ini, Flutter mampu bekerja di berbagai platform dengan kinerja yang mendekati aplikasi native (Google Developers, 2023).



Gambar 3. 1 Arsitektur Flutter

3.4.2 Plugin dan Library Tambahan

Flutter juga didukung oleh ekosistem plugin dan library pihak ketiga yang sangat luas. Dalam pengembangan aplikasi pencatatan aset ini, digunakan beberapa plugin penting untuk menunjang kebutuhan fungsionalitas dan arsitektur aplikasi.

Pertama, digunakan plugin `flutter_bloc`, yaitu implementasi dari pola arsitektur Business Logic Component (BLoC). Plugin ini memfasilitasi pemisahan antara logika bisnis dan tampilan antarmuka pengguna, sehingga membuat struktur kode menjadi lebih rapi, modular, dan mudah dipelihara. Penerapan BLoC juga memungkinkan manajemen state yang efisien dan prediktif, khususnya dalam menangani interaksi pengguna yang kompleks. Kedua, digunakan plugin `sqflite` untuk mendukung penyimpanan data secara lokal di perangkat pengguna. Plugin ini mengimplementasikan SQLite, yang memungkinkan aplikasi untuk menyimpan dan mengakses data secara offline, sehingga tetap dapat berfungsi dengan baik meskipun tidak terhubung ke jaringan internet. Ketiga, plugin `image_picker` digunakan untuk memungkinkan pengguna mengambil gambar secara langsung melalui kamera maupun memilih gambar dari galeri perangkat. Fitur ini sangat penting dalam konteks pencatatan kerusakan aset, di mana dokumentasi visual menjadi bukti pendukung utama. Selanjutnya, plugin `path_provider` digunakan untuk mengakses lokasi direktori penyimpanan perangkat seperti direktori

dokumen atau cache. Hal ini diperlukan agar file gambar maupun database lokal dapat disimpan dengan aman dan terstruktur.

Pemanfaatan plugin-plugin tersebut disesuaikan dengan kebutuhan pengembangan aplikasi serta mempertimbangkan kompatibilitas dengan Flutter. Dengan mengintegrasikan plugin yang tepat, aplikasi yang dibangun menjadi lebih fungsional, responsif, dan siap digunakan dalam konteks operasional kampus. Android Studio adalah Integrated Development Environment (IDE) resmi yang disediakan oleh Google untuk pengembangan aplikasi Android. IDE ini berbasis IntelliJ IDEA dan menyediakan berbagai alat bantu seperti layout editor, code completion, debugging tools, emulator Android, serta integrasi dengan sistem build Gradle. Android Studio juga mendukung pengembangan aplikasi berbasis Flutter melalui plugin resmi, sehingga memungkinkan pengembang untuk menulis, menjalankan, dan menguji aplikasi Flutter langsung dalam satu lingkungan kerja yang terpadu. Dengan Android Studio, proses pengembangan aplikasi menjadi lebih terstruktur dan produktif karena berbagai fitur otomatisasi dan analisis kodennya (Google Developers, 2023).

3.5 Dart

Dart adalah bahasa pemrograman yang dikembangkan oleh Google dan digunakan secara utama dalam framework Flutter. Dart merupakan bahasa berorientasi objek yang dirancang untuk kebutuhan pengembangan aplikasi modern, dengan sintaks yang mirip JavaScript dan Java, namun dengan performa dan fitur yang lebih optimal untuk pengembangan antarmuka pengguna. Dart mendukung kompilasi just-in-time (JIT) untuk kecepatan pengembangan, serta ahead-of-time (AOT) untuk kecepatan eksekusi pada aplikasi yang telah selesai dibangun. Selain itu, Dart memiliki fitur seperti null safety, asynchronous programming (async-await), dan struktur data koleksi yang membuatnya ideal untuk pengembangan aplikasi yang kompleks dan responsif (Dart.dev, 2023).

3.6 Proses Bisnis

Sistem eksisting yang digunakan oleh Bagian Aset Universitas Ma Chung masih bersifat manual dan memiliki tingkat otomasi yang sangat rendah. Dalam pelaksanaannya, petugas lapangan melakukan inspeksi berkala ke ruangan-ruangan

kampus. Ketika ditemukan aset seperti kursi yang mengalami kerusakan, petugas akan mengambil gambar menggunakan ponsel, lalu mengunggahnya ke aplikasi web internal sederhana yang dibangun oleh SIPUSDA. Pada aplikasi tersebut, petugas juga diwajibkan menuliskan deskripsi kerusakan secara manual. Seluruh data berupa identitas aset, lokasi, foto, dan deskripsi kerusakan disimpan tanpa adanya proses analisis otomatis. Proses identifikasi kerusakan dan tindak lanjutnya sangat bergantung pada pengamatan dan interpretasi subjektif petugas. Setelah laporan masuk, pihak atas akan meninjau dan menentukan apakah aset perlu diperbaiki atau diganti. Sistem ini tidak memiliki standarisasi dalam penulisan deskripsi kerusakan, sangat bergantung pada ketelitian petugas, serta tidak efisien apabila jumlah aset rusak yang ditemukan meningkat.

Untuk mengatasi berbagai keterbatasan tersebut, sistem usulan yang dikembangkan dalam proyek ini dirancang menggunakan aplikasi mobile berbasis Flutter yang terintegrasi dengan model deteksi kerusakan berbasis CNN melalui API eksternal. Dalam sistem baru ini, petugas cukup mengambil gambar kursi yang dicurigai rusak melalui kamera aplikasi atau memilih gambar dari galeri, kemudian aplikasi akan secara otomatis mengirim gambar tersebut ke server untuk diproses oleh model CNN. Petugas hanya perlu memverifikasi hasil prediksi; jika sesuai, data dapat langsung disimpan. Jika tidak sesuai, tersedia opsi untuk memilih kategori kerusakan dari daftar yang telah ditentukan. Data yang disimpan bersifat terstruktur, sehingga meningkatkan konsistensi dan memungkinkan analisis historis secara efisien. Riwayat hasil deteksi tersimpan di perangkat dan dapat diakses kapan saja, bahkan tanpa koneksi internet. Dengan sistem ini, proses dokumentasi kerusakan menjadi lebih cepat, akurat, dan terstandardisasi, serta membuka peluang untuk integrasi data ke sistem pelaporan real-time milik SIPUSDA.

3.7 Aplikasi Sebelumnya

Pada tahap sebelumnya, Universitas Ma Chung telah menggunakan aplikasi Mobile Asset Information System (MAIS) sebagai sarana untuk melakukan pencatatan dan monitoring kondisi aset kampus. Aplikasi ini berfungsi sebagai pendukung kegiatan administrasi, khususnya dalam proses pencatatan manual hasil inspeksi aset yang dilakukan oleh petugas lapangan. Setiap aset diberi identitas berupa kode atau QR code yang dapat dipindai melalui perangkat mobile, kemudian data kondisi aset, catatan, dan bukti foto dimasukkan secara manual ke dalam sistem. Meskipun aplikasi ini sudah membantu mengurangi ketergantungan pada pencatatan berbasis kertas, sistem lama masih memiliki sejumlah keterbatasan, terutama dalam hal validasi data, keakuratan informasi, serta efisiensi proses monitoring. Oleh karena itu, diperlukan pengembangan aplikasi baru dengan fitur otomatisasi verifikasi agar proses pencatatan aset menjadi lebih akurat, terintegrasi, dan mampu mendukung pengambilan keputusan secara real-time.



**Mobile Asset
Information System**

Universitas Machung

MAC IS Username
0/100

>Password
0/100

Login

Gambar 3. 2 Layar Login Aplikasi MAIS

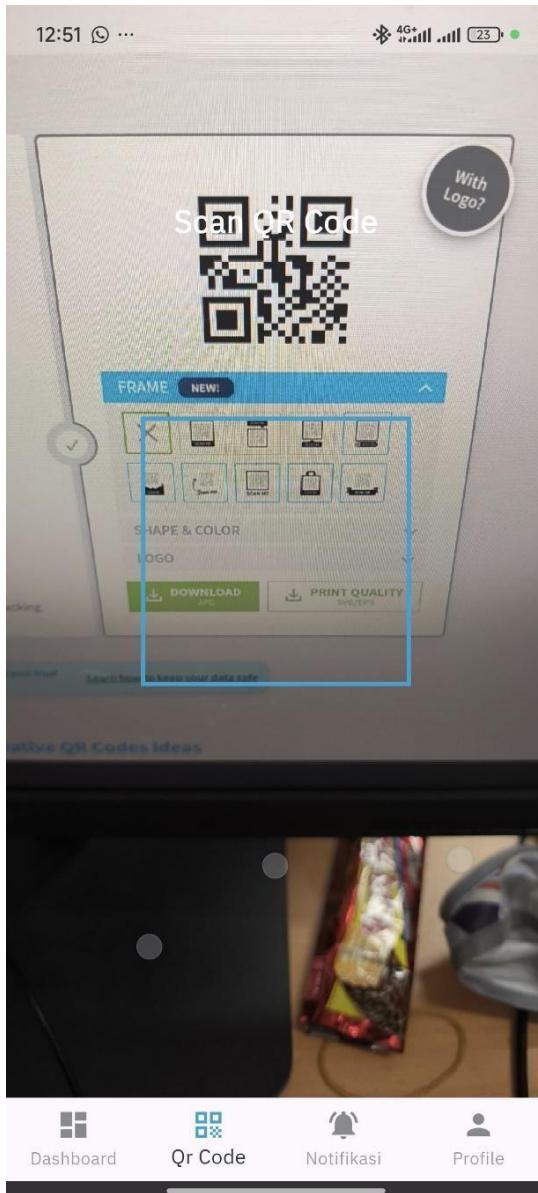
Pada tahap awal, pengguna harus melakukan login ke aplikasi Mobile Asset Information System (MAIS) Universitas Ma Chung. Layar login meminta username dan password sebagai autentifikasi pengguna sebelum mengakses fitur-fitur monitoring asset. Proses ini memastikan hanya petugas atau user yang berwenang yang dapat masuk ke sistem untuk melakukan pemantauan aset. Setelah berhasil login, pengguna diarahkan ke menu utama aplikasi untuk memilih fungsi yang diinginkan, termasuk fungsi monitoring aset.



Gambar 3. 3 Layar Dashboard Aplikasi MAIS

Pada menu dashboard, ditampilkan berbagai modul pengelolaan aset secara mobile. Terdapat ikon List Monitoring dan Jadwal Monitoring yang berkaitan dengan aktivitas inspeksi rutin aset, serta fitur lain seperti Pengaduan Aset Rusak dan Perbaikan Aset. Untuk memulai proses monitoring aset, pengguna dapat memilih menu QR Code pada bottom navigation aplikasi. Fitur ini memungkinkan petugas memindai kode QR yang terpasang pada aset, sehingga identitas aset dapat

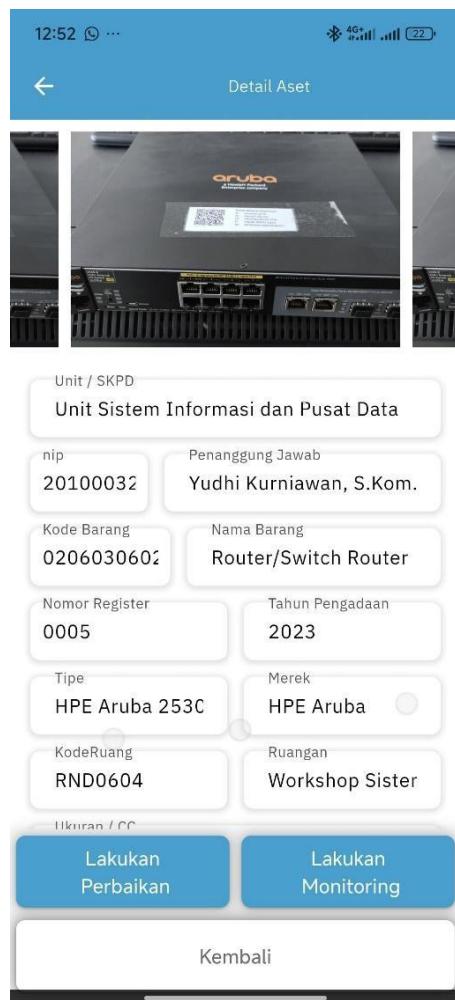
diperoleh secara instan tanpa perlu input manual yang rawan kesalahan. Hal ini sejalan dengan praktik terbaik manajemen aset dimana pemanfaatan barcode atau QR code mampu mengotomatiskan identifikasi aset dan mengurangi human error



Gambar 3. 4 Layar Scan QR Code Aplikasi MAIS

Setelah memilih mode pemindaian, aplikasi mengaktifkan kamera untuk scan QR code. Petugas mengarahkan kamera ke label QR di aset; sistem kemudian membaca kode unik tersebut dan mencocokkannya dengan database aset. Jika sesuai, aplikasi akan menampilkan informasi aset yang bersangkutan secara otomatis. Metode ini mempercepat akses data dan memastikan aset yang diinspeksi benar dan sesuai catatan, menggantikan pencatatan manual yang lambat dan rentan

salah identifikasi . Dengan scan QR code, informasi dasar aset dapat ditarik secara real-time, seperti dijelaskan Benrahman et al. (2021) bahwa pemindaian QR mampu menampilkan data aset secara instan dengan akurasi sangat tinggi (99,9%) . Pada proses manual sebelumnya (misalnya menggunakan daftar aset cetak atau spreadsheet), petugas harus mencari data aset secara manual, sehingga penggunaan QR code pada aplikasi mobile ini adalah langkah maju untuk efisiensi. Meskipun demikian, langkah ini masih memerlukan interaksi manual oleh petugas (memindai satu per satu), dan sistem belum melakukan verifikasi lain selain pengecekan kode aset.



Gambar 3. 5 Layar Detail Aset Aplikasi MAIS

Setelah QR code berhasil dipindai, aplikasi menampilkan Detail Aset. Pada layar ini tertera informasi lengkap aset, antara lain: unit pemilik (Unit/SKPD),

penanggung jawab, kode barang, nama barang, nomor register, tahun pengadaan, merek/tipe, lokasi (ruangan), hingga umur aset. Tampilan detail ini berfungsi untuk verifikasi visual oleh petugas bahwa aset yang akan dimonitor memang benar (misalnya memastikan nama dan lokasi sesuai dengan aset fisik di lapangan). Apabila aset yang dimaksud memerlukan tindakan, tersedia dua pilihan: Lakukan Monitoring untuk pencatatan kondisi rutin, atau Lakukan Perbaikan jika ditemukan kerusakan yang membutuhkan tindak lanjut. Dalam konteks monitoring berkala, petugas akan menekan tombol Lakukan Monitoring. Seluruh proses ini masih berjalan manual dalam arti petugas yang mengambil inisiatif memindai dan memutuskan tindakan; sistem belum memberikan prompt otomatis atau pengecekan selain menampilkan data aset.

The screenshot shows a mobile application interface for monitoring an asset. At the top, there's a blue header bar with the time '12:52', signal strength, and battery level. Below it, a back arrow and the word 'Monitoring' are visible. The main content area has a white background. It displays the asset details: 'Router/Switch Router', 'Workshop Sistem Info', and 'Usia: 20 bulan'. There's a placeholder 'NO IMAGE' with a camera icon. A section for 'Tanggal Monitoring' shows '12 Aug 2025'. A dropdown menu for 'Kondisi' is set to 'Baik'. Below it, a text input field contains the note 'catatan Inspeksi / Monitoring bagus'. A blue button at the bottom left says 'Upload Bukti Foto'. To its right is a blue button labeled 'Upload gam... 📸'. At the very bottom, there are two buttons: 'Kembali' (white background) and 'Simpan Data' (blue background).

Gambar 3. 6 Formulir Monitoring Aset Aplikasi MAIS

Gambar 3.6 menunjukkan formulir monitoring aset yang harus diisi petugas secara manual. Pada formulir Monitoring ini, terdapat beberapa komponen input:

(1) Tanggal Monitoring – biasanya terisi otomatis dengan tanggal hari ini, namun bisa diedit secara manual; (2) Kondisi – pilihan kondisi aset (misalnya Baik, Rusak Ringan, Rusak Berat) yang dipilih dari daftar; (3) Catatan Inspeksi – teks bebas untuk mencatat hasil pengamatan atau tindakan (pada contoh di gambar terisi “bagus”); dan (4) Upload Bukti Foto – fitur unggah foto kondisi fisik aset sebagai evidensi. Petugas melakukan penilaian kondisi secara subjektif dan memasukkan catatan sesuai inspeksi lapangan. Selanjutnya, petugas dapat mengambil foto aset (misalnya foto perangkat dan nomor seri) lalu mengunggahnya melalui tombol Upload Gambar. Setelah semua data dirasa cukup, tombol Simpan Data ditekan untuk menyimpan hasil monitoring.

Form pencatatan kondisi di atas merefleksikan proses monitoring yang manual: kualitas data sangat bergantung pada ketelitian petugas dalam mengisi. Potensi kelemahan pada tahap ini antara lain: tidak ada validasi otomatis untuk memastikan semua input valid dan masuk akal. Sebagai contoh, sistem memungkinkan tanggal monitoring diubah secara manual; jika petugas keliru atau lalai, tanggal bisa tidak akurat. Kasus semacam ini tampak pada data yang tersimpan dengan tanggal anomali. Input teks catatan juga tidak tervalidasi panjang maupun formatnya – petugas bisa saja memasukkan informasi minimal (“bagus” saja) tanpa detail, dan sistem tetap menerimanya. Bahkan kondisi aset ditentukan hanya berdasarkan observasi petugas; tidak ada sensor atau mekanisme verifikasi otomatis yang memastikan kondisi “Baik” memang benar (misalnya perangkat benar-benar menyala atau berfungsi). Dengan demikian, kesalahan manual sangat mungkin terjadi tanpa terdeteksi. Hal ini sejalan dengan temuan umum bahwa proses pencatatan aset secara manual rentan terhadap kesalahan manusia dalam entri data . Prosedur seperti di atas juga mengandalkan kedisiplinan petugas, sebab sistem belum memberikan pengecekan otomatis (misalnya mencegah tanggal tidak logis atau mencegah kolom wajib kosong). Tanpa adanya kontrol tambahan, kualitas data monitoring sangat bergantung pada keakuratan input manual, yang menurut Hendrikson dapat menurunkan kualitas informasi secara keseluruhan



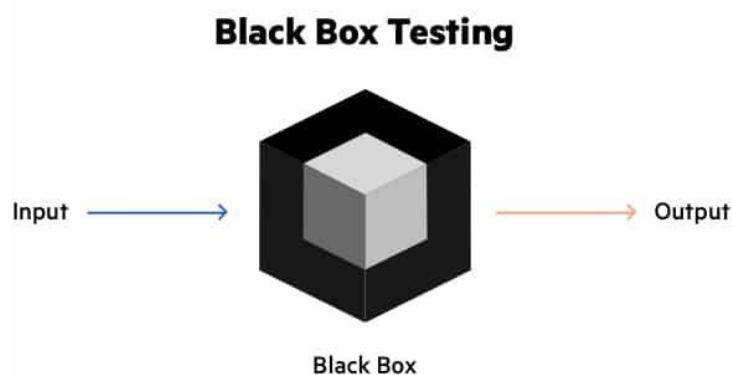
Gambar 3. 7 Layar List Data Monitoring Aplikasi MAIS

Setiap baris memuat identitas aset (kode dan nama barang, lokasi, umur) beserta tanggal terakhir dimonitor. Fungsi daftar ini adalah memudahkan petugas meninjau riwayat monitoring dan memastikan tiap aset terpantau secara berkala. Terlihat pada contoh, aset “Router/Switch Router” di Workshop SI telah dimonitor pada 12-08-2025, dan sebelumnya pada 17-10-2024. Demikian pula aset PC tercatat terakhir dimonitor 16-08-2024. Namun, tampak anomali pada salah satu entri: sebuah aset PC memiliki tanggal monitoring 30-11-0001. Tanggal yang tidak wajar ini menandakan data tidak valid yang tersimpan akibat ketiadaan validasi saat input – besar kemungkinan terjadi kesalahan input atau bug (misal pengguna lupa memilih tanggal sehingga sistem menyimpan nilai default tahun 0001). Contoh tersebut menggarisbawahi kelemahan proses manual: tanpa validasi otomatis, kesalahan input dapat langsung masuk ke database dan mengurangi akurasi informasi aset. Dalam jangka panjang, akumulasi kesalahan semacam ini akan

menyulitkan pelacakan aset dan pengambilan keputusan. Penelitian Kusumojati & Mediawati (2024) mencatat bahwa tanpa sistem informasi aset internal, institusi pendidikan tinggi mengalami kesulitan menelusuri data aset, dan rentan terhadap kesalahan pencatatan serta kehilangan data . Hal ini menunjukkan perlunya sistem yang lebih terstruktur untuk memastikan data aset (termasuk data hasil monitoring) lebih terorganisir, akurat, dan mudah ditelusuri. Secara umum, berbagai studi dan laporan menyimpulkan bahwa penggunaan sistem manual atau terpisah (seperti Excel) dalam manajemen aset menyebabkan proses menjadi tidak efisien, rawan kesalahan, dan menyulitkan konsolidasi data.

3.8 Evaluasi Kinerja

Evaluasi kinerja aplikasi merupakan tahapan penting dalam proses pengembangan perangkat lunak untuk memastikan bahwa aplikasi yang dihasilkan berfungsi sesuai dengan kebutuhan pengguna dan memiliki kualitas yang memadai. Dalam konteks proyek PKL ini, evaluasi kinerja aplikasi difokuskan pada dua aspek utama, yaitu pengujian fungsional menggunakan metode *blackbox*.



Gambar 3. 8 Black Box Testing

Pengujian fungsional dilakukan dengan metode *blackbox testing*, yaitu dengan menguji aplikasi berdasarkan input dan output yang dihasilkan tanpa memperhatikan struktur internal dari kode program. Metode ini bertujuan untuk memastikan bahwa fitur-fitur utama seperti pencatatan data aset, pengambilan gambar, dan pembaruan informasi aset dapat berjalan dengan baik sesuai dengan skenario penggunaan yang telah ditentukan. Hasil pengujian ini memberikan

gambaran mengenai sejauh mana aplikasi dapat memenuhi spesifikasi fungsional yang telah dirancang sebelumnya.

Pengujian kinerja merupakan salah satu jenis pengujian non-fungsional yang bertujuan mengevaluasi seberapa baik suatu aplikasi bekerja dari segi responsivitas, stabilitas, dan efisiensi di bawah beban kerja tertentu . Dalam konteks rekayasa perangkat lunak, pengujian kinerja dilakukan untuk memastikan aplikasi dapat merespons dengan cepat dan tetap stabil ketika dihadapkan pada skenario penggunaan nyata, termasuk jumlah data atau pengguna yang tinggi. Menurut Lestari (2024), pengujian kinerja mengevaluasi responsivitas dan efisiensi aplikasi di bawah beban kerja yang berbeda . Artinya, metrik seperti waktu respons, throughput, pemanfaatan sumber daya (CPU, memori), serta durabilitas aplikasi ketika diberi beban berlebih (stress test) menjadi fokus utama. Tanpa pengujian kinerja, sebuah aplikasi mungkin berfungsi benar secara logika namun gagal memenuhi ekspektasi pengguna karena lambat atau sering hang ketika digunakan.

Pada aplikasi mobile, pengujian kinerja memiliki tantangan dan pendekatan tersendiri. Perangkat mobile memiliki keterbatasan sumber daya (CPU, memori, dan baterai) serta variasi spesifikasi hardware yang luas, sehingga aplikasi harus diujikan pada berbagai kondisi. Norberto et al. (2019) melalui pemetaan literatur sistematis mengidentifikasi berbagai alat, metode, dan strategi yang digunakan dalam pengujian kinerja aplikasi mobile, termasuk teknik simulasi beban, penggunaan profiling tools, dan otomasi pengujian di berbagai platform . Pendekatan-pendekatan tersebut mencakup analisis metrik spesifik mobile seperti waktu startup, penggunaan memori, jumlah frame per detik (FPS) pada animasi UI, hingga konsumsi daya baterai (Ramu, 2023) . Melalui kombinasi metode tersebut, pengembang dapat mengidentifikasi bottleneck kinerja dan memastikan aplikasi mobile tetap responsif serta efisien dalam kondisi operasional sesungguhnya.

Beberapa parameter kinerja kunci yang umum dijadikan acuan dalam pengujian aplikasi mobile antara lain: waktu muat aplikasi (startup time), penggunaan memori saat aplikasi berjalan, responsivitas antarmuka pengguna (misalnya diukur dari kestabilan frame rate), serta kinerja jaringan (jika aplikasi terhubung ke API) . Pengembang perlu menetapkan tolok ukur atau standar untuk tiap metrik tersebut. Sebagai contoh, waktu startup yang ideal untuk aplikasi mobile

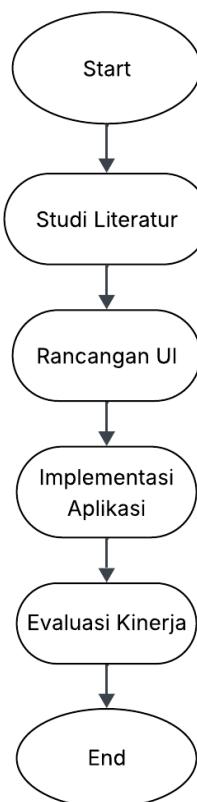
biasanya berada di bawah beberapa detik pertama; menurut pedoman performa Android, peluncuran dingin (cold start) sebaiknya di bawah 5 detik untuk memastikan pengalaman pengguna yang baik . Demikian pula, penggunaan memori perlu dijaga agar tidak berlebihan – aplikasi yang menghabiskan ratusan MB memori berpotensi menyebabkan perangkat kelas menengah kebawah menjadi lambat atau crash. Oleh karena itu, praktik optimal mencakup optimasi kode (misalnya menghapus objek yang tidak terpakai, mengoptimalkan gambar), penggunaan teknik caching, dan memanfaatkan fitur profiling pada Android Studio atau DevTools Flutter untuk memantau penggunaan memori dan CPU selama pengujian. Dengan melakukan pengujian kinerja secara sistematis, tim pengembang dapat menemukan dan memperbaiki potensi masalah kinerja sejak dini, memastikan bahwa aplikasi yang dibangun tidak hanya benar secara fungsional tetapi juga cepat, ringan, dan andal saat digunakan pengguna akhir .

BAB IV

HASIL DAN PEMBAHASAN

4.1 Alur Penelitian

Proyek Praktik Kerja Lapangan ini bertujuan untuk merancang aplikasi pengelolaan dan manajemen aset. Alur penelitian yang digunakan dapat dilihat pada gambar 3.1 berikut.



Gambar 4. 1 Alur Penelitian

4.2 Studi Literatur

Pada tahap ini dilakukan studi pustaka terhadap penelitian dan sumber-sumber relevan yang mendukung proses perancangan dan pengembangan aplikasi mobile pencatatan aset. Studi literatur difokuskan pada pemahaman terhadap penggunaan framework Flutter dan lingkungan Android Studio dalam pengembangan aplikasi lintas platform, serta pemanfaatan plugin seperti sqflite, image_picker, dan flutter_bloc untuk mendukung fungsionalitas dan struktur

aplikasi. Selain itu, ditelaah pula konsep evaluasi perangkat lunak melalui metode blackbox testing dan usability testing untuk menilai fungsionalitas serta kemudahan penggunaan aplikasi. Studi ini menjadi landasan teoritis yang penting dalam pemilihan teknologi dan perancangan fitur aplikasi yang dikembangkan selama kegiatan Praktik Kerja Lapangan.

Tabel 4. 1 Studi Literatur

No	Topik	Pengetahuan			Temuan
1	Flutter sebagai framework pengembangan aplikasi	Apa keunggulan Flutter dalam pengembangan aplikasi mobile	keunggulan dalam pengembangan aplikasi mobile	Flutter memiliki performa tinggi, lintas platform?	memungkinkan pengembangan satu basis kode untuk Android dan iOS, mendukung UI deklaratif, dan dilengkapi fitur hot reload yang mempercepat proses pengembangan. (Flutter, 2022; Google Developers, 2023)
2	Android Studio sebagai lingkungan pengembangan	Apa peran Android Studio dalam pengembangan aplikasi mobile Flutter?	peran Android Studio dalam pengembangan aplikasi mobile	Android Studio merupakan IDE resmi Google yang mendukung Flutter melalui plugin khusus, memungkinkan pengujian dan debugging langsung pada emulator maupun perangkat fisik.	(Google Developers, 2023)
3	SQLite dalam aplikasi mobile	Bagaimana SQLite digunakan untuk menyimpan data lokal dalam aplikasi Flutter?	digunakan untuk menyimpan data lokal dalam aplikasi	Plugin sqflite digunakan untuk menyimpan data asset secara lokal tanpa perlu koneksi internet, cocok untuk kebutuhan mobile di lapangan.	(sqlite.org, 2022)

Tabel 4. 2 Studi Literatur (Lanjutan)

No	Topik	Pengetahuan	Temuan
4	Evaluasi fungsional aplikasi mobile (Blackbox)	Bagaimana menguji fungsiонаlitas aplikasi dari sisi pengguna?	Blackbox testing menguji output dari fitur berdasarkan input pengguna tanpa melihat kode internal, cocok untuk memastikan fitur dasar berjalan sesuai. (Sommerville, 2016)
5	Studi kasus aplikasi manajemen aset mobile	Apakah aplikasi manajemen aset mobile telah digunakan di institusi pendidikan?	Beberapa studi menunjukkan aplikasi berbasis Android atau multiplatform dapat meningkatkan efisiensi pencatatan aset kampus. (Riyanto, 2019; Ahmad & Maulana, 2020; Kholid & Huda, 2019)

4.3 Analisis Kebutuhan

Dalam pelaksanaan kegiatan Praktik Kerja Lapangan yang berfokus pada pengembangan aplikasi mobile pencatatan kerusakan aset di Universitas Ma Chung, dilakukan analisis kebutuhan guna memastikan bahwa proses perancangan dan implementasi sistem berjalan secara efektif dan sesuai dengan tujuan. Analisis ini mencakup identifikasi kebutuhan dari sisi pengguna akhir dan pengembang, serta perangkat keras, perangkat lunak, dan data pendukung yang dibutuhkan selama proses pengembangan. Perlu diketahui bahwa seluruh data berupa citra aset kampus diperoleh melalui izin dari Bagian Aset Universitas Ma Chung, dan dokumentasi citra dilakukan secara manual oleh penulis melalui pengambilan foto langsung di lapangan.

4.3.1 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna dilakukan untuk memahami kebutuhan, harapan, serta kendala yang dihadapi oleh pengguna akhir dalam menjalankan

tugasnya. Dalam konteks pengembangan aplikasi pencatatan kerusakan aset di lingkungan Universitas Ma Chung, pengguna utama aplikasi ini adalah petugas dari Bagian Aset dan staf dari SIPUSDA (Unit Sistem Informasi dan Pusat Data) yang bekerja sama dalam pengelolaan data aset kampus.

Berdasarkan hasil observasi lapangan dan diskusi dengan pihak-pihak terkait, diketahui bahwa pengguna membutuhkan sistem yang dapat mempermudah proses dokumentasi kerusakan aset secara cepat dan efisien. Fitur utama yang dibutuhkan adalah kemampuan untuk mengambil gambar kerusakan secara langsung melalui kamera perangkat, serta menyimpannya ke dalam sistem sebagai bukti visual. Dokumentasi ini akan mempermudah proses pelaporan dan tindak lanjut tanpa perlu melakukan verifikasi fisik secara langsung ke lokasi aset. Untuk meningkatkan efisiensi, dibutuhkan pula mekanisme otomasi verifikasi melalui integrasi model deteksi kerusakan berbasis AI. Mekanisme ini memungkinkan sistem untuk secara otomatis mengidentifikasi dan mengklasifikasi jenis kerusakan dari gambar yang diunggah, sehingga proses verifikasi tidak sepenuhnya bergantung pada pengecekan manual oleh petugas. Hasil verifikasi otomatis ini dapat langsung digunakan sebagai acuan awal dalam pelaporan, sementara petugas hanya perlu melakukan pengecekan tambahan jika diperlukan, sehingga mempercepat proses dan meminimalkan potensi keterlambatan tindak lanjut.

Aplikasi juga diharapkan memiliki antarmuka yang sederhana dan mudah dipahami, sehingga dapat digunakan oleh staf non-teknis tanpa memerlukan pelatihan khusus. Untuk menunjang fleksibilitas penggunaan, sistem perlu mendukung penyimpanan data secara lokal sehingga tetap dapat digunakan meskipun tidak terhubung ke jaringan internet. Pengguna juga memerlukan akses terhadap riwayat dokumentasi yang telah dicatat sebelumnya untuk mendukung monitoring dan pengambilan keputusan terkait perawatan atau penggantian aset.

Di samping itu, performa aplikasi juga menjadi perhatian utama. Aplikasi harus dapat berjalan secara optimal pada berbagai jenis perangkat Android, termasuk perangkat dengan spesifikasi menengah ke bawah. Hasil dari analisis ini menjadi acuan dalam perancangan fitur serta alur kerja aplikasi, sehingga solusi yang dikembangkan dapat menjawab kebutuhan nyata pengguna akhir di lapangan.

4.3.2 Kebutuhan Peneliti

Berikut kebutuhan perangkat keras dan lunak yang akan dibutuhkan dan digunakan oleh peneliti dalam mengerjakan penelitian ini.

1. Perangkat Keras

- d. PC
 - i. Prosesor : Intel i3-9100F 3.60 GHz
 - ii. RAM: 16 GB
 - iii. SSD: 1 TB
 - iv. GPU: NVIDIA Geforce RTX 3050
 - v. Sistem Operasi: Windows 10 Pro 64 bit
- e. Smartphone (Perangkat Uji Aplikasi)
 - i. Merek dan Tipe: Xiaomi Redmi Note 12 Pro 4G
 - ii. Prosesor: RAM 8GB
 - iii. Penyimpanan Internal: 256 GB
 - iv. Sistem Operasi: Android 13
 - v. Fitur Tambahan: Mendukung debugging USB dan instalasi aplikasi melalui Android Studio

2. Perangkat Lunak

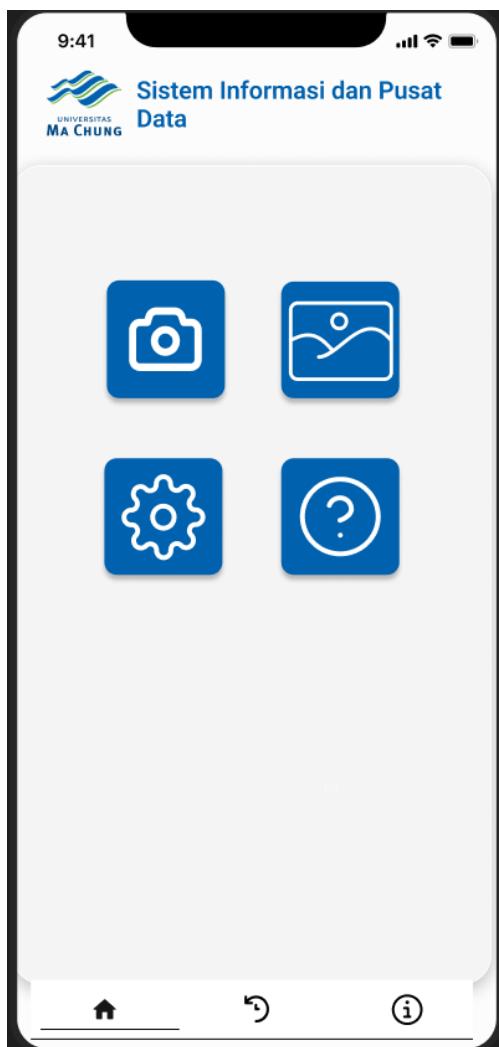
- a. Android Studio
- b. Flutter
- c. SQLite
- d. BLoC
- e. Path_provider
- f. Data

4.4 Rancangan UI

Perancangan antarmuka pengguna dilakukan dengan mempertimbangkan kebutuhan pengguna yang telah dianalisis sebelumnya. Antarmuka aplikasi dirancang agar sederhana, fungsional, dan dapat digunakan oleh pengguna non-teknis tanpa pelatihan khusus. Setiap elemen UI dikembangkan untuk memfasilitasi proses dokumentasi aset secara efisien, termasuk navigasi antar halaman, input data, serta pengambilan dan penyimpanan gambar kerusakan aset.

Antarmuka aplikasi pendekripsi kerusakan aset ini dirancang secara sistematis untuk memberikan pengalaman pengguna yang intuitif dan fungsional. Saat

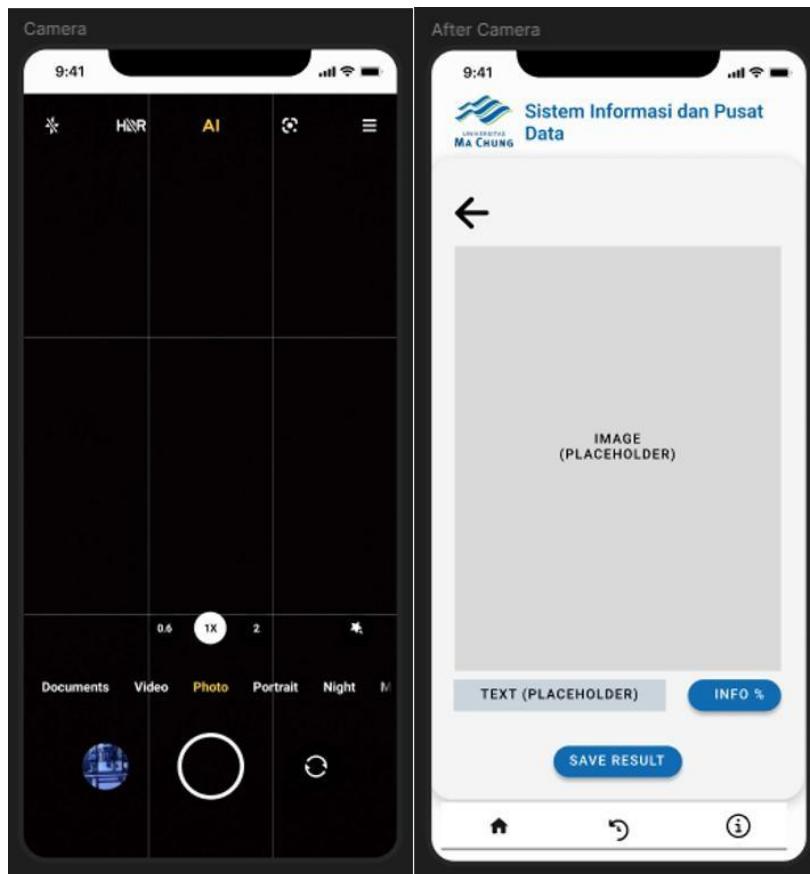
aplikasi pertama kali dijalankan, pengguna akan diarahkan ke tampilan awal atau splash screen yang menampilkan identitas institusi, yaitu Universitas Ma Chung. Setelah beberapa detik, aplikasi akan secara otomatis beralih ke halaman utama atau beranda (home screen).



Gambar 4. 1 Home Screen

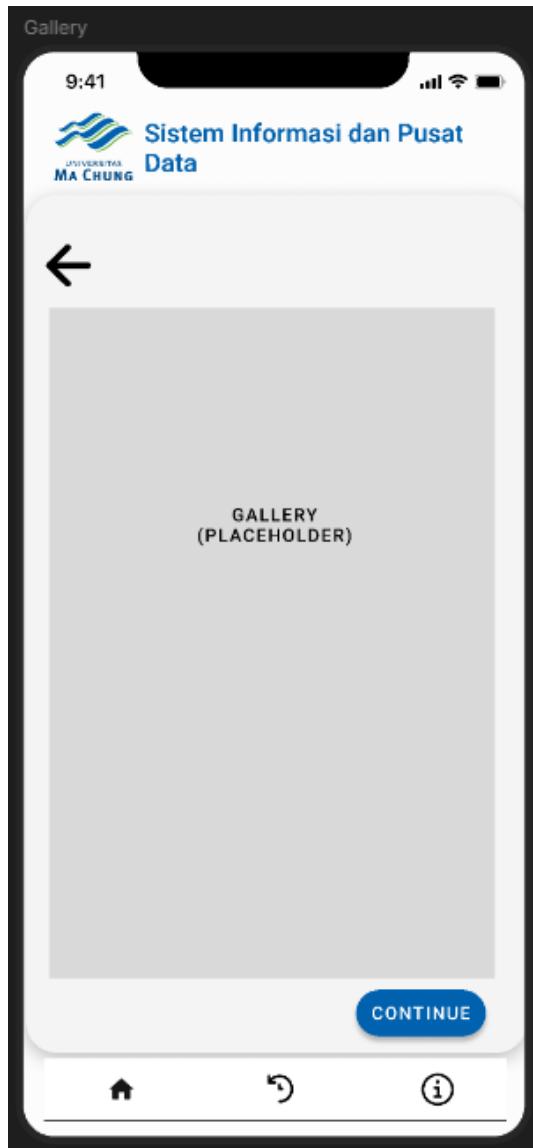
Pada halaman utama, pengguna disajikan dengan empat tombol utama yang masing-masing mewakili fitur utama aplikasi, yaitu: Kamera, Galeri, Konfigurasi API, dan Tutorial. Tombol-tombol ini disusun secara grid untuk memudahkan akses dan memperjelas fungsi setiap fitur. Tombol Kamera dan Galeri berfungsi sebagai metode input citra untuk proses deteksi kerusakan aset, sedangkan tombol Konfigurasi API memungkinkan pengguna mengatur model deteksi yang digunakan, dan tombol Tutorial memberikan panduan penggunaan aplikasi. Di bagian bawah layar, terdapat navigasi bawah (*bottom navigation bar*) yang

terdiri dari tiga ikon: Home, History, dan Info, yang selalu tersedia di setiap halaman untuk memudahkan perpindahan antar fitur utama aplikasi.



Gambar 4. 2 Fungsi Camera

Jika pengguna memilih tombol kamera, aplikasi akan membuka aplikasi kamera bawaan perangkat. Di sini, pengguna dapat memotret objek atau aset yang ingin dianalisis. Setelah gambar diambil, aplikasi akan mengalihkan pengguna ke halaman “After Camera”, yaitu tampilan hasil deteksi. Pada halaman ini, aplikasi akan menampilkan klasifikasi objek (Rusak atau Tidak Rusak) berdasarkan hasil inferensi model Yolo, serta confidence score dalam bentuk persentase yang menunjukkan tingkat kepercayaan sistem terhadap hasil deteksi tersebut. Di bawahnya tersedia tombol “Save Result” yang memungkinkan pengguna menyimpan hasil ke menu riwayat.



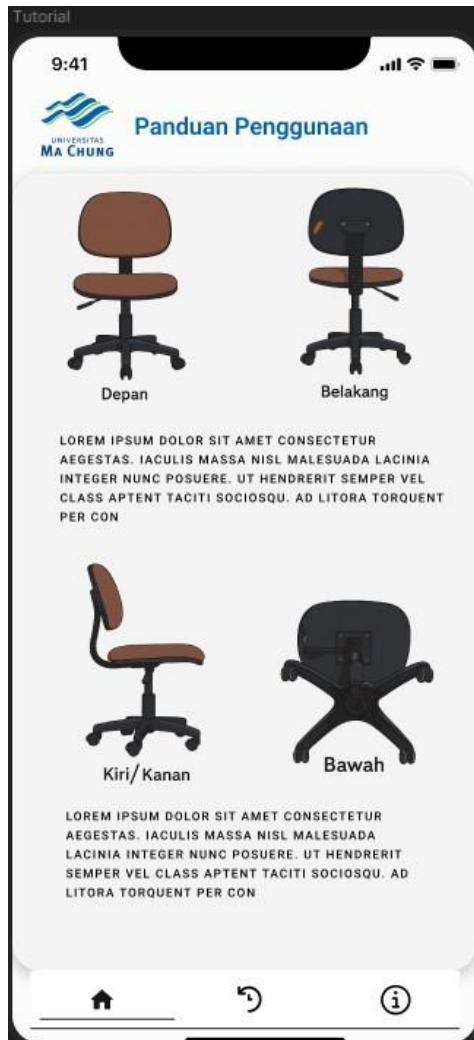
Gambar 4. 3 Fungsi Gallery

Sementara itu, tombol galeri mengarahkan pengguna ke penyimpanan galeri perangkat, di mana mereka dapat memilih foto yang telah ada untuk dianalisis. Setelah pengguna memilih gambar dari galeri, aplikasi akan menampilkan hasil deteksi seperti pada mode kamera: klasifikasi kerusakan dan confidence score.



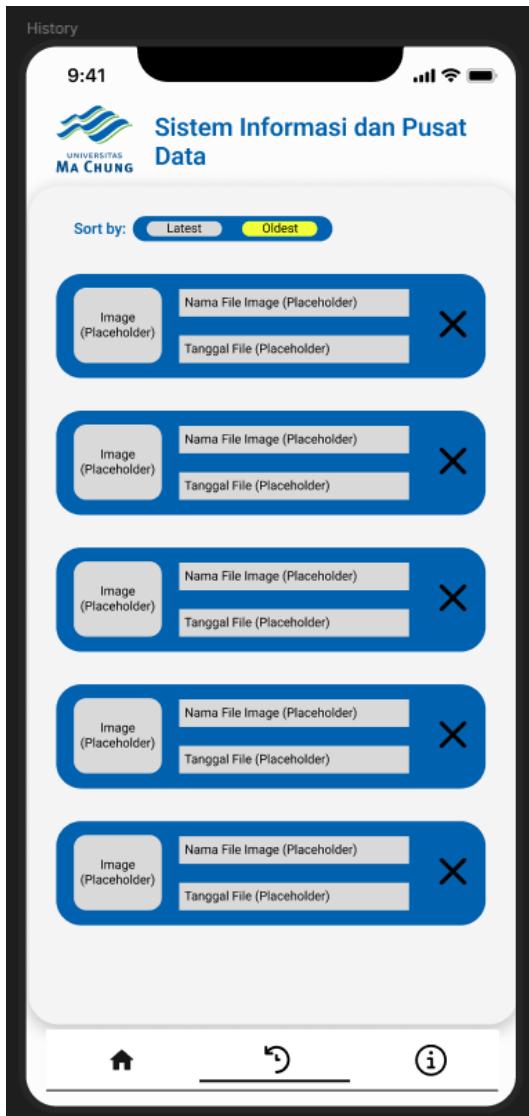
Gambar 4. 4 Konfigurasi API

Halaman Konfigurasi API merupakan fitur yang dirancang untuk memberikan fleksibilitas kepada pengguna dalam memilih model deteksi kerusakan aset yang digunakan aplikasi. Melalui halaman ini, pengguna dapat memasukkan *Model ID* dan *API Key* yang diperoleh dari layanan Ultralytics secara manual. Setiap pasangan Model ID dan API Key mewakili satu konfigurasi model yang dapat digunakan untuk proses inferensi gambar. Tersedia tombol tambah (“+”) untuk menambahkan konfigurasi baru, serta tombol hapus (“X”) untuk menghapus konfigurasi yang tidak diinginkan. Konfigurasi yang dimasukkan akan disimpan secara lokal, sehingga tetap tersedia meskipun aplikasi ditutup atau perangkat dimatikan. Fitur ini memungkinkan aplikasi tetap adaptif dan dapat mengikuti perkembangan model deteksi tanpa perlu dilakukan pembaruan versi aplikasi secara menyeluruh.



Gambar 4. 5 Tutorial Screen

Halaman Tutorial berfungsi sebagai panduan visual bagi pengguna dalam melakukan pengambilan gambar aset secara tepat. Halaman ini menampilkan ilustrasi kursi dari empat sudut pandang utama: depan, belakang, kiri/kanan, dan bawah. Setiap posisi disertai dengan keterangan arah guna memastikan bahwa petugas lapangan memahami sudut pengambilan gambar yang sesuai. Panduan ini penting karena kualitas dan sudut gambar yang baik akan sangat memengaruhi akurasi hasil deteksi dari model. Dengan memberikan referensi visual yang jelas, halaman Tutorial membantu pengguna, khususnya petugas non-teknis, untuk melakukan dokumentasi kerusakan aset secara terstandardisasi dan konsisten, sehingga mendukung proses deteksi dan pelaporan yang lebih akurat dan efisien.



Gambar 4. 6 Tab Results

Semua hasil yang telah disimpan melalui mode kamera atau galeri akan muncul di menu “Results”. Pada halaman ini, pengguna dapat melihat riwayat hasil identifikasi sebelumnya secara kronologis. Aplikasi menyediakan fitur sortir berdasarkan tanggal (dari yang terbaru maupun yang terlama), serta opsi untuk menghapus hasil secara satuan atau keseluruhan. Setiap entri hasil menampilkan gambar mini (*thumbnail*), nama file atau asset, dan tanggal penyimpanan.



Gambar 4. 7 Tab Information

Terakhir, aplikasi juga menyediakan halaman “Information” atau “About” yang berisi informasi mengenai pengembang aplikasi, logo institusi, versi aplikasi, dan kontak resmi. Menu ini mencerminkan latar belakang akademik dari aplikasi, mencantumkan nama-nama mahasiswa pengembang serta dosen pembimbing atau kepala unit yang bertanggung jawab atas proyek ini.

Perlu dicatat bahwa antarmuka pengguna (UI) yang ditampilkan dalam perancangan aplikasi mungkin tidak sepenuhnya identik dengan tampilan akhir pada versi aplikasi yang dihasilkan.

4.5 Implementasi Aplikasi

Implementasi aplikasi dilakukan menggunakan framework Flutter dengan bahasa pemrograman Dart. Proyek ini terdiri dari beberapa layar (screen) utama yang dibangun menggunakan pendekatan komponen dan navigasi bottom bar. Beberapa file utama dalam pengembangan meliputi main.dart sebagai titik masuk aplikasi, serta file lain seperti home_screen.dart, history_screen.dart, info_screen.dart, result_screen.dart, dan welcome_screen.dart yang mengatur tampilan dan logika antar halaman.

Aplikasi memiliki lima tampilan utama, yaitu:

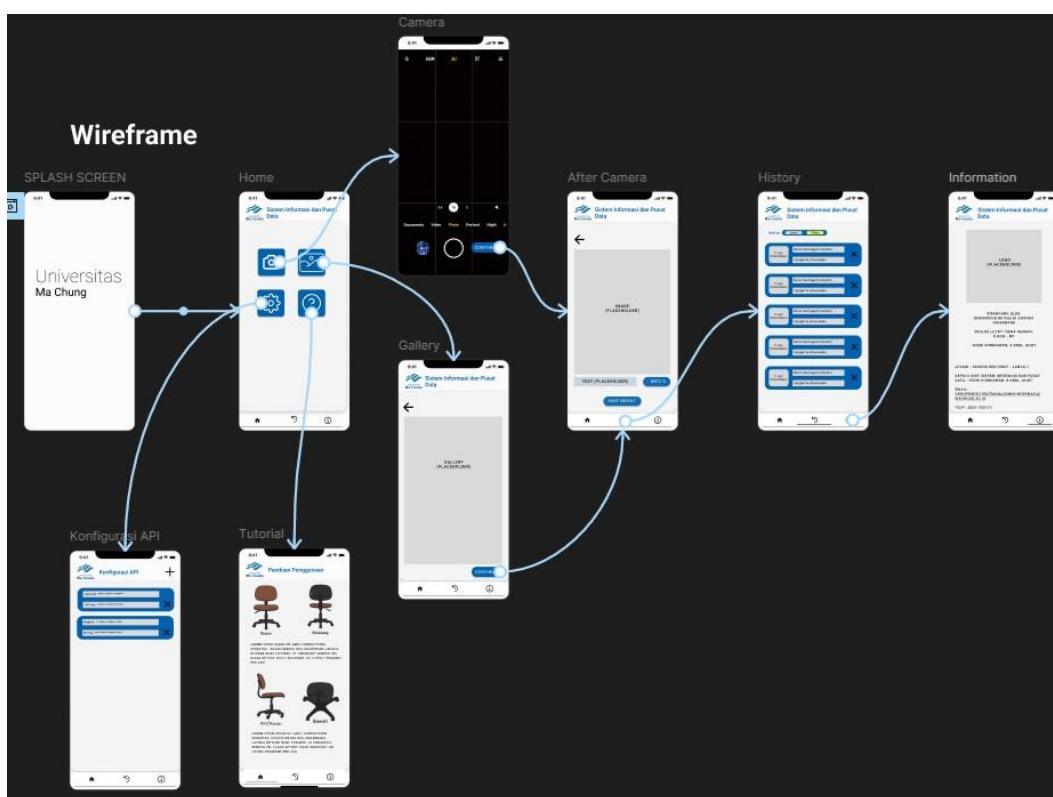
1. Splash Screen dan Welcome Screen: Halaman pembuka yang menampilkan logo dan nama institusi, digunakan sebagai pengantar sebelum masuk ke menu utama.
2. Home Screen: Berfungsi sebagai halaman utama dengan dua tombol untuk memilih gambar dari galeri atau kamera. Tampilan ini menekankan kesederhanaan dan kemudahan akses.
3. Result Screen: Halaman yang menampilkan hasil deteksi kerusakan dari API. Gambar akan ditampilkan bersama bounding box berisi label dan skor deteksi (contoh: “Jamur 39.6%”).
4. History Screen: Berisi daftar riwayat hasil deteksi yang tersimpan secara lokal. Setiap entri menampilkan nama file, tanggal, dan gambar pratinjau, serta fitur hapus data.
5. Info Screen: Menampilkan informasi institusi, perancang aplikasi, serta kontak SIPUSDA yang dapat dihubungi.

Aplikasi ini memanfaatkan plugin seperti image_picker untuk pengambilan gambar, path_provider untuk akses penyimpanan lokal, dan sqflite sebagai database lokal untuk menyimpan hasil deteksi.

Navigasi dilakukan melalui bottom navigation bar dengan ikon dan label (Home, History, Info) agar pengguna dapat berpindah antar halaman dengan mudah. Implementasi logika visualisasi deteksi dilakukan melalui file bounding_box_painter.dart, yang bertugas menggambar kotak dan label hasil deteksi.

Secara keseluruhan, implementasi dilakukan dengan pendekatan modular dan penggunaan state management sederhana. Fitur deteksi mengandalkan pemanggilan ke API eksternal (api_helper.dart) yang mengirim gambar ke server dan menerima hasil dalam format bounding box dan label.

Tampilan antarmuka telah disesuaikan dengan gaya desain institusi dan diuji secara fungsional melalui pengujian blackbox serta evaluasi usability terbatas. Hasil implementasi menunjukkan bahwa aplikasi dapat berjalan dengan baik dan memenuhi kebutuhan dokumentasi kerusakan aset oleh pihak Bagian Aset Universitas Ma Chung.



Gambar 4. 8 Wireframe Alur Aplikasi

Diagram ini menunjukkan rancangan navigasi dan antarmuka aplikasi secara keseluruhan. Terlihat hubungan antar layar utama: mulai dari Splash Screen awal yang menampilkan logo Universitas, kemudian menuju Home (beranda) sebagai pusat menu. Dari halaman Home, pengguna dapat mengakses fitur Kamera atau Galeri untuk memasukkan foto aset. Setelah pengguna mengambil foto melalui kamera, aplikasi menampilkan layar After Camera berupa pratinjau hasil dan opsi penyimpanan. Alur dari kamera berlanjut ke proses deteksi otomatis sebelum hasil

ditampilkan. Sebaliknya, jika pengguna memilih foto dari galeri, alurnya menuju proses deteksi yang sama. Diagram ini juga memperlihatkan halaman History (Riwayat) yang dapat diakses melalui navigasi bawah untuk melihat daftar hasil deteksi tersimpan, serta halaman Information yang berisi detail informasi aset atau tentang aplikasi. Selain itu, terdapat halaman Konfigurasi API untuk pengaturan model AI yang digunakan, dan halaman Tutorial yang memberikan panduan penggunaan aplikasi. Panah-panah pada diagram menggambarkan transisi atau perpindahan antar layar; misalnya, dari Home ke Kamera/Galeri, lalu ke Hasil Deteksi, dan seterusnya. Dengan diagram wireframe ini, alur interaksi pengguna dengan aplikasi dapat dipahami secara menyeluruh sebelum aplikasi diimplementasikan.

4.5.1 Alur Kerja Deteksi dan Integrasi API

Fitur utama dari aplikasi ini adalah proses deteksi kerusakan aset berdasarkan gambar yang diambil melalui kamera atau dipilih dari galeri. Setelah pengguna memilih atau mengambil gambar, aplikasi akan mengirimkan file gambar tersebut ke server menggunakan metode *POST* melalui plugin `http`. Pengiriman dilakukan dalam bentuk *byte array (multipart)* dan kemudian diproses oleh API eksternal yang berbasis model deteksi objek.

Proses ini diatur melalui file `api_helper.dart` yang menangani beberapa fungsi penting, yaitu: pengubahan file menjadi format *base64* atau *byte stream*, pemanggilan *endpoint* API menggunakan `http.post()`, penanganan *response* dalam bentuk *JSON*, serta pengambilan data seperti *bounding box*, label objek, dan *confidence score*. Respons dari API berisi hasil deteksi dalam format koordinat dan label objek. Hasil ini kemudian ditampilkan pada layar hasil (`result_screen.dart`) menggunakan metode visualisasi *bounding box* melalui file `bounding_box_painter.dart`. Gambar asli akan ditampilkan bersamaan dengan anotasi hasil deteksi, misalnya seperti “Jamur 39.6%”, yang memberikan informasi awal mengenai kondisi kerusakan aset.

Proses ini memungkinkan terjadinya integrasi antara antarmuka pengguna yang sederhana dengan fungsionalitas backend yang dinamis. Sistem ini dirancang agar dapat dikembangkan lebih lanjut pada tahap Laporan Praktik Kerja Lapangan.

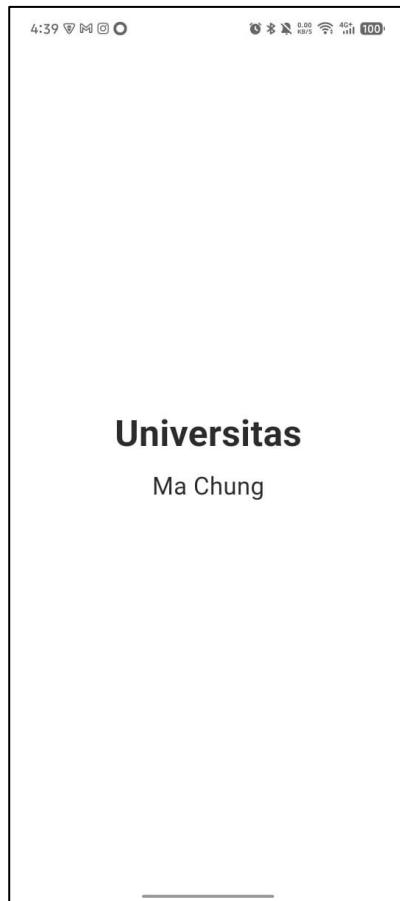
Pada tahap tersebut, model deteksi kerusakan berbasis *Convolutional Neural Network* (CNN) akan dirancang dan dilatih secara khusus untuk mengenali berbagai jenis kerusakan pada aset kampus, seperti jamur, sobekan, atau retakan.

Model CNN akan dibangun menggunakan arsitektur populer seperti YOLOv11, dan selanjutnya akan *di-deploy* dalam bentuk API melalui layanan cloud berbasis Ultralytics API. Aplikasi mobile ini telah dirancang untuk mendukung integrasi dengan model tersebut melalui *endpoint* yang dapat disesuaikan. Dengan demikian, fitur deteksi yang saat ini masih menggunakan model placeholder atau prototipe, nantinya akan diperkuat oleh model hasil pelatihan lokal menggunakan dataset nyata dari kampus. Selain itu, sistem juga dirancang agar mendukung pemilihan model secara dinamis (*model switching*), misalnya dengan menyediakan opsi pemilihan versi model terbaru melalui panel admin atau API gateway. Langkah ini memungkinkan pembaruan model tanpa perlu membangun ulang aplikasi mobile, cukup dengan mengganti *endpoint* atau token akses yang digunakan. Dengan rancangan terbuka seperti ini, aplikasi akan menjadi lebih fleksibel dan berkelanjutan sebagai sistem pendukung manajemen aset berbasis AI di masa depan. Fitur-fitur seperti deteksi lanjutan dan pemilihan model AI secara fleksibel dapat terus dikembangkan. Hasil evaluasi *usability* yang diperoleh dari pengguna juga akan dijadikan acuan untuk perbaikan desain pada iterasi pengembangan berikutnya.

4.6 Evaluasi Kinerja

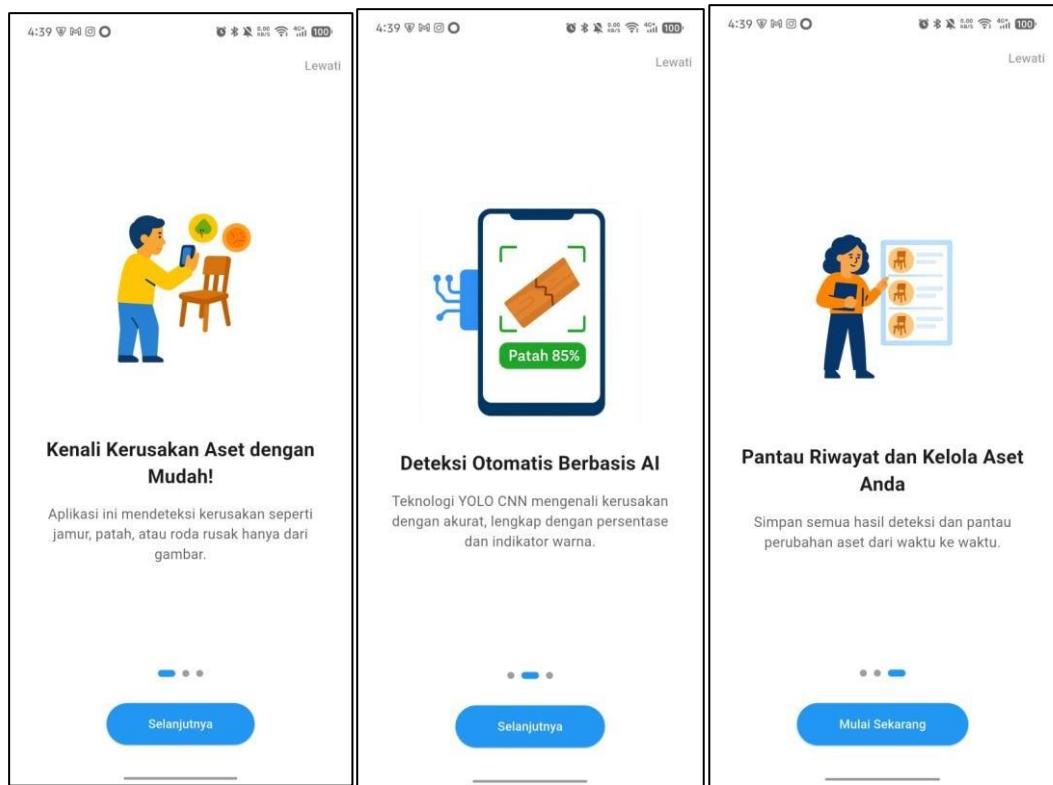
Pengujian blackbox dilakukan dengan menilai apakah setiap fungsi dalam sistem bekerja sesuai dengan tujuan, tanpa perlu mengetahui bagaimana cara kerja internal atau implementasi teknis dari fungsi tersebut. Dalam pengujian ini, fokus diarahkan pada fitur-fitur utama yang tersedia dalam aplikasi, dan hasil yang ditampilkan dibandingkan dengan keluaran yang diharapkan. Jika hasil aktual sesuai dengan ekspektasi, maka dapat disimpulkan bahwa aplikasi telah berjalan sesuai dengan perancangan awal. Namun, apabila ditemukan ketidaksesuaian, maka diperlukan proses evaluasi dan perbaikan lebih lanjut. Berikut merupakan hasil pengujian blackbox terhadap sistem yang telah dikembangkan.

4.6.1 Pengujian Halaman Utama



Gambar 4. 9 Splash Screen Prototype

Saat aplikasi pertama kali dibuka, pengguna akan disambut oleh tampilan awal berupa splash screen yang menampilkan identitas institusi, yaitu *Universitas Ma Chung*. Halaman ini berfungsi sebagai pengenal aplikasi dan memberikan kesan profesional sebelum pengguna masuk ke fitur utama.



Gambar 4. 10 Onboarding Screen

Setelah itu, pengguna diarahkan ke halaman onboarding pertama yang memperkenalkan fungsi utama aplikasi, yaitu kemudahan dalam mengenali kerusakan aset seperti jamur, patah, atau roda rusak hanya dari gambar. Pengguna cukup mengambil foto menggunakan kamera atau memilih gambar dari galeri, dan aplikasi akan memprosesnya secara otomatis.

Pada halaman selanjutnya, aplikasi menjelaskan bahwa deteksi kerusakan dilakukan secara otomatis menggunakan teknologi berbasis YOLO CNN (You Only Look Once Convolutional Neural Network). Teknologi ini memungkinkan aplikasi mengenali jenis kerusakan dengan tingkat akurasi yang tinggi, lengkap dengan persentase keyakinan dan indikator warna yang membantu pengguna memahami tingkat kerusakan secara visual.

Halaman terakhir onboarding menjelaskan fitur pelacakan riwayat deteksi. Setiap hasil analisis akan disimpan secara otomatis dan dapat diakses kembali kapan saja, sehingga pengguna dapat memantau perkembangan atau perubahan kondisi aset dari waktu ke waktu. Dengan fitur ini, aplikasi mendukung pengelolaan

aset secara berkelanjutan dan lebih efisien. Setelah onboarding selesai, pengguna dapat langsung memulai penggunaan aplikasi melalui tombol "Mulai Sekarang".

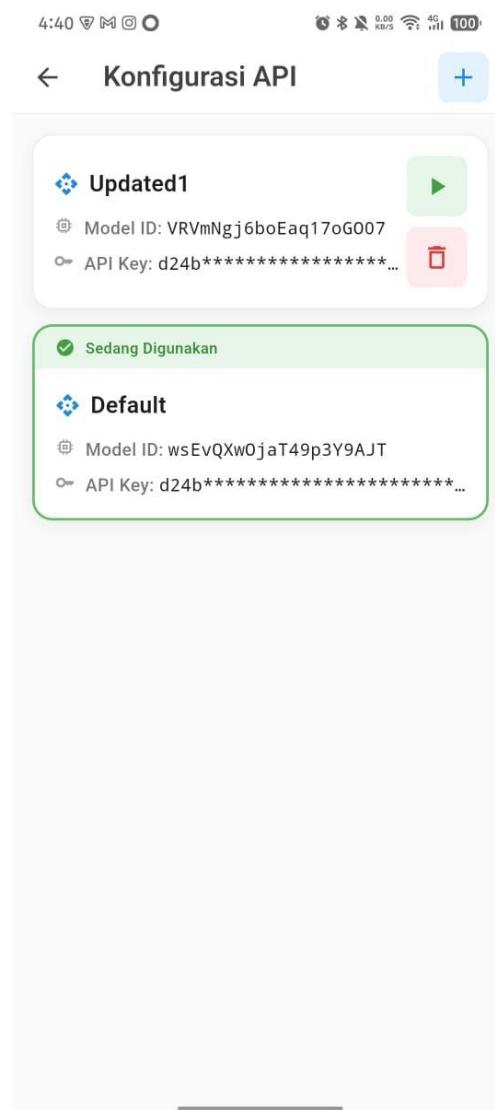
Seluruh tahapan onboarding ini telah diuji menggunakan metode black-box testing, dan hasilnya menunjukkan bahwa semua komponen—termasuk navigasi antar halaman, tampilan ilustrasi, serta tombol interaksi seperti "Selanjutnya" dan "Mulai Sekarang"—berfungsi dengan baik sesuai dengan yang diharapkan. Hal ini menegaskan bahwa fitur onboarding telah siap digunakan secara optimal oleh pengguna.



Gambar 4. 11 Fungsi Home Screen

Pengguna menekan tombol "Ambil Foto" atau "Pilih dari Galeri" di halaman utama. Sistem kemudian membuka kamera atau galeri dan menampilkan gambar yang dipilih di layar hasil. Jika gambar berhasil ditampilkan, maka fitur dinyatakan berjalan sesuai. Aplikasi mengirim gambar ke server API Ultralytics. Bila hasil berupa gambar dengan bounding box dan label muncul di layar, maka fungsi

deteksi dianggap berhasil.



Gambar 4. 11 Konfigurasi API

Fitur Konfigurasi API dirancang untuk memberikan fleksibilitas kepada pengguna dalam mengelola dan memilih model deteksi yang digunakan oleh aplikasi. Melalui antarmuka ini, pengguna dapat menambahkan, menghapus, maupun mengaktifkan konfigurasi model deteksi berbasis layanan cloud Ultralytics secara dinamis tanpa perlu melakukan pembangunan ulang (rebuild) aplikasi.

Pada halaman utama konfigurasi, sistem menampilkan daftar konfigurasi yang telah disimpan. Setiap entri konfigurasi memuat informasi penting berupa nama konfigurasi, Model ID, dan API key yang ditampilkan sebagian untuk

menjaga keamanan. Konfigurasi yang sedang aktif ditandai secara visual dengan bingkai berwarna hijau serta label “Sedang Digunakan”. Pengguna dapat mengganti model yang digunakan dengan menekan tombol aktivasi yang tersedia pada salah satu konfigurasi. Selain itu, tersedia pula opsi penghapusan yang disertai dengan dialog konfirmasi untuk memastikan bahwa tindakan tersebut tidak dilakukan secara tidak sengaja.

Untuk menambahkan konfigurasi baru, pengguna dapat menekan tombol tambah di bagian atas layar yang akan mengarahkan ke halaman “Tambah Konfigurasi API”. Pada halaman ini, pengguna diminta mengisi nama konfigurasi, API key yang valid, dan Model ID yang sesuai. Tersedia juga informasi tambahan berupa tips yang mengingatkan pengguna untuk memastikan bahwa API key memiliki izin akses terhadap model yang dituju dan agar informasi tersebut disimpan dengan aman. Setelah data diisi, pengguna dapat menyimpan konfigurasi yang kemudian langsung ditambahkan ke dalam daftar.

Seluruh alur proses, mulai dari penambahan, penghapusan, hingga pemilihan konfigurasi, telah melalui pengujian menggunakan metode black-box testing. Hasil pengujian menunjukkan bahwa seluruh komponen antarmuka dan fungsi berjalan dengan baik sesuai skenario, tanpa ditemukan kesalahan fungsi atau tampilan. Dengan adanya fitur ini, aplikasi dapat mendukung pembaruan model secara mandiri, pengujian model alternatif, serta fleksibilitas dalam pengelolaan deteksi berbasis AI di masa mendatang.

← Panduan Penggunaan

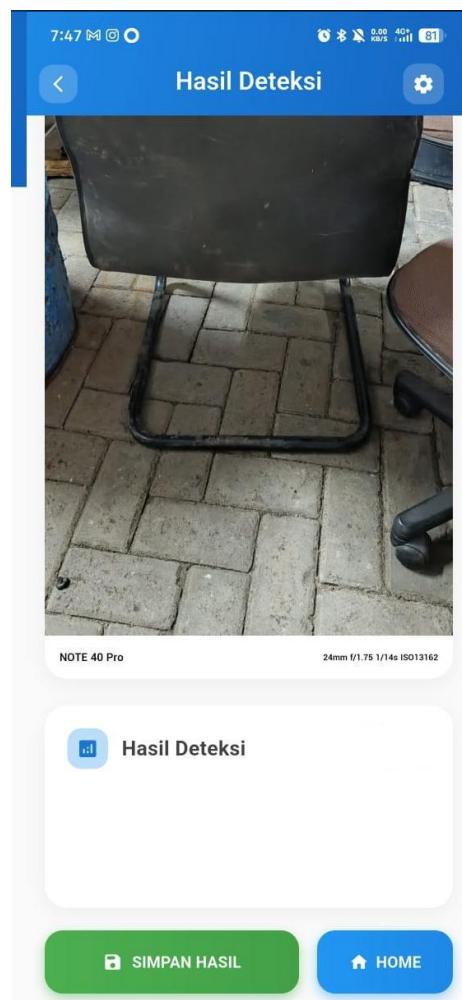


Gambar 4. 12 Tutorial Screen

Fitur Panduan Penggunaan pada aplikasi ini dirancang untuk membantu pengguna memahami prosedur pengambilan gambar yang benar serta interpretasi hasil deteksi kerusakan aset secara tepat. Panduan ini terdiri dari dua bagian utama. Bagian pertama menjelaskan cara pengambilan foto kursi dari empat sudut yang berbeda guna memastikan cakupan visual yang optimal. Pengguna diarahkan untuk memulai dengan mengambil gambar bagian depan kursi dengan memastikan pencahayaan cukup dan seluruh struktur kursi tampak jelas. Selanjutnya, pengguna diminta mengambil gambar dari sisi kiri atau kanan kursi, dengan posisi kamera yang sejajar terhadap objek dan latar belakang yang bersih agar bentuk samping

kursi dapat dikenali dengan baik. Panduan kemudian dilanjutkan dengan pengambilan gambar bagian belakang kursi, khususnya untuk mendeteksi potensi kerusakan pada sandaran, dan diakhiri dengan pengambilan gambar bagian bawah kursi secara tegak lurus dari atas, fokus pada kaki dan roda, untuk mengidentifikasi kerusakan struktural pada komponen bawah. Bagian kedua dari panduan menjelaskan interpretasi hasil deteksi berdasarkan nilai confidence yang dihasilkan oleh model deteksi. Sistem mengategorikan tingkat kepercayaan hasil deteksi dalam tiga label warna: hijau ($\geq 70\%$) yang berarti objek terdeteksi dengan tingkat kepercayaan tinggi, kuning (30–69%) yang menunjukkan kemungkinan deteksi namun perlu verifikasi ulang, dan merah ($<30\%$) yang mengindikasikan ketidakpastian dan disarankan untuk melakukan pengambilan gambar ulang. Untuk mendukung keberhasilan deteksi, panduan ini juga menyertakan sejumlah tips tambahan seperti memastikan pencahayaan cukup terang, menghindari bayangan pada objek, membersihkan lensa kamera sebelum memotret, serta menjaga jarak dan sudut pandang kamera agar objek terlihat dengan jelas. Seluruh alur dan tampilan pada fitur panduan ini telah diuji menggunakan pendekatan black-box testing dan berfungsi sebagaimana mestinya tanpa mengalami kendala. Kehadiran panduan ini tidak hanya mempermudah proses penggunaan aplikasi oleh pengguna umum, tetapi juga berperan penting dalam menjaga kualitas data masukan sehingga meningkatkan akurasi model dalam mendeteksi kerusakan aset.

4.6.2 Pengujian Halaman Riwayat



Gambar 4. 13 Layar Hasil

Halaman Hasil Deteksi menampilkan proses dan hasil analisis otomatis terhadap gambar yang dikirimkan oleh pengguna. Pada tahap awal, sistem akan menampilkan status pemrosesan dengan indikator “Memproses gambar...”, menunjukkan bahwa gambar sedang dikirim dan dianalisis oleh model deteksi yang di-host melalui API eksternal. Setelah proses selesai, hasil deteksi ditampilkan dalam dua bagian utama: visualisasi pada gambar dan daftar hasil deteksi dalam format teks. Pada bagian atas, sistem menampilkan gambar asli yang telah dianotasi dengan *bounding box* berwarna, dilengkapi label kelas objek dan tingkat kepercayaan (*confidence score*), misalnya "Roda Rusak 78.1%". Visualisasi ini

membantu pengguna secara langsung melihat letak dan jenis kerusakan pada aset.

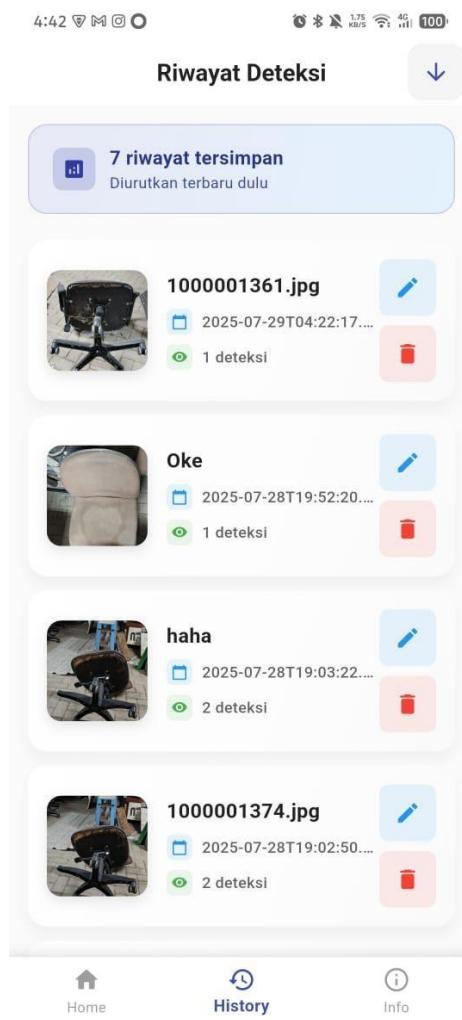
Di bagian bawah, terdapat daftar hasil deteksi dalam bentuk teks yang tersusun rapi, menampilkan setiap jenis kerusakan yang berhasil teridentifikasi beserta persentase tingkat kepercayaannya. Setiap entri dilengkapi dengan ikon validasi dan indikator warna sesuai dengan kategori confidence yang telah dijelaskan sebelumnya dalam panduan interpretasi (hijau untuk $\geq 70\%$). Sebagai contoh, pada gambar ditampilkan dua hasil deteksi: "Mengelupas" dengan tingkat keyakinan 94.6%, dan "Roda Rusak" dengan 78.1%, keduanya masuk dalam kategori terdeteksi dengan tingkat kepercayaan tinggi.

Di bagian bawah layar, terdapat dua tombol aksi utama: "Simpan Hasil" yang memungkinkan pengguna menyimpan hasil deteksi ke dalam riwayat lokal, serta "Home" untuk kembali ke halaman utama aplikasi. Seluruh fungsi dan tampilan pada halaman ini telah diuji melalui pendekatan black-box testing dan dinyatakan berfungsi dengan baik, baik dari sisi alur kerja, pemrosesan data, hingga responsivitas UI. Halaman ini menjadi elemen kunci dalam aplikasi karena memberikan umpan balik instan dan informatif kepada pengguna terkait kondisi aset yang dianalisis.

Gambar memperlihatkan salah satu skenario penting yang telah diantisipasi dalam pengembangan aplikasi, yaitu kondisi tanpa deteksi objek (*no detection*). Setelah pengguna mengambil atau memilih gambar kursi dan sistem menyelesaikan proses analisis visual, halaman Hasil Deteksi akan tetap menampilkan gambar asli meskipun tidak ditemukan kerusakan apapun oleh model deteksi. Hal ini ditandai dengan tidak adanya anotasi atau *bounding box* pada gambar, serta bagian daftar deteksi yang kosong. Meskipun demikian, aplikasi tetap menyediakan tombol "Simpan Hasil" yang dapat digunakan untuk menyimpan data gambar tersebut ke dalam riwayat pengguna. Fitur ini dirancang secara sengaja untuk mendukung dokumentasi penuh terhadap seluruh proses inspeksi aset, termasuk ketika tidak terdapat kerusakan yang teridentifikasi. Hal ini sangat relevan dalam konteks audit, pelacakan aset berkala, atau pencatatan kondisi visual terkini dari suatu aset.

Implementasi logika ini juga telah melalui pengujian menggunakan metode black-box testing, yang memastikan bahwa sistem tetap berjalan stabil, tidak mengalami kegagalan logika atau error, dan berhasil menyimpan data meskipun

hasil deteksi kosong. Pendekatan ini mempertegas bahwa aplikasi tidak hanya berfungsi sebagai alat deteksi otomatis, tetapi juga sebagai sistem pencatatan dan monitoring kondisi aset secara menyeluruh, baik pada kasus kerusakan maupun tidak adanya temuan.



Gambar 4. 14 Layar Riwayat

Halaman Riwayat Deteksi merupakan salah satu fitur penting dalam aplikasi yang berfungsi sebagai pusat dokumentasi hasil analisis gambar yang telah dilakukan oleh pengguna. Setiap kali pengguna menyimpan hasil deteksi, entri baru akan secara otomatis ditambahkan ke daftar riwayat, lengkap dengan informasi nama file, waktu penyimpanan, jumlah objek yang terdeteksi, serta tampilan pratinjau gambar mini. Data riwayat ini ditampilkan dalam urutan waktu terbaru terlebih dahulu secara default, dan dapat diubah melalui kontrol pengurutan di

bagian atas layar. Pengguna memiliki fleksibilitas dalam mengelola setiap entri riwayat. Dengan menekan ikon pensil, pengguna dapat mengubah nama file menjadi nama kustom yang lebih mudah dikenali. Proses pengeditan nama dilakukan melalui dialog input yang intuitif dan dilengkapi pembatasan karakter untuk menjaga konsistensi data. Selain itu, terdapat pula tombol hapus berwarna merah untuk menghapus entri yang tidak lagi dibutuhkan. Tindakan penghapusan dilindungi oleh dialog konfirmasi yang secara eksplisit memperingatkan pengguna bahwa tindakan tersebut bersifat permanen dan tidak dapat dibatalkan.

Fitur ini telah diuji menggunakan pendekatan black-box testing dan terbukti stabil dalam menangani berbagai skenario interaksi, termasuk pengeditan nama, penghapusan data, serta perubahan urutan tampilan. Dengan tersedianya fitur riwayat deteksi, aplikasi mendukung kebutuhan dokumentasi, pelacakan, dan audit visual secara berkelanjutan, menjadikannya alat bantu yang andal dalam proses manajemen aset kampus atau organisasi.

4.6.3 Pengujian Halaman Informasi



Gambar 4. 12 Information

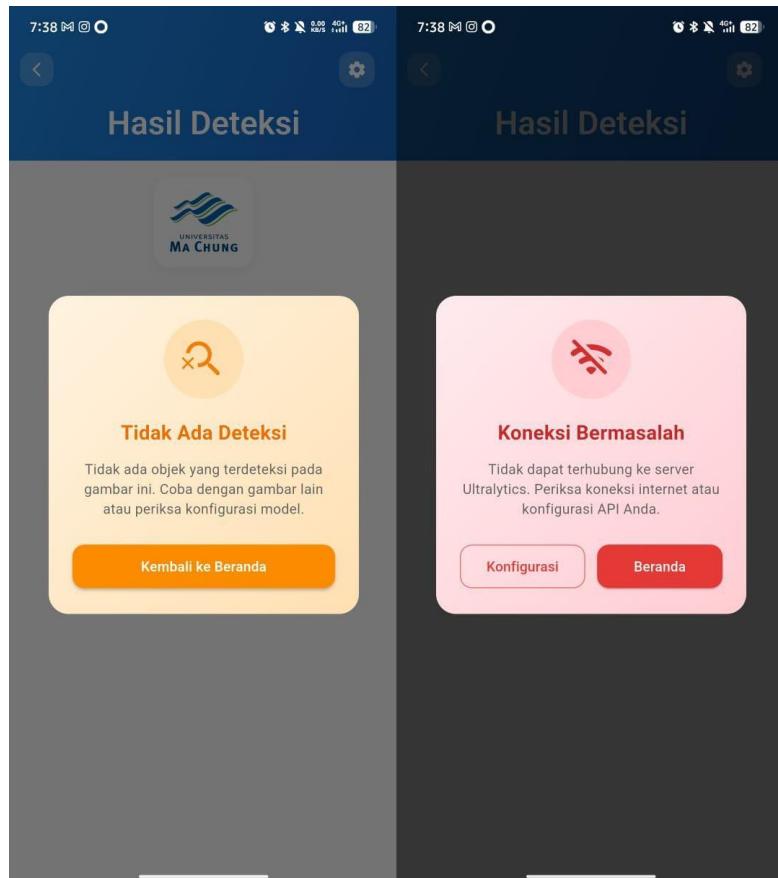
Halaman Informasi (*Info Screen*) merupakan salah satu bagian statis dari aplikasi yang berfungsi untuk menampilkan informasi institusi pengembang serta kontak resmi dari pihak yang bertanggung jawab atas pengelolaan sistem, yaitu Unit Sistem Informasi dan Pusat Data (SIPUSDA) Universitas Ma Chung.

Pengujian dilakukan untuk memastikan bahwa halaman ini: Dapat diakses melalui ikon navigasi bawah “Info”, menampilkan seluruh informasi dengan lengkap dan rapi, memuat logo institusi, nama pengembang, lokasi unit, serta informasi kontak (email dan nomor telepon).

Berdasarkan hasil pengujian, seluruh elemen pada halaman ini tampil dengan baik, termasuk posisi logo Universitas Ma Chung di bagian atas, diikuti dengan informasi nama pengembang (berisi nama mahasiswa dan dosen pembimbing),

lokasi unit, serta email dan nomor telepon resmi SIPUSDA. Antarmuka halaman ini menggunakan kombinasi warna latar lembut dan teks dengan kontras tinggi, sehingga mudah dibaca oleh pengguna.

4.6.4 Pengujian Error Handling



Gambar 4. 13 *Error Handling*

Aplikasi ini telah dilengkapi dengan sistem penanganan error (*error handling*) yang dirancang untuk memberikan umpan balik yang jelas dan informatif kepada pengguna apabila terjadi kegagalan selama proses deteksi. Berdasarkan gambar di atas, terdapat dua jenis skenario error yang telah diantisipasi dan ditampilkan melalui antarmuka pengguna.

Pertama, pada kasus "Tidak Ada Deteksi", sistem menampilkan pesan peringatan berwarna oranye apabila tidak ditemukan objek kerusakan pada gambar yang dikirimkan. Hal ini dapat terjadi jika gambar yang digunakan tidak mengandung objek relevan sesuai pelatihan model, pencahayaan kurang, atau sudut pengambilan gambar tidak sesuai. Aplikasi memberikan saran kepada pengguna

untuk mencoba dengan gambar lain atau memeriksa kembali konfigurasi model yang digunakan. Pesan ini ditampilkan secara elegan dengan tombol tindakan langsung untuk kembali ke halaman beranda. Kedua, pada skenario "Koneksi Bermasalah", sistem menampilkan pesan kesalahan berwarna merah apabila aplikasi gagal melakukan koneksi ke server Ultralytics. Hal ini bisa disebabkan oleh koneksi internet yang tidak stabil, atau kesalahan pada pengaturan API key dan model ID. Dalam kondisi ini, pengguna diberikan dua opsi tindakan yang jelas, yaitu membuka halaman konfigurasi untuk memperbaiki pengaturan, atau kembali ke beranda. Pendekatan ini tidak hanya menjaga kejelasan navigasi, tetapi juga memperkecil risiko frustrasi pengguna karena error yang tidak dijelaskan.

Seluruh skenario penanganan error ini telah diuji menggunakan metode black-box testing dan dinyatakan berfungsi sesuai dengan harapan. Dengan demikian, aplikasi memiliki ketahanan yang baik terhadap kondisi gagal deteksi maupun gangguan koneksi, serta tetap menjaga pengalaman pengguna secara optimal melalui antarmuka yang intuitif dan informatif.

4.7 Evaluasi Black Box Testing

Pengujian ini dilakukan dengan metode Black-Box Testing, yaitu pengujian yang berfokus pada fungsionalitas aplikasi berdasarkan spesifikasi dan skenario penggunaan, tanpa melihat kode internal aplikasi. Setiap fitur utama pada aplikasi diuji dengan memberikan input atau tindakan tertentu dan mengamati keluaran (output) yang dihasilkan. Apabila output yang muncul sesuai dengan yang diharapkan pada skenario tersebut, maka fitur dinyatakan berhasil memenuhi fungsinya. Sebaliknya, jika output tidak sesuai dengan yang diharapkan, berarti terdapat cacat (bug) yang perlu diperbaiki. Pada subbab ini dilakukan evaluasi Black-Box Testing untuk sejumlah fitur utama aplikasi, antara lain fitur Splash Screen, Kamera, Galeri, Konfigurasi API, Tutorial, Hasil Deteksi, Riwayat, dan Error Handling (penanganan kesalahan). Tabel 4.7 di bawah ini menyajikan skenario pengujian untuk tiap fitur beserta hasil pengujian yang diperoleh.

Tabel 4. 2 Hasil Pengujian Black-Box

No Fitur	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian (Aktual)	Status
1 Splash Screen	Membuka aplikasi untuk pertama kali.	Splash screen tampil dan dalam beberapa detik beralih ke halaman utama.	Splash screen muncul dan setelah 3 detik aplikasi beralih ke halaman Home.	Berhasil
2 Kamera	Pengguna menekan tombol kamera untuk mengambil foto.	Aplikasi membuka kamera, pengguna dapat mengambil foto, dan foto yang diambil muncul di aplikasi untuk diproses.	Saat tombol kamera ditekan, kamera terbuka, foto berhasil diambil, dan gambar hasil foto tampil di aplikasi untuk proses deteksi.	Berhasil
3 Galeri	Pengguna memilih gambar dari galeri perangkat.	Aplikasi membuka menu galeri, pengguna dapat memilih foto, dan foto terpilih muncul di aplikasi untuk diproses.	Saat tombol galeri ditekan, galeri terbuka, foto berhasil dipilih, dan gambar tersebut tampil di aplikasi untuk proses deteksi.	Berhasil
4 Konfigurasi API	Pengguna memasukkan URL endpoint API baru dan menyimpan konfigurasi.	URL API tersimpan dalam aplikasi dan digunakan untuk pemanggilan layanan (misalnya untuk deteksi atau sinkronisasi data).	Pengguna berhasil mengatur URL endpoint API; aplikasi menyimpan konfigurasi tersebut dan menggunakan URL baru untuk memuat data/model sesuai pengaturan.	Berhasil
5 Tutorial	Pengguna baru membuka aplikasi (pertama kali) atau memilih menu tutorial.	Aplikasi menampilkan layar tutorial (panduan penggunaan) secara berurutan yang dapat dilihat pengguna.	Pada pertama kali aplikasi dijalankan, tutorial pengenalan fitur ditampilkan secara berurutan. Tutorial juga dapat diakses kembali melalui menu dan tampil sesuai desain.	Berhasil
6 Hasil Deteksi	Setelah mengambil atau memilih gambar, sistem melakukan proses deteksi menggunakan model CNN dan menampilkan hasilnya.	Aplikasi menampilkan hasil deteksi (misalnya klasifikasi aset atau informasi yang teridentifikasi) sesuai gambar input, termasuk detail yang relevan.	Setelah pengguna memberikan gambar (foto kamera/galeri), sistem memproses gambar dengan model CNN dan menampilkan hasil deteksi aset beserta informasi klasifikasinya di layar	Berhasil

Tabel 4. 1 Hasil Pengujian Black-Box (Lanjutan)

No Fitur	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian (Aktual)	Status
7 Riwayat	Pengguna membuka halaman riwayat (history) Riwayat setelah melakukan beberapa kali deteksi.	Aplikasi menampilkan daftar yang berisi hasil-hasil deteksi sebelumnya lengkap dengan informasi waktu.	Halaman Riwayat berhasil menampilkan list riwayat hasil deteksi beserta timestamp setiap deteksi, sesuai urutan terbaru ke terlama.	Berhasil
8 Error Handling	Menguji skenario kesalahan, misalnya memutus koneksi internet lalu mencoba memuat data atau melakukan deteksi.	Aplikasi menampilkan pesan error yang sesuai (misalnya "Tidak ada koneksi internet") dan tidak terjadi <i>crash</i> .	Saat koneksi internet dimatikan dan pengguna mencoba fitur deteksi, muncul notifikasi error "Tidak ada koneksi internet". Aplikasi tetap berjalan stabil tanpa macet.	Berhasil

Berdasarkan hasil pengujian pada Tabel di atas, seluruh fitur utama aplikasi berfungsi sesuai dengan skenario yang telah ditentukan. Setiap fitur menghasilkan output yang sesuai dengan harapan dan tidak ditemukan bug atau kegagalan fungsional selama pengujian. Sebagai contoh, fitur *Kamera* dan *Galeri* berhasil mengakses perangkat (kamera dan penyimpanan) dan menampilkan gambar untuk dideteksi, fitur *Riwayat* merekam seluruh hasil deteksi terdahulu, dan mekanisme *error handling* mampu mengatasi kondisi tanpa koneksi internet dengan menampilkan pesan yang tepat. Dengan demikian, dapat dikatakan bahwa dari sisi fungsionalitas, aplikasi telah berjalan dengan baik dan sesuai spesifikasi. Hasil evaluasi *Black-Box Testing* ini menunjukkan bahwa aplikasi yang dikembangkan telah memenuhi kebutuhan fungsional yang diharapkan dalam mendukung manajemen aset.

Setelah memahami teori pengujian performa, tahap selanjutnya adalah melakukan pengujian kinerja pada prototipe aplikasi Pendukung Manajemen Aset berbasis Flutter yang telah dikembangkan. Pengujian kinerja dilakukan dengan mengukur beberapa indikator utama, yaitu waktu startup aplikasi, pemakaian memori, dan responsivitas UI. Pengukuran dilakukan menggunakan perangkat Android kelas menengah (RAM 6 GB, OS Android 12) untuk mensimulasikan

lingkungan operasional khas pengguna. Metode pengujian meliputi mencatat timestamp saat aplikasi diluncurkan hingga tampilan home muncul sepenuhnya (untuk waktu startup), memantau penggunaan RAM aplikasi melalui Android Studio Profiler setelah aplikasi stabil di halaman utama, serta mengukur frame rate antarmuka menggunakan alat profiling bawaan Flutter (*Flutter Performance Monitor*) selama interaksi normal dengan aplikasi.

Tabel 4. 2 Hasil Pengujian *Performance*

No Uji	Parameter	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian (Aktual)	Status
1	Waktu Startup	Mengukur waktu dari aplikasi dibuka hingga halaman utama tampil.	Waktu startup \leq 5 detik	Rata-rata 3,2 detik pada perangkat uji (Xiaomi Redmi Note 12 Pro 4G).	Berhasil
2	Penggunaan Memori	Memantau penggunaan memori saat aplikasi idle di halaman utama.	Penggunaan memori $<$ 200 MB pada perangkat kelas menengah.	Rata-rata 148 MB saat idle.	Berhasil
3	Responsivitas UI	Menguji kelancaran navigasi antar halaman dan <i>scrolling</i> .	Navigasi lancar tanpa <i>lag</i> atau <i>frame drop</i> signifikan.	FPS rata-rata 58–60 pada perangkat uji.	Berhasil
4	Waktu Respons API Deteksi	Mengukur waktu dari gambar dikirim hingga hasil deteksi muncul.	Waktu respons \leq 5 detik pada koneksi internet stabil.	Rata-rata 4,1 detik pada jaringan Wi-Fi 30 Mbps.	Berhasil
5	Penggunaan CPU	Memantau penggunaan CPU saat proses deteksi gambar.	Penggunaan CPU $<$ 50% pada perangkat kelas menengah.	Rata-rata 27% saat deteksi berlangsung.	Berhasil
6	Penggunaan Baterai	Mengukur konsumsi daya selama 30 menit penggunaan baterai aplikasi.	Konsumsi daya wajar ($<5\%$ penurunan baterai/30 menit).	Penurunan baterai 3% setelah 30 menit simulasi penggunaan.	Berhasil
7	Stabilitas Aplikasi	Menjalankan aplikasi selama 2 jam dengan penggunaan berulang fitur kamera dan galeri.	Tidak terjadi <i>crash</i> atau <i>force close</i> .	Aplikasi berjalan stabil tanpa <i>crash</i> .	Berhasil

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Proyek Laporan Praktik Kerja Lapangan yang berjudul "Pengembangan Aplikasi Pencatatan Kerusakan Aset Universitas Ma Chung Berbasis Flutter" ini telah dilaksanakan dengan baik. Berdasarkan proses implementasi dan pengujian yang dilakukan, maka diperoleh beberapa kesimpulan sebagai berikut:

1. Prototipe Aplikasi Mobile Tercipta sesuai Kebutuhan – Penelitian ini berhasil menghasilkan sebuah prototipe aplikasi mobile pendukung manajemen aset berbasis Flutter yang memungkinkan petugas lapangan mencatat data dan kondisi aset secara langsung di tempat. Aplikasi mampu mengambil atau memuat foto aset, mengintegrasikan deteksi kerusakan otomatis, serta menyimpan hasilnya, sehingga memenuhi kebutuhan digitalisasi pencatatan aset di lingkungan Universitas Ma Chung sebagaimana ditargetkan pada tujuan pertama.
2. Antarmuka Pengguna Sederhana dan User-Friendly – Antarmuka (UI) aplikasi dirancang dan diimplementasikan secara sederhana, intuitif, dan ramah pengguna, sesuai dengan tujuan kedua penelitian. Seluruh elemen UI utama (tombol navigasi, ikon, tata letak) telah diuji coba oleh pengguna non-teknis dan terbukti mudah dipahami. Halaman utama menyajikan menu jelas untuk setiap fitur, panduan penggunaan (tutorial) disertakan untuk membantu pengguna mengambil foto dengan benar, dan skema warna/ikon pada hasil deteksi memudahkan interpretasi. Dengan demikian, aspek usability aplikasi ini sudah sesuai dengan harapan, mendukung operasional petugas tanpa memerlukan pelatihan khusus.
3. Fungsionalitas Teruji dan Sesuai Spesifikasi – Seluruh fitur utama aplikasi telah dijalankan melalui pengujian (terutama black-box testing) dan dievaluasi memenuhi spesifikasi serta kebutuhan pengguna akhir, menjawab tujuan ketiga penelitian. Setiap modul, mulai dari pengambilan foto, pengiriman ke API deteksi, penampilan hasil, konfigurasi model, hingga penyimpanan riwayat, telah menunjukkan output yang sesuai

dengan yang diharapkan pada skenario uji. Hasil uji coba memperlihatkan aplikasi berfungsi stabil tanpa error pada skenario normal, dan performanya tergolong baik (waktu respon cepat, UI lancar, dan penggunaan sumber daya efisien) untuk penggunaan di perangkat Android kelas menengah. Dengan terpenuhinya aspek fungsional dan performa tersebut, dapat disimpulkan bahwa prototipe aplikasi ini layak digunakan sebagai landasan pengembangan lebih lanjut, sekaligus berhasil menjawab permasalahan manajemen aset kampus yang diangkat dalam penelitian ini.

5.2 Saran

Meskipun aplikasi telah berhasil memenuhi tujuan fungsional dan menunjukkan performa yang stabil selama pengujian, terdapat beberapa aspek yang perlu dipertimbangkan untuk pengembangan lebih lanjut agar aplikasi ini dapat benar-benar optimal digunakan dalam lingkungan operasional yang sesungguhnya. Salah satu saran utama adalah perlunya integrasi model CNN hasil pelatihan final menggunakan dataset internal Universitas Ma Chung. Saat ini, aplikasi masih menggunakan API eksternal yang berperan sebagai *placeholder* untuk kebutuhan deteksi visual. Dengan mengintegrasikan model CNN lokal yang telah dilatih secara khusus terhadap citra aset kampus, maka akurasi sistem dalam mengenali jenis kerusakan (misalnya patah, jamur, roda rusak, dan mengelupas) dapat ditingkatkan secara signifikan. Penggunaan model lokal juga akan memungkinkan pengembangan fitur lanjutan seperti deteksi offline atau pemrosesan batch.

Selain itu, disarankan agar aplikasi menjalani tahap pengujian lanjutan yang melibatkan pengguna akhir, dalam hal ini petugas Bagian Aset Universitas Ma Chung. Metode yang dapat digunakan adalah *User Acceptance Testing* (UAT) atau evaluasi kebergunaan (usability testing), agar dapat diperoleh umpan balik nyata dari pengguna terkait kemudahan penggunaan aplikasi, alur kerja, kebutuhan fitur tambahan, maupun potensi kendala yang muncul saat digunakan di lapangan. Melalui masukan tersebut, pengembang dapat melakukan iterasi desain yang lebih tepat sasaran dan menyesuaikan aplikasi dengan kondisi kerja sebenarnya.

Saran berikutnya adalah melakukan penyempurnaan pada dokumentasi teknis, baik dalam bentuk dokumentasi kode, struktur proyek, maupun dokumentasi penggunaan aplikasi. Dokumentasi yang lengkap akan memudahkan tim

pengembang berikutnya, baik dari kalangan mahasiswa maupun staf SIPUSDA, untuk memahami struktur sistem, memperbaiki bug, atau menambahkan fitur baru. Disarankan pula agar sistem penyimpanan data lokal diperluas untuk mendukung ekspor data atau integrasi dengan sistem pelaporan aset yang dimiliki universitas. Langkah ini akan meningkatkan nilai guna aplikasi dalam proses pelaporan formal dan pemantauan aset jangka panjang.

Dengan mempertimbangkan saran-saran tersebut, aplikasi ini diharapkan dapat terus berkembang menjadi sistem pendukung manajemen aset yang tidak hanya fungsional dan efisien, tetapi juga adaptif terhadap kebutuhan organisasi dan siap digunakan secara berkelanjutan dalam skala institusional

DAFTAR PUSTAKA

1. Ahmad, R., & Maulana, A. (2020). Perancangan Sistem Informasi Manajemen Aset Berbasis Android. *Jurnal Teknologi Informasi dan Komputer*, 6(1), 45–53.
2. Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*. <https://doi.org/10.48550/arXiv.2004.10934>
3. Dart.dev. (2023). *Dart Language Overview*. <https://dart.dev>
4. Effendy, M. Y., Nurninawati, E., & Setiyawan, A. A. (2022). Design and Build a Web-Based Asset Management Information System at PT Thamrin Telekomunikasi Network. *Aptisi Transactions on Technopreneurship (ATT)*, 4(1), 48–58. <https://doi.org/10.34306/att.v4i1.233>
5. Flutter. (2022). *Flutter Documentation: Build Apps for Any Screen*.
6. Google Developers. (2023). *Android Studio: The Official IDE for Android*. <https://developer.android.com/studio>
7. Gu, J. X., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., et al. (2018). Recent Advances in Convolutional Neural Networks. *Pattern Recognition*, 77, 354–377. <https://doi.org/10.1016/j.patcog.2017.10.013>
8. Kholis, M., & Huda, S. (2019). Aplikasi Manajemen Aset Multiplatform pada Universitas Yudharta. *Jurnal Sistem Informasi*, 14(2), 81–88.
9. Lewis, J. R. (2018). The System Usability Scale: Past, Present, and Future. *International Journal of Human–Computer Interaction*, 34(7), 577–590. <https://doi.org/10.1080/10447318.2018.1455307>
10. Nawrocki, P., Wrona, K., Marczak, M., & Śnieżyński, B. (2021). A Comparison of Native and Cross-Platform Frameworks for Mobile Applications. *Computer*, 54(3), 18–27. <https://doi.org/10.1109/MC.2020.2983893>
11. Nielsen, J. (1994). *Usability Engineering*. San Francisco: Morgan Kaufmann.

12. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.
13. Riyanto, Y. (2019). Sistem Informasi Manajemen Aset Berbasis Web pada Universitas Pamulang. *Jurnal Teknologi dan Sistem Informasi*, 7(2), 102–110.
14. Safaat, N. (2012). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
15. Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.
16. Sqlite.org. (2022). *SQLite: Self-Contained, High-Reliability, Embedded, Full-Featured SQL Database Engine*. <https://www.sqlite.org>