

PENGEMBANGAN APLIKASI NOTIFIKASI MASA STUDI MAHASISWA UIN AR RANIRY BERBASIS ANDROID

Muhammad Akram¹, Sarini Vita Dewi²

^{1,2}Pendidikan Teknologi Informasi, Tarbiyah dan Keguruan, Universitas Islam Negeri Ar-Raniry
Jl. Syekh Abdul Rauf No.1, Kopelma Darussalam, Banda Aceh 23111, Aceh, Indonesia.
e-mail: *1220212038@student.ar.raniry.ac.id, 2vita.sarini@ar-raniry.ac.id

Abstrak

Universitas Islam Negeri Ar-Raniry di Banda Aceh menetapkan waktu maksimum studi untuk program sarjana adalah tujuh tahun. Namun, komunikasi mengenai durasi studi dan pencapaian SKS masih dilakukan secara tradisional dan tidak aktif, sehingga dapat meningkatkan risiko Drop Out (DO). Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi notifikasi untuk masa studi yang berbasis Android dengan arsitektur full-stack yang terintegrasi, serta mengevaluasi efektivitasnya. Inovasi dari penelitian ini adalah penerapan sistem notifikasi yang proaktif dan sepenuhnya terintegrasi dalam konteks akademis di perguruan tinggi keagamaan Islam negeri. Metode yang digunakan adalah Research and Development (R&D) dengan model Waterfall, di mana aplikasi dikembangkan menggunakan Flutter untuk front-end, FastAPI untuk back-end, dan PostgreSQL sebagai basis data. Pengujian dilakukan dengan tiga pendekatan: black-box testing (28 skenario), pengujian usability dengan System Usability Scale (SUS) terhadap 20 mahasiswa, serta pengujian performa. Hasil penelitian menunjukkan bahwa semua fitur aplikasi berfungsi dengan baik, di mana 28 skenario dalam black-box testing berhasil dilaksanakan. Pengujian SUS menghasilkan rata-rata skor sebesar 73,0 (kategori Baik) dengan 70% responden memberikan umpan balik positif. Sementara itu, evaluasi User Experience (UX) memperoleh rata-rata 4,43 (Sangat Baik). Hasil dari pengujian performa menunjukkan waktu respon API rata-rata 187 ms, waktu muat aplikasi berkisar antara 1,2 hingga 2,3 detik, dan tingkat keberhasilan notifikasi mencapai 96,7%. Sistem ini terbukti efektif dalam meningkatkan kesadaran mahasiswa mengenai sisa masa studi dan memiliki potensi untuk mengurangi risiko DO.

Kata kunci: Aplikasi Notifikasi, Masa Studi Mahasiswa, Push Notification, Full-Stack Development, Android

Abstract

Ar-Raniry State Islamic University in Banda Aceh has set a maximum study period of seven years for its undergraduate programs. However, communication regarding study duration and credit hour completion is still conducted in a traditional and passive manner, which may increase the risk of student dropout (DO). This study aims to develop an Android-based study period notification application with an integrated full-stack architecture and to evaluate its effectiveness. The innovation of this study lies in the implementation of a proactive and fully integrated notification system within the academic context of a state Islamic religious university. The method used is Research and Development (R&D) with the Waterfall model, where the application was developed using Flutter for the front-end, FastAPI for the back-end, and PostgreSQL as the database. Testing was conducted using three approaches: black-box testing (28 scenarios), usability testing using the System Usability Scale (SUS) on 20 students, and performance testing. The results showed that all application features functioned properly, with all 28 scenarios in the black-box testing successfully executed. The SUS testing yielded an average score of 73.0 (Good category), with 70% of respondents providing positive feedback. Meanwhile, the User Experience (UX) evaluation received an average score of 4.43 (Very Good). The results of the performance testing showed an average API response time of 187 ms, and the application load time was.

Keywords: Notification Application, Student Study Period, Push Notification, Full-Stack Development, Android

I. PENDAHULUAN

Universitas Islam Negeri (UIN) Ar-Raniry di Banda Aceh adalah salah satu institusi pendidikan terkemuka di Indonesia yang termasuk dalam kelompok Perguruan Tinggi Keagamaan Islam Negeri (PTKIN). Institusi ini bertanggung jawab untuk melahirkan lulusan yang tidak hanya memiliki kecerdasan akademis dan spiritual yang tinggi, tetapi juga dapat menyelesaikan studi mereka dalam waktu yang ditentukan. Masa studi mahasiswa merupakan indikator penting dalam mengukur efektivitas proses pendidikan dan efisiensi manajemen akademik sebuah universitas. Berdasarkan Peraturan Menteri Agama dan statuta universitas, masa studi program sarjana (S1) umumnya dibatasi hingga 7 (tujuh) tahun. Batas waktu ini menjadi acuan kritis bagi mahasiswa dan pihak universitas.

Namun, dalam praktiknya, UIN Ar-Raniry masih menghadapi tantangan signifikan dalam mengkomunikasikan dan memantau batas masa studi ini. Banyak mahasiswa yang tidak menyadari bahwa mereka telah mendekati atau bahkan melampaui batas waktu studi yang ditetapkan. Ketidaksadaran ini seringkali berakibat fatal, seperti Droup Out (DO) secara akademik, yang merugikan mahasiswa secara psikologis dan finansial, serta berdampak pada statistik kinerja universitas[1].

Dampak kumulatif dari sistem yang tidak proaktif ini adalah tingginya risiko mahasiswa telat lulus atau bahkan DO (Drop Out). Kelebihan masa studi tidak hanya menyebabkan stres bagi mahasiswa, tetapi juga memboroskan sumber daya pendidikan dan secara signifikan menurunkan Indeks Kinerja Utama (IKU) universitas, khususnya indikator tingkat kelulusan tepat waktu.

Di sisi lain, perkembangan teknologi informasi dan komunikasi (TIK) telah mencapai titik yang memungkinkan solusi yang lebih cerdas dan efisien. Telepon pintar (smartphone) berbasis Android telah menjadi perangkat yang hampir tidak terpisahkan dari kehidupan mahasiswa sehari-hari. Fitur-fitur seperti notifikasi push memungkinkan pengiriman informasi langsung ke genggaman pengguna secara instan dan personal. Pendekatan mobile-first ini menjawab tantangan komunikasi tradisional dengan menciptakan saluran yang proaktif, langsung, dan memiliki tingkat keterbukaan (open rate) yang sangat tinggi[2],[3].

Keefektifan metode push notification dalam lingkungan akademik telah dibuktikan melalui perancangan aplikasi pengingat jadwal perkuliahan yang membantu mahasiswa mendapatkan informasi secara real-time[4]. Selain itu, notifikasi akan dikirimkan secara strategis pada momen- momen kritis dalam siklus akademik, seperti menjelang akhir semester atau pada saat pengisian nilai oleh dosen. Waktu-waktu tersebut dipilih karena merupakan periode di mana mahasiswa perlu lebih sadar akan pencapaian SKS dan sisa masa studi mereka. Dengan demikian, aplikasi tidak hanya bersifat reaktif tetapi juga proaktif dalam mengingatkan mahasiswa pada fase-fase penting perkuliahan, seperti pengambilan SKS berdasarkan IPK dan sisa SKS untuk sidang.

Meskipun telah terdapat beberapa penelitian terkait aplikasi notifikasi akademik, masih terdapat kesenjangan yang perlu diisi. Penelitian sebelumnya hanya merancang UI/UX tanpa implementasi sistem fungsional[2]. Selain itu, terdapat penelitian yang berfokus pada push notification namun belum terintegrasi dengan database akademik dinamis[4].

Penelitian lain juga masih terbatas pada perancangan antarmuka[5]. Dengan demikian, belum ada penelitian yang mengimplementasikan aplikasi notifikasi masa studi dengan arsitektur full-stack terintegrasi serta evaluasi komprehensif melalui pengujian black-box, usability (SUS), user experience, dan performa sistem.

Oleh karena itu, diperlukan terobosan berupa implementasi aplikasi notifikasi masa studi berbasis Android dengan pendekatan pengembangan full-stack terintegrasi.

II. TINJAUAN PUSTAKA

2.1 Flutter

Flutter adalah SDK open-source dari Google untuk membangun aplikasi native multi-platform (Android, iOS, web, desktop) dengan satu basis kode menggunakan bahasa Dart[6]. Framework ini mengadopsi arsitektur berbasis widget yang memungkinkan kustomisasi UI tinggi. Keunggulan Flutter terletak pada kompilasi ahead-of-time (AOT) untuk performa native dan just-in-time (JIT) dengan fitur Hot Reload yang mempercepat proses pengembangan. Pendekatan cross-platform ini semakin populer karena efisiensi pengembangan untuk berbagai platform[7].

2.2 FastAPI

FastAPI adalah framework web Python modern yang dirancang untuk membangun API dengan performa tinggi, setara Node.js dan Go. Memanfaatkan type hints Python, FastAPI menawarkan pengembangan cepat, dokumentasi otomatis, dan validasi data secara real-time, menjadikannya pilihan ideal untuk pengembangan back-end aplikasi modern.

2.3 PostgreSQL

PostgreSQL adalah sistem manajemen basis data objek-relasional

open-source yang mengutamakan integritas data dan kepatuhan terhadap standar SQL. Fitur unggulannya seperti Information Schema memudahkan pengelolaan metadata basis data[8]. Studi menunjukkan PostgreSQL mampu menangani volume data besar dengan performa tinggi, di mana algoritma terintegrasi langsung dalam DBMS menghasilkan waktu respons lebih cepat dibanding alat analisis eksternal[9]. Hal ini menjadikannya salah satu RDBMS open-source paling kuat dan fleksibel.

2.4 Insomnia

Insomnia adalah aplikasi cross-platform yang berfungsi sebagai REST client dan lingkungan pengujian API. Dengan antarmuka grafis intuitif, Insomnia memungkinkan pengembang membuat, mengelola, dan menguji permintaan HTTP tanpa menulis kode manual, serta mendukung pengujian endpoint API secara efisien di berbagai sistem operasi (Windows, macOS, Linux).

2.5 Docker Engine

Docker Engine adalah teknologi kontainerisasi yang mengemas aplikasi beserta seluruh dependensinya ke dalam container terisolasi. Berbeda dengan virtualisasi tradisional, container berbagi kernel sistem operasi host sehingga menghasilkan overhead lebih rendah, startup time lebih cepat, dan efisiensi sumber daya lebih baik. Dalam arsitektur full-stack modern, Docker menjadi standar untuk menjamin konsistensi lingkungan dari pengembangan hingga produksi[10].

2.6 Penelitian Terdahulu

Penelitian sebelumnya yang relevan mencakup:

1. Annisa Zahra & Sarini Vita Dewi (2025). Desain UI/UX untuk Aplikasi Notifikasi Masa Studi Mahasiswa UIN Ar-Raniry

Menggunakan Android. Studi ini menghasilkan desain UI/UX dengan skor SUS 81,5 (sangat baik), yang menunjukkan bahwa antarmuka yang intuitif sangat penting dalam meningkatkan kesadaran mahasiswa tentang tenggat waktu studi[2].

2. Rina, J., Siska, A., & Irawatia (2021). Pengembangan Aplikasi Pemberitahuan Jadwal Kuliah Melalui Metode Push Notification Berbasis Mobile. Temuan studi menunjukkan bahwa pemberitahuan push sangat efektif dalam membantu mahasiswa untuk tetap ingat pada jadwal kuliah dan meningkatkan disiplin akademik[4].
3. Daruyani, S., Purbiyanti, O. D., Irawaty, I., & Putra, A. W. (2024). Desain UI/UX untuk Aplikasi Pengingat Sholat (Ar-Rayyan) dengan Menggunakan Metode Design Thinking dan Figma. Penelitian ini memperoleh tingkat kepuasan pengguna 90% dengan desain yang mudah digunakan, interaktif, dan ramah pengguna[5].

III. METODOLOGI PENELITIAN

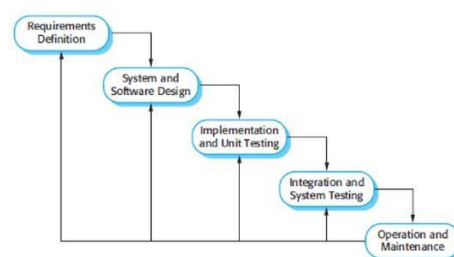
3.1 Metode Pengembangan Sistem

Penelitian ini adalah tipe penelitian Research and Development (R&D) yang bertujuan untuk menciptakan aplikasi pengingat masa studi berbasis Android dengan arsitektur full-stack yang terintegrasi.[11]. Metode yang diterapkan adalah model Waterfall yang mencakup lima tahap: analisis kebutuhan, desain sistem, penerapan, pengujian, dan pemeliharaan.[12].

Metode Waterfall dipilih dalam penelitian ini dengan sejumlah faktor utama yang dipertimbangkan:

1. Kebutuhan sistem telah teridentifikasi secara jelas dan stabil melalui studi kebijakan dan wawancara awal dengan pihak akademik[13].
2. Skala proyek yang terdefinisi dengan baik dengan dua aktor utama (mahasiswa dan admin) membuat pendekatan linier lebih efisien[14].
3. Kebutuhan integrasi dengan database akademik yang memiliki struktur tetap memerlukan perancangan yang matang di awal untuk mengurangi risiko kesalahan[12].
4. Keterbatasan tim pengembang menjadikan model Waterfall dengan kerangka kerja terstruktur sebagai pilihan yang tepat[15].
5. Target pengguna yang homogen (mahasiswa UIN Ar-Raniry) tidak memerlukan umpan balik berulang seperti pada pengembangan dengan pendekatan iteratif[16].

Model Waterfall yang diterapkan dalam studi ini mencakup lima tahap utama, yaitu: pengidentifikasian kebutuhan (Requirements definition), perancangan sistem dan perangkat lunak (System and Software Design), proses pengembangan serta pengujian unit (Implementation and Unit Testing), pengujian sistem secara menyeluruh (Integration and System Testing), dan pemeliharaan yang disertai dengan evaluasi antarmuka pengguna dan pengalaman pengguna (Operation and Maintenance).



Gambar 1. Model Waterfall

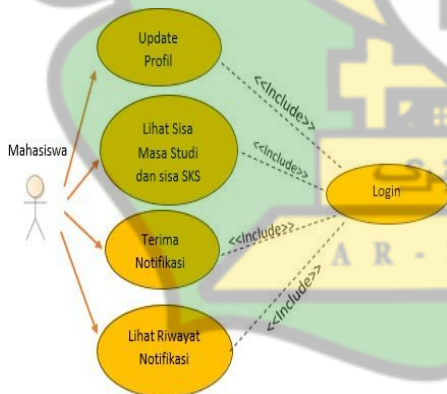
Sumber : repository.upi.edu

1. *Requirements definition* (Analisis kebutuhan)

Pada tahap analisis kebutuhan, peneliti melakukan studi kebijakan dengan menganalisis Peraturan Menteri Agama dan statuta universitas untuk memastikan dasar hukum perhitungan masa studi, serta wawancara dengan pihak akademik untuk memahami penyebab tingginya risiko Drop Out akibat ketidaksadaran batas masa studi tujuh tahun. Peneliti juga mencermati mekanisme komunikasi akademik yang selama ini berjalan konvensional dan tidak proaktif, serta melakukan studi literatur terhadap teknologi yang akan digunakan (Flutter, FastAPI, PostgreSQL) untuk merumuskan kebutuhan sistem secara lengkap.

Berdasarkan hasil analisis, sistem memiliki 2 aktor utama:

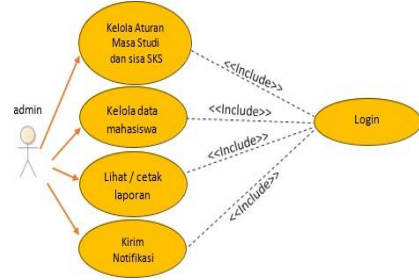
- **Mahasiswa:** mengakses informasi sisa masa studi, melihat SKS, menerima notifikasi, mengelola profil.



Gambar 2. Use Case Mahasiswa

- **Admin:** pengelola sistem yang bertugas mengelola data mahasiswa, mengatur aturan masa studi dan batas SKS, serta mengirimkan notifikasi peringatan kepada mahasiswa. Admin juga dapat memantau dan mencetak laporan mahasiswa yang mendekati batas masa studi untuk

kebutuhan evaluasi akademik dan pelaporan IKU universitas.

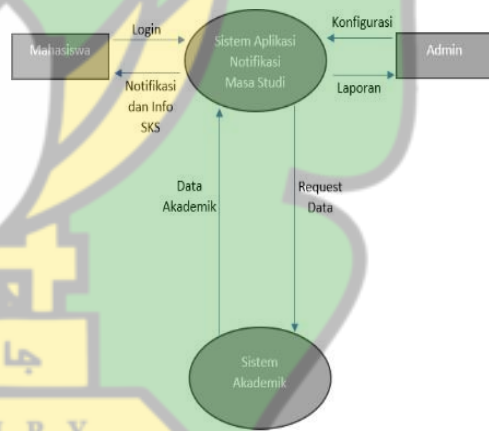


Gambar 3. Use Case Admin

2. *System and Software Design* (Perancangan sistem)

Pada fase ini dirancang arsitektur full-stack yang menghubungkan Flutter (front-end), FastAPI (back-end), dan PostgreSQL (database), serta skema database dan spesifikasi API.

Di bawah ini adalah DFD Level 0 yang menunjukkan pergerakan data antara entitas dalam suatu sistem.



Gambar 4. Data Flow Diagram Level 0

Merujuk pada Gambar 4, ada tiga entitas utama yang berhubungan dengan sistem, yaitu Mahasiswa, Admin, dan Data Akademik. Mahasiswa mendapatkan pemberitahuan dan informasi mengenai SKS melalui sistem serta melakukan proses login. Admin melakukan konfigurasi sistem dan menerima laporan dari sistem. Sistem juga berinteraksi dengan database akademik untuk melakukan request data terkait informasi mahasiswa.

3. *Implementation and Unit Testing*
(Mengembangkan kode front-end dan back-end)

Pada tahap implementasi, peneliti merealisasikan rancangan ke dalam bentuk kode program dengan mengembangkan aplikasi front-end menggunakan Flutter yang terintegrasi dengan back-end, membangun API back-end menggunakan FastAPI yang memuat logika perhitungan sisa masa studi, serta melakukan setup database PostgreSQL sesuai skema yang telah dirancang.

4. *Integration and System Testing*
(Pengujian sistem)

Selama proses evaluasi sistem, peneliti menerapkan tiga metode pengujian guna memastikan kualitas aplikasi secara menyeluruh. Pengujian sistem dilaksanakan dengan tiga pendekatan. Pertama, black-box testing yang melibatkan 28 skenario untuk mencakup semua fitur bagi mahasiswa dan admin dengan menggunakan Insomnia.[17]. Kedua, evaluasi kegunaan dilakukan dengan menggunakan System Usability Scale (SUS) pada 20 mahasiswa, yang melibatkan 10 pernyataan dengan skala 1-5 (lihat Tabel 1), ditambah dengan 5 pertanyaan untuk evaluasi pengalaman pengguna (UX) (lihat Tabel 2).[18]. Ketiga, pengujian performa meliputi response time API, waktu muat aplikasi, keberhasilan notifikasi, dan penggunaan sumber daya[19].

5. *Operation and Maintenance*
(Pengoperasian dan pemeliharaan serta evaluasi UI/UX dari kebutuhan pengguna)

Tahap akhir meliputi deployment menggunakan Docker Engine, pembuatan file APK, penyusunan dokumentasi, dan identifikasi area pengembangan lanjutan.

3.2 Metode Pengujian Sistem

Pengujian sistem merupakan fase kritis untuk memastikan aplikasi beroperasi sesuai spesifikasi, handal, dan memenuhi kebutuhan pengguna. Dalam penelitian ini, pengujian dilakukan melalui tiga pendekatan saling melengkapi: fungsionalitas, kebergunaan, dan performa.

Pertama, pengujian fungsionalitas dilaksanakan dengan cara black-box testing yang menekankan pada analisis hasil sistem tanpa melihat komponen internal kode, yang efektif dalam mengidentifikasi kesalahan di fungsi, antarmuka, serta akses data.[20]. Sebanyak 28 skenario uji disusun berdasarkan kebutuhan fungsional sistem, mencakup seluruh fitur mahasiswa (12 skenario) dan admin (16 skenario). Pengujian dilakukan dengan mengirimkan permintaan HTTP ke setiap endpoint API menggunakan Insomnia untuk memverifikasi kesesuaian respons, serta menguji mekanisme notifikasi dalam berbagai kondisi aplikasi (aktif, latar belakang, tertutup).

Kedua, pengujian kebergunaan dilakukan untuk menilai seberapa mudah aplikasi digunakan dari sudut pandang pengguna. Metode yang diterapkan adalah System Usability Scale (SUS), yaitu kuesioner baku yang terdiri dari 10 pernyataan dengan skala Likert 1-5 (1 = Sangat Tidak Setuju, 5 = Sangat Setuju).[18]. Kuesioner SUS telah terbukti efektif dan konsisten dalam menilai kegunaan suatu sistem dengan cepat dan tepat. Responden dalam pengujian ini adalah 20 mahasiswa aktif UIN Ar-Raniry yang dipilih secara purposive sampling dengan kriteria: (a) merupakan mahasiswa aktif program sarjana, (b)

telah mengikuti demonstrasi aplikasi di Laboratorium Rekayasa Perangkat Lunak, dan (c) bersedia berpartisipasi secara sukarela. Seluruh pernyataan kuesioner SUS disajikan pada Tabel 1.

Tabel 1. Kuesioner System Usability Scale (SUS)

No	Pernyataan	Skala
1	Saya berencana untuk memakai aplikasi ini secara teratur.	1-5
2	Aplikasi ini terasa sulit untuk dipakai.	1-5
3	Saya merasa bahwa penggunaan aplikasi ini sangat sederhana.	1-5
4	Saya memerlukan dukungan dari pihak lain atau panduan teknis untuk memanfaatkan aplikasi ini.	1-5
5	Saya merasa bahwa fungsi-fungsi dalam aplikasi ini beroperasi dengan baik dan sesuai dengan yang diharapkan.	1-5
6	Saya merasakan banyak ketidaksesuaian dalam aplikasi ini.	1-5
7	Saya percaya orang lain dapat dengan mudah belajar menggunakan aplikasi ini.	1-5
8	Aplikasi ini membuat saya merasa kebingungan.	1-5
9	Saya tidak merasakan adanya kendala saat menggunakan aplikasi ini.	1-5
10	Saya harus terbiasa terlebih dahulu sebelum memanfaatkan aplikasi ini.	1-5

Selain pengujian SUS, dilakukan pula evaluasi aspek User Experience (UX) menggunakan 5 pernyataan tambahan yang dikembangkan

berdasarkan dimensi UX, meliputi daya tarik visual, kerapian tata letak, kemudahan navigasi, kejelasan informasi sisa masa studi, dan kualitas notifikasi (Tabel 2). Evaluasi ini bertujuan untuk memahami persepsi mahasiswa terhadap kualitas interaksi dengan aplikasi secara lebih mendalam.

Tabel 2. Kuesioner Evaluasi User Experience (UX)

No	Aspek	Pernyataan	Skala
1	Daya tarik visual	Tampilan aplikasi ini menarik secara visual	1-5
2	Tata letak	Tata letak (layout) aplikasi rapi dan teratur	1-5
3	Navigasi	Menu dan tombol mudah ditemukan dan digunakan	1-5
4	Kejelasan Informasi	Informasi sisa masa studi dan status akademik (SKS, IPK) ditampilkan dengan jelas	1-5
5	Notifikasi	Notifikasi peringatan masa studi ditampilkan dengan cara yang menarik perhatian	1-5

Ketiga, Pengujian performa mengukur response time API, waktu muat aplikasi pada perangkat RAM 3GB dan 6GB, keberhasilan notifikasi (30 percobaan), serta penggunaan CPU dan memori[19].

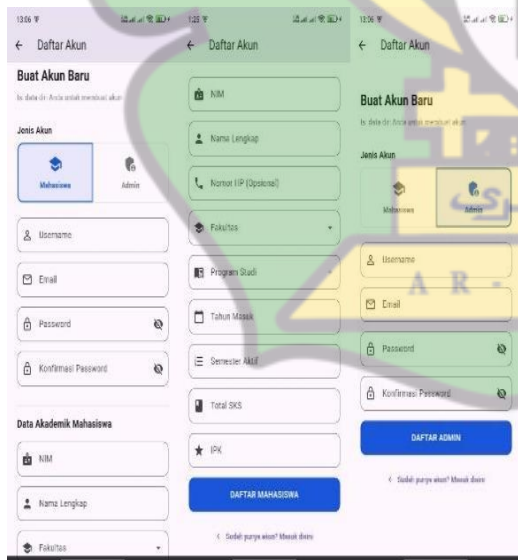
IV. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan aplikasi notifikasi masa studi berbasis Android dengan arsitektur full-stack terintegrasi menggunakan Flutter, FastAPI, dan PostgreSQL. Aplikasi memiliki dua peran pengguna yaitu mahasiswa yang dapat memantau sisa masa studi dan notifikasi, serta admin yang mengelola data dan mengirim peringatan.

Aplikasi memiliki halaman register, login, dashboard mahasiswa, notifikasi, profil, admin dashboard, daftar mahasiswa, form notifikasi, manajemen notifikasi, laporan, dan menu cetak laporan.

4.1 Tampilan Antarmuka Aplikasi

1. Halaman Register Mahasiswa dan Admin



Gambar 5. Register Akun Mahasiswa dan Admin

Pada gambar 5 register akun ini, mahasiswa dan admin dapat mendaftarkan akunya untuk mulai menggunakan aplikasi.

2. Halaman Login



Gambar 6. Halaman Login

Pada gambar 6 halaman login, halaman ini dibuat khusus untuk login akun mahasiswa dan akun admin.

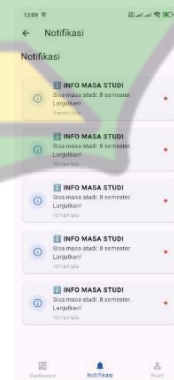
3. Dashboard Mahasiswa



Gambar 7. Dashboard Mahasiswa

Pada Gambar 7, dashboard mahasiswa menampilkan progress studi (semester, SKS, IPK), status masa studi (Normal, Peringatan, Kritis, Masa Studi Habis), serta bilah notifikasi terbaru.

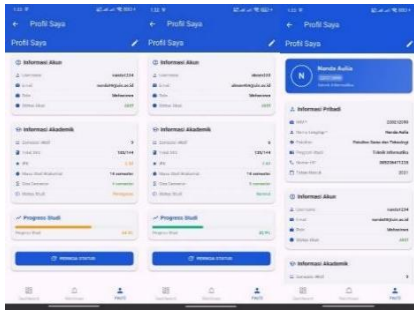
4. Halaman Notifikasi Mahasiswa



Gambar 8. Notifikasi Mahasiswa

Pada gambar 8 notifikasi mahasiswa, mahasiswa dapat melihat riwayat notifikasi yang telah masuk dan bisa dibaca detailnya dan ditandai telah dibaca.

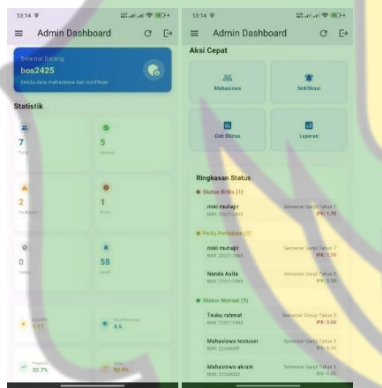
5. Halaman Profil Mahasiswa



Gambar 9. Halaman Profil Mahasiswa

Pada Gambar 9 Halaman Profil Mahasiswa ini, mahasiswa bisa melihat data pribadi dan juga informasi akademik. Serta mahasiswa bisa mengedit untuk informasi pribadi.

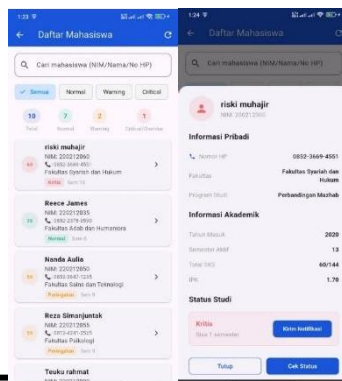
6. Admin Dashboard



Gambar 10. Admin Dashboard

Pada gambar 10 admin dashboard, halaman ini admin dapat melihat dashboard yang berupa statistik keseluruhan data progress mahasiswa yang ada di aplikasi, kemudian ada tombol tombol aksi cepat yang akan berpindah kehalaman lain seperti Mahasiswa, Notifikasi, Laporan. Serta ada ringkasan status dari setiap mahasiswa.

7. Daftar Mahasiswa dan Detail Info



Gambar 11. Daftar Mahasiswa dan Detail Info

Pada gambar 11 daftar mahasiswa dan detail info, halaman ini menampilkan seluruh data mahasiswa untuk admin dan juga admin bisa melakukan pencarian serta admin bisa juga melihat detail dari salah satu mahasiswa kemudian admin bisa mengirimkan notifikasi ke salah satu mahasiswa

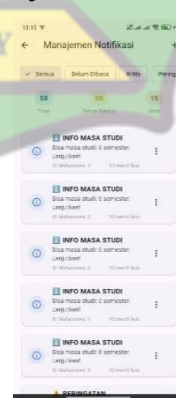
8. Form Kirim Notifikasi



Gambar 12. Form Kirim Notifikasi

Pada Gambar 12 form kirim notifikasi, form ini digunakan oleh admin untuk mengirim notifikasi ke mahasiswa.

9. Manajemen Notifikasi



Gambar 13. Manajemen Notifikasi

Pada gambar 13 manajemen notifikasi, halaman ini dibuat agar admin dapat mudah dalam mengelola notifikasi.

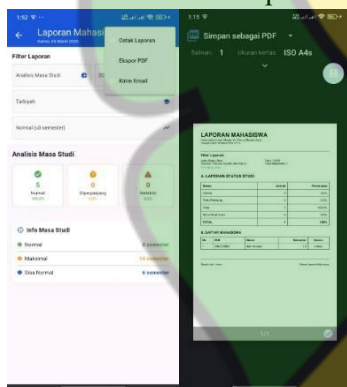
10. Laporan Mahasiswa



Gambar 14. Laporan Mahasiswa

Pada gambar 14 laporan mahasiswa yang ditampilkan, admin dapat memanfaatkan halaman ini untuk memonitor laporan yang dibuat oleh mahasiswa serta melakukan penyaringan terhadap laporan-laporan tersebut.

11. Menu Cetak Laporan



Gambar 15. Menu Cetak Laporan

Pada gambar 15 menu cetak laporan, admin dapat mencetak laporan mahasiswa, mengekspor PDF dan kirim Email.

4.2 Hasil Pengujian Black-Box Testing

Tabel 3. Hasil Pengujian Black Box

No	Fitur Aplikasi	Skenario Pengujian	Status
1	Register Mahasiswa	Mahasiswa mendaftar akunnya di	Berhasil

		halaman daftar akun	
2	Register Admin	Admin mendaftar akunnya di halaman daftar akun	Berhasil
3	Login Mahasiswa	Mahasiswa memasukkan username dan password valid	Berhasil
4	Login Gagal	Mahasiswa memasukkan email dan password salah	Berhasil
5	Dashboard Mahasiswa	Mahasiswa membuka halaman dashboard	Berhasil
6	Periksa Status Studi	Mahasiswa melakukan periksa status studi	Berhasil
7	Riwayat Notifikasi	Mahasiswa membuka dan melihat halaman riwayat notifikasi	Berhasil
8	Detail Notifikasi	Mahasiswa klik salah satu notifikasi	berhasil
9	Hapus Notifikasi	Mahasiswa menghapus salah satu notifikasi	berhasil
10	Profil Mahasiswa	Mahasiswa membuka halaman profil	Berhasil
11	Edit Profil	Mahasiswa mengubah informasi pribadi	Berhasil
12	Logout Mahasiswa	Mahasiswa melakukan proses logout akun	Berhasil
13	Login Admin	Admin memasukkan	Berhasil

		username dan password valid	
14	Login Gagal	Admin memasukkan username dan password salah	Berhasil
15	Dashboard Admin	Admin membuka halaman dashboard	Berhasil
16	Daftar Mahasiswa	Admin membuka halaman daftar mahasiswa	Berhasil
17	Tombol pencarian	Admin mencari mahasiswa berdasarkan NIM>Nama	Berhasil
18	Detail Info Mahasiswa	Admin klik salah satu mahasiswa yang ada di dalam daftar mahasiswa	Berhasil
19	Kirim Notifikasi	Admin mengirimkan notifikasi ke mahasiswa	Berhasil
20	Form Notifikasi	Admin membuat form notifikasi dan kirim	Berhasil
21	Manajemen Notifikasi	Admin membuka halaman manajemen notifikasi	Berhasil
22	Tambah Notifikasi	Admin menambahkan notifikasi baru	Berhasil
23	Edit Notifikasi	Admin mengubah isi notifikasi	Berhasil
24	Hapus Notifikasi	Admin menghapus data notifikasi	Berhasil

25	Cek Status Mahasiswa	Admin memeriksa seluruh status mahasiswa	Berhasil
26	Laporan Mahasiswa	Admin membuka halaman laporan mahasiswa	Berhasil
27	Filter Laporan	Admin menyaring laporan mahasiswa dan tampil bentuk grafik	Berhasil
28	Cetak Laporan	Admin mencetak laporan dalam bentuk pdf dan dibagikan	Berhasil

Berdasarkan Tabel 3, seluruh 28 skenario pengujian berhasil, menunjukkan semua fitur aplikasi berfungsi sesuai spesifikasi dengan integrasi komponen yang harmonis dan fitur notifikasi yang andal.

4.3 Hasil Pengujian Usability dengan System Usability Scale (SUS)

Usability test dengan memanfaatkan System Usability Scale (SUS) dilaksanakan pada 20 mahasiswa UIN Ar-Raniry yang telah menyaksikan demonstrasi aplikasi di Laboratorium Rekayasa Perangkat Lunak. Kuesioner berisi 10 pertanyaan dengan skala Likert 1-5.

Setelah mengumpulkan data, langkah selanjutnya adalah menghitung skor SUS untuk masing-masing responden. Metode perhitungan yang digunakan adalah: untuk pertanyaan dengan nomor ganjil (Q1, Q3, Q5, Q7, Q9) skor dihitung dengan rumus (jawaban - 1), sedangkan untuk pertanyaan bernomor genap (Q2, Q4, Q6, Q8, Q10) skor ditentukan dengan rumus (5 - jawaban). Jumlah semua skor kemudian dikalikan dengan 2,5 untuk memperoleh skor akhir dalam rentang 0-100. Hasil dari

perhitungan skor SUS untuk setiap responden ditampilkan di Tabel 5.

Tabel 4. Skor Hasil SUS

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah Nilai	Jumlah Nilai
4	0	4	1	3	3	2	3	3	0	23	57.5
4	4	4	4	4	4	4	4	4	4	40	100
4	3	4	3	4	1	4	3	2	2	30	75
2	1	3	1	4	2	3	1	2	1	20	50
3	4	4	2	3	4	4	4	4	0	32	80
3	3	3	2	4	3	3	3	3	2	29	72.5
4	2	4	0	3	1	3	1	3	1	22	55
4	4	4	3	4	4	4	4	4	1	36	90
4	3	4	0	4	3	3	4	3	0	28	70
4	4	3	2	2	3	3	3	4	0	28	70
3	4	4	0	3	3	2	3	4	3	29	72.5
3	3	3	2	3	3	3	3	2	2	27	67.5
4	3	3	2	4	3	3	4	4	2	32	80
4	4	4	3	4	4	4	4	4	3	38	95
3	3	4	3	4	3	2	4	4	2	32	80
3	0	4	2	3	4	1	4	3	1	25	62.5
4	2	4	2	4	2	4	4	3	2	31	77.5
2	3	3	4	4	3	0	3	3	2	27	67.5
3	3	3	0	3	2	3	4	3	2	26	65
1	2	2	1	4	3	3	2	2	0	20	50
Rata-rata											73

Berdasarkan Tabel 4, skor SUS tertinggi 100 dicapai oleh responden R2,

menunjukkan penilaian sangat positif terhadap kemudahan penggunaan aplikasi. Skor terendah 50 diperoleh responden R4 dan R20, yang masih berada dalam kategori cukup. Skor rata-rata SUS dari 20 orang responden adalah 73.

Kategori penilaian yang digunakan untuk menginterpretasikan skor SUS mengacu pada [18], dengan distribusi skor 20 responden disajikan pada Tabel 5.

Tabel 5. Distribusi Kategori Skor SUS

Kategori	Rentang Skor	Jumlah Responden	Persentase
Sangat Baik	>80.3	5	25%
Baik	68 - 80.3	9	45%
Cukup	51 - 68	5	25%
Buruk	<51	1	5%
Total		20	100%

Tabel 5 menunjukkan 70% responden berada dalam kategori Baik dan Sangat Baik, mengindikasikan tingkat usability yang baik.

Tabel 6. Grade SUS Skor

SUS score	Grade	Adjective Rating
74 – 100	A	Sangat Layak
63 – 73	B	Layak digunakan
51 – 62	C	Marginal
0 – 50	D	Tidak Layak Digunakan

Skor rata-rata SUS 73 berada pada kategori Baik (Grade B), menunjukkan aplikasi telah memenuhi standar kebergunaan yang memadai dan layak digunakan untuk membantu mahasiswa memantau masa studi mereka.

4.4 Hasil Evaluasi User Experience (UX)

Selain pengujian SUS, evaluasi UX dilakukan untuk mengetahui pandangan mahasiswa mengenai mutu interaksi dengan aplikasi, yang meliputi lima dimensi dengan skala Likert 1-5. Hasil dari evaluasi ini dapat dilihat secara menyeluruh pada Tabel 7.

Tabel 7. Hasil Evaluasi User Experience (UX)

No	Aspek UX	Rata-rata	Kategori
1	Daya tarik visual	4.55	Sangat Baik
2	Kerapian tata letak	4.35	Sangat Baik
3	Kemudahan navigasi	4.35	Sangat Baik
4	Kejelasan informasi sisa masa studi	4.75	Sangat Baik
5	Daya tarik pemberitahuan	4.15	Baik
Rata-rata keseluruhan		4.43	Sangat Baik

Seluruh aspek UX memperoleh nilai di atas 4,0 dengan kejelasan informasi tertinggi (4,75) dan kemenarikan notifikasi terendah (4,15). Rata-rata 4,43 (Sangat Baik) menunjukkan aplikasi memberikan pengalaman positif bagi mahasiswa.

4.5 Hasil Pengujian Performa Sistem

Pengujian performa mengukur responsivitas dan stabilitas sistem melalui parameter: response time API, waktu muat aplikasi pada perangkat berbeda, keberhasilan notifikasi, serta penggunaan CPU dan memori. Data hasil uji kinerja ditampilkan dalam Tabel 8.

Tabel 8. Hasil Pengujian Performa Sistem

Parameter	Hasil	Keterangan
Response time API	120-350 ms	Rata-rata 187 ms
Waktu muat aplikasi (perangkat entry-level)	2.3 detik	RAM 3GB
Waktu muat aplikasi (perangkat mid-range)	1.2 detik	RAM 6GB
Keberhasilan pengiriman notifikasi	96.7%	29 dari 30 pengujian
Penggunaan CPU	15-25%	Saat idle
Penggunaan memori	80-120 MB	Aplikasi berjalan

Tabel 8 menunjukkan response time API 187 ms, waktu muat 1,2-2,3 detik, keberhasilan notifikasi 96,7%, serta penggunaan CPU dan memori yang efisien.

V. KESIMPULAN

Dengan merujuk pada langkah-langkah penelitian dan pengembangan yang telah dilaksanakan, dapat disimpulkan hal berikut:

1. Penelitian ini berhasil mengembangkan aplikasi notifikasi masa studi berbasis Android dengan arsitektur full-stack terintegrasi (Flutter, FastAPI, PostgreSQL) yang memiliki dua peran pengguna: mahasiswa dan admin.
2. Pengujian black-box terhadap 28 skenario menunjukkan keberhasilan 100%, menandakan seluruh fitur berfungsi sesuai spesifikasi dengan integrasi komponen yang harmonis.
3. Evaluasi usability dengan SUS pada 20 mahasiswa memperoleh skor rata-rata 73 (kategori Baik) dengan 70% responden memberikan penilaian positif, menunjukkan tingkat kebergunaan yang memadai.

4. Evaluasi UX pada lima aspek menghasilkan rata-rata 4,43 (Sangat Baik), dengan aspek kejelasan informasi sisa masa studi mencapai skor tertinggi (4,75).
5. Pengujian performa menunjukkan response time API 187 ms, waktu muat aplikasi 1,2-2,3 detik, keberhasilan notifikasi 96,7%, serta penggunaan CPU dan memori yang efisien.
6. Kelebihan aplikasi terletak pada arsitektur terintegrasi dan sistem notifikasi proaktif, sementara kekurangannya pada aspek kemenarikan notifikasi (skor 4,15) dan belum terintegrasi dengan SIAKAD.
7. Pengembangan selanjutnya dapat difokuskan pada integrasi dengan SIAKAD, implementasi Firebase Cloud Messaging, serta pengembangan fitur prediksi kelulusan berbasis machine learning.

VI. SARAN

Saran untuk penelitian selanjutnya:

1. Mengintegrasikan aplikasi dengan Sistem Informasi Akademik (SIAKAD) UIN Ar-Raniry untuk pengujian menggunakan data mahasiswa sebenarnya.
2. Mengimplementasikan Firebase Cloud Messaging (FCM) agar notifikasi dapat dikirim secara otomatis dan efisien, termasuk saat aplikasi dalam keadaan background atau closed.
3. Melakukan load testing untuk memastikan kestabilan sistem, serta mengembangkan fitur prediksi kelulusan berbasis machine learning dan ekspansi aplikasi ke platform iOS.

VII. DAFTAR PUSTAKA

- [1] A. Yaghi and N. Alabed, "Factors Affecting University Dropout: Comparison of STEM and Public Affairs and Management Students," *International Journal of Public Administration*, vol. 49, no. 5, pp. 324–338, 2026, doi: 10.1080/01900692.2025.2476676.
- [2] A. Zahra and S. Vita Dewi, "Perancangan Ui/Ux Aplikasi Notifikasi Masa Studi Mahasiswa Uin Ar-Raniry Berbasis Android (Ui/Ux Design Of An Android-Based Study Duration Notification Application For Student Of Uin Ar-Raniry)," *Desember*, vol. 6, no. 2, pp. 202–211, 2025.
- [3] B. B. Mumcu and A. Çebi, "You have a notification: the role of push notifications in shaping students' engagement, self-regulation and academic procrastination," *International Journal of Educational Technology in Higher Education*, vol. 22, no. 1, p. 36, Dec. 2025, doi: 10.1186/s41239-025-00537-x.
- [4] R. Junita Basri and S. Anraeni, "Buletin Sistem Informasi dan Teknologi Islam Perancangan Aplikasi Pengingat Jadwal Perkuliahan Menggunakan Metode Push Notification Berbasis Mobile INFORMASI ARTIKEL ABSTRAK," vol. 2, no. 1, pp. 52–55, 2021.
- [5] S. Daruyani, O. D. Purbiyanti, I. Irawaty, and A. W. Putra, "Perancangan UI/UX Aplikasi Pengingat Shalat (Ar-Rayyan) Dengan Metode Design Thinking Menggunakan Figma," *Journal of Information System, Informatics and Computing*, vol. 8, no. 2, p. 264, Dec. 2024, doi: 10.52362/jisicom.v8i2.1669.
- [6] R. Azilah, F. Al Huda, and L. Fanani, "Jip (Jurnal Informatika Polinema) Pengembangan Aplikasi Berbagi Makanan (Food Sharing) Berbasis Mobile Menggunakan Framework Flutter," *Jurnal Informatika Polinema*, vol. 10, no. 4, pp. 453–462, 2024, doi: 10.33795/jip.v10i4.4967.
- [7] A. Burkutbayeva, Y. Daineko, and Z. Dushimova, "Cross-Platform Augmented Reality Mobile Application for Visualization of Genetic Facial Traits: System Design and Use Case,"

- in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 596–601. doi: 10.1016/j.procs.2025.07.225.
- [8] D. Aji Bayu Prasetyo and Y. Alfa Susetyo, “Implementasi Information Schema Database Pada Postgre SQL Untuk Pembuatan Tabel Informasi Dengan Menggunakan Python Di PT XYZ,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 3, pp. 1961–1972, 2022, doi: 10.35957/jatisi.v9i3.2221.
- [9] A. Makris, K. Tserpes, G. Spiliopoulos, D. Zissis, and D. Anagnostopoulos, “MongoDB Vs PostgreSQL: A comparative study on performance aspects,” *Geoinformatica*, vol. 25, no. 2, pp. 243–268, Apr. 2021, doi: 10.1007/s10707-020-00407-w.
- [10] T. Alyas, S. Ali, H. U. Khan, A. Samad, K. Alissa, and M. A. Saleem, “Container Performance and Vulnerability Management for Container Security Using Docker Engine,” *Security and Communication Networks*, vol. 2022, 2022, doi: 10.1155/2022/6819002.
- [11] Agus Rustamana, Khansa Hasna Sahl, Delia Ardianti, and Ahmad Hisyam Syauqi Solihin, “Penelitian dan Pengembangan (Research & Development) dalam Pendidikan,” *Jurnal Bima : Pusat Publikasi Ilmu Pendidikan bahasa dan Sastra*, vol. 2, no. 3, pp. 60–69, Jun. 2024, doi: 10.61132/bima.v2i3.1014.
- [12] G. B. M. T. Sitorus, R. A. G. Efendi, J. F. Napitupulu, H. Pranoto, and E. S. Hermawan, “Bridging the Gap: A Comparative Evaluation of Agile, Waterfall, and Hybrid Methodologies in Modern Software Projects,” in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 1259–1268. doi: 10.1016/j.procs.2025.09.067.
- [13] E. M. T. Sanmocte and J. A. Costales, “Exploring Effectiveness in Software Development: A Comparative Review of System Analysis and Design Methodologies,” *International Journal of Computer Theory and Engineering*, vol. 17, no. 1, pp. 36–37, 2025, doi: 10.7763/IJCTE.2025.V17.1367.
- [14] A. Mishra and Y. I. Alzoubi, “Structured software development versus agile software development: a comparative analysis,” *International Journal of System Assurance Engineering and Management*, vol. 14, no. 4, pp. 1504–1522, Aug. 2023, doi: 10.1007/s13198-023-01958-5.
- [15] D. Ly, M. Overeem, S. Brinkkemper, and F. Dalpiaz, “The Power of Words in Agile vs. Waterfall Development: Written Communication in Hybrid Software Teams,” *Journal of Systems and Software*, vol. 219, p. 112243, Jan. 2025, doi: 10.1016/j.jss.2024.112243.
- [16] H. Gunatilake, J. Grundy, R. Hoda, and I. Mueller, “The impact of human aspects on the interactions between software developers and end-users in software engineering: A systematic literature review,” *Inf. Softw. Technol.*, vol. 173, p. 107489, Sep. 2024, doi: 10.1016/j.infsof.2024.107489.
- [17] Yulherniwati, M. Gunawan, A. Febbrian Kasmar, R. Hadi, E. Asri, and Humaira, “Black-Box Testing pada Sistem Evaluasi Capaian Pembelajaran Berdasarkan Outcome Based Education,” *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 6, no. 2, pp. 202–207, Jun. 2025, doi: 10.62527/jitsi.6.2.279.
- [18] Q. Khan *et al.*, “Psychometric evaluation of the System Usability Scale in the context of a childrearing app co-designed for low- and middle-income countries,” *Digit. Health*, vol. 11, pp. 1–12, Jan. 2025, doi: 10.1177/20552076251335413.
- [19] A. Poth, O. Rrjolli, and A. Arcuri, “Technology adoption performance evaluation applied to testing industrial REST APIs,” *Automated Software Engineering*, vol. 32, no. 1, p. 5, Jun. 2025, doi: 10.1007/s10515-024-00477-2.
- [20] J. Shadiq, A. Safei, R. Wahyudin Ratu Loly, C. sitasi, L. Rwr, and P. Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing,

“Information Management For Educators And Professionals Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing,” *Information Management for Educators and Professionals*, vol. 5, no. 2, pp. 97–110, 2021.

