

## ANALISIS PERBANDINGAN ANTARA *FRAMEWORK FLUTTER* DENGAN *NATIVE* UNTUK PENGEMBANGAN APLIKASI *MOBILE ABSENSI FACE RECOGNITION*

Nur Farid Mufid NR<sup>1</sup>, Adi Rizaldy<sup>2</sup>

<sup>1,2</sup>Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Alauddin Makassar  
Email: <sup>1</sup>nurfarid776@gmail.com, <sup>2</sup>adhy.rizaldy@uin-alauddin.ac.id  
*\*Penulis Korespondensi*

### Abstrak

Penelitian ini membandingkan performa antara framework Flutter dan pengembangan native dalam pembuatan aplikasi absensi berbasis face recognition. Dua prototipe aplikasi dengan fitur yang sama dikembangkan menggunakan kedua pendekatan tersebut, diikuti dengan pengujian performa yang mencakup kecepatan request data, penggunaan CPU dan RAM, serta kecepatan kompilasi file .apk. Metodologi yang digunakan adalah pendekatan komparatif dengan metode Rapid Application Development (RAD). Data pengujian performa dianalisis untuk mengidentifikasi perbedaan antara kedua pendekatan tersebut. Hasil penelitian menunjukkan bahwa pengembangan aplikasi dengan pendekatan native memiliki performa yang lebih baik dalam segala aspek yang diuji, yakni kecepatan request data, penggunaan CPU dan RAM, serta kecepatan kompilasi file .apk. Implikasi penelitian ini memberikan panduan bagi pengembang aplikasi dalam memilih antara Flutter dan pengembangan native, terutama untuk aplikasi yang membutuhkan performa tinggi dan efisiensi penggunaan sumber daya. Penelitian ini juga berkontribusi pada literatur teknologi, memberikan pemahaman yang lebih mendalam tentang kelebihan dan kekurangan masing-masing framework. Selain itu, membuka peluang untuk studi lebih lanjut mengenai optimalisasi pengembangan aplikasi mobile.

**Kata kunci:** *Perbandingan framework, Rapid Application Development (RAD), Pengujian performa, Face recognition, Aplikasi absensi, Native development, Flutter, Penggunaan RAM, Penggunaan CPU, Kecepatan request data, Kompilasi file APK.*

### Abstract

*This study compares the performance between the Flutter framework and native development in creating a face recognition-based attendance application. Two application prototypes with identical features were developed using both approaches, followed by performance testing covering data request speed, CPU and RAM usage, and APK file compilation speed. The research methodology uses a comparative approach with Rapid Application Development (RAD) for system development. Performance test data were analyzed to identify the differences between the two approaches. The results show that native development provides better performance in all tested parameters, including data request speed, CPU and RAM usage, and APK file compilation speed. The implications of this research provide guidance for application developers in choosing between Flutter and native development, especially for applications requiring high performance and resource efficiency. This study also contributes to the technology literature by offering a deeper understanding of the strengths and weaknesses of each framework. Furthermore, it opens opportunities for further studies on mobile application development optimization.*

**Keywords:** *Framework comparison, Rapid Application Development (RAD), Performance testing, Face recognition, Attendance application, Native development, Flutter, RAM usage, CPU usage, Data request speed, APK file compilation.*

## 1. PENDAHULUAN

Dalam perjalanan menuju masa depan yang semakin modern, pemanfaatan teknologi informasi telah menjadi pilar utama dalam menghadapi tantangan zaman. Perkembangan teknologi ini mengubah cara kita berinteraksi dengan dunia, memungkinkan komunikasi instan dan akses cepat ke informasi melalui perangkat seperti smartphone. Aplikasi mobile kini menjadi bagian tak terpisahkan dari kehidupan, memainkan peran penting dalam

berbagai sektor, termasuk hiburan, produktivitas, dan bisnis. Untuk itu, pemilihan framework pengembangan aplikasi yang tepat menjadi langkah krusial dalam menciptakan aplikasi mobile yang efisien dan berkualitas. Flutter dan native adalah dua framework yang banyak digunakan untuk pengembangan aplikasi mobile. Flutter, sebagai framework cross-platform, memudahkan pengembangan aplikasi untuk berbagai platform seperti iOS dan Android dengan satu kode sumber. Di

sisi lain, framework native menawarkan kontrol lebih mendalam terhadap platform tertentu, seperti menggunakan Swift untuk iOS atau Java untuk Android. Pemilihan framework yang tepat bergantung pada berbagai faktor, termasuk kecepatan pengembangan, biaya, kebutuhan fitur, dan target audiens aplikasi.

Salah satu aplikasi mobile yang saat ini banyak digunakan adalah aplikasi absensi berbasis face recognition. Teknologi ini memanfaatkan pengenalan wajah untuk mencatat kehadiran dengan akurat, meningkatkan efisiensi dan mengurangi potensi kesalahan dalam proses absensi. Aplikasi semacam ini digunakan dalam berbagai konteks, dari dunia pendidikan hingga dunia kerja. Penelitian ini bertujuan untuk membandingkan dua framework utama dalam pengembangan aplikasi absensi berbasis face recognition, yaitu Flutter dan native, dengan fokus pada aspek-aspek penting seperti kinerja, kecepatan request data, kompilasi file APK, serta penggunaan prosesor dan memori. Diharapkan penelitian ini dapat memberikan panduan yang akurat bagi pengembang untuk memilih framework yang paling sesuai dengan kebutuhan pengembangan aplikasi absensi yang efisien dan andal.

Berdasarkan latar belakang tersebut, penulis terdorong untuk melakukan penelitian berjudul "Analisis Perbandingan Antara Framework Flutter dengan Native Untuk Pengembangan Aplikasi Mobile Absensi Face Recognition". Penelitian ini bertujuan untuk menganalisis perbandingan kedua framework dengan parameter performa untuk membantu pengembang memilih framework yang paling sesuai dalam menciptakan aplikasi mobile absensi berbasis face recognition yang efisien dan efektif.

Tujuan dari penelitian ini adalah untuk melakukan analisis perbandingan antara framework Flutter dan native pada pengembangan aplikasi absensi face recognition. Penelitian ini bertujuan untuk memberikan panduan yang jelas bagi pengembang dalam memilih framework yang sesuai dengan kebutuhan aplikasi mobile, khususnya dalam hal performa dan fleksibilitas. Dengan demikian, hasil penelitian ini diharapkan dapat membantu pengembang dalam memilih framework yang lebih efisien, andal, dan sesuai dengan tujuan pengembangan aplikasi mobile absensi berbasis face recognition.

Dalam penelitian ini, fokus utama akan diberikan pada perbandingan dua pendekatan utama dalam pengembangan aplikasi mobile absensi berbasis teknologi face recognition, yaitu framework Flutter dan pendekatan native. Penelitian ini akan memusatkan perhatian pada perbandingan kinerja kedua framework dalam beberapa aspek yang dianggap penting dalam pengembangan aplikasi mobile. Adapun parameter yang akan dianalisis dalam perbandingan antara framework Flutter dan native mencakup beberapa hal yang relevan, yaitu:

pertama, kecepatan request data, yang mengukur seberapa cepat aplikasi dapat melakukan komunikasi dan pengambilan data untuk mencatat kehadiran pengguna. Kedua, kecepatan kompilasi file APK, yang menilai seberapa cepat aplikasi dapat disiapkan dan dibangun menjadi file APK untuk platform Android. Ketiga, penggunaan prosesor, yang akan mengukur seberapa besar daya komputasi yang dibutuhkan aplikasi, terutama dalam proses pengenalan wajah yang membutuhkan pemrosesan gambar secara intensif. Keempat, penggunaan memori, yang akan melihat berapa banyak memori yang digunakan oleh aplikasi dalam menjalankan proses absensi dan face recognition. Dengan menganalisis parameter-parameter ini, penelitian bertujuan untuk memberikan gambaran yang jelas mengenai kelebihan dan kekurangan masing-masing framework dalam pengembangan aplikasi absensi berbasis teknologi face recognition.

Tinjauan pustaka menjadi hal yang penting dalam penelitian ini untuk memberikan gambaran tentang perkembangan teknologi dan framework yang digunakan dalam aplikasi mobile. Beberapa penelitian sebelumnya telah membandingkan Flutter dengan framework lain, seperti React Native. Salah satunya adalah penelitian yang dilakukan oleh Zakki (2021) yang membandingkan Flutter dan React Native dalam hal customisability, performance, modifiability, dan testability. Penelitian ini memiliki kesamaan dengan penelitian yang dilakukan oleh penulis, yang juga bertujuan untuk membandingkan framework Flutter dengan framework lain. Namun, perbedaannya terletak pada pemilihan framework yang dibandingkan, di mana penelitian ini fokus pada perbandingan antara Flutter dan framework native, sedangkan penelitian Zakki menggunakan Flutter dan React Native.

Selain itu, penelitian Muhammad (2023) juga relevan dengan topik yang dibahas dalam penelitian ini. Penelitian tersebut membahas aplikasi monitoring presensi dengan teknologi face recognition berbasis Android. Penelitian ini memiliki kesamaan dengan penelitian yang akan dilakukan oleh penulis dalam hal penggunaan teknologi face recognition, tetapi perbedaannya terletak pada pemilihan framework. Penelitian ini akan membandingkan Flutter dan native untuk mengetahui framework mana yang lebih tepat dalam pengembangan aplikasi absensi berbasis face recognition.

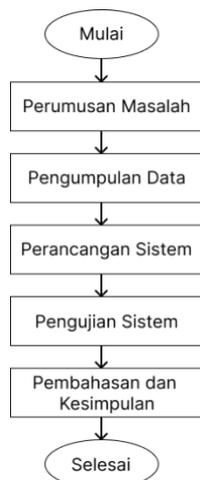
Tujuan dari penelitian ini adalah untuk melakukan analisis perbandingan antara framework Flutter dan native pada pengembangan aplikasi absensi face recognition. Penelitian ini bertujuan untuk memberikan panduan yang jelas bagi pengembang dalam memilih framework yang sesuai dengan kebutuhan aplikasi mobile, khususnya dalam hal performa dan fleksibilitas. Dengan demikian, hasil penelitian ini diharapkan dapat membantu pengembang dalam memilih framework yang lebih efisien, andal, dan sesuai dengan tujuan

pengembangan aplikasi mobile absensi berbasis face recognition.

## 2. METODELOGI

### 2.1. Prosedur Penelitian

Prosedur penelitian merupakan kerangka proses penelitian yang digunakan oleh penulis sebagai acuan dari proses penelitian yang dilaksanakan. Prosedur penelitian yang digunakan pada penelitian dapat dilihat pada Gambar 1.



Gambar 1. Prosedur Penelitian

### 2.2. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam proses pengumpulan data penelitian adalah sebagai berikut.

#### 1. Pengujian dan Simulasi

Metode pengumpulan data utama dalam penelitian ini adalah melalui pengujian dan simulasi kedua aplikasi, yakni yang dikembangkan dengan Flutter dan native. Pengujian ini fokus pada aspek performa dan fleksibilitas aplikasi. Dengan simulasi yang teliti, peneliti dapat memperoleh data akurat untuk mengevaluasi perbedaan antara kedua pendekatan dalam pengembangan aplikasi absensi berbasis face recognition, yang akan digunakan untuk analisis perbandingan.

#### 2. Dokumentasi Teknis

Dokumentasi teknis resmi dari Flutter serta platform pengembangan native (Android dan iOS) akan menjadi sumber data penting dalam penelitian ini. Dokumentasi tersebut mencakup panduan teknis, perangkat lunak, dan informasi terkait pengembangan aplikasi mobile. Data ini akan membantu peneliti memahami karakteristik, fitur, dan batasan masing-masing

pendekatan, mendukung analisis perbandingan penggunaan Flutter dan native dalam pengembangan aplikasi absensi berbasis face recognition.

#### 3. Studi Literatur

Peneliti akan menelusuri jurnal, artikel, buku, dan publikasi akademis terkait pengembangan aplikasi mobile, teknologi face recognition, serta perbandingan antara Flutter dan pengembangan native. Data yang diperoleh akan membangun kerangka teoritis, mendukung analisis perbandingan, dan memastikan pemahaman yang mendalam tentang topik penelitian.

### 2.3. Metode Pengembangan Sistem

Pengembangan sistem ini menerapkan metode *RAD (Rapid Application Development)* dalam proses pengembangannya. Terdapat 4 tahap dalamnya yang meliputi

#### 1. Perencanaan Kebutuhan

Tahap ini *User* dan *Analyst* mengadakan pertemuan untuk mengidentifikasi tujuan sistem dan kebutuhan informasi yang diperlukan untuk mencapai tujuan tersebut. Keterlibatan aktif dari kedua pihak pada tahap ini sangat penting.

#### 2. Design Sistem

Peneliti membuat rancangan Use Case Diagram, Class Diagram, Entity Relationship Diagram, dan Activity Diagram dari sistem yang akan dikembangkan.

#### 3. Pengembangan

Pada tahap ini, sistem yang disepakati dikembangkan menjadi versi beta hingga final, dengan mengintegrasikan komponen dan memperhatikan umpan balik pengguna. Jika aplikasi memenuhi kebutuhan, proses dilanjutkan ke tahap berikutnya; jika tidak, kembali ke tahap desain untuk perbaikan.

#### 4. Implementasi

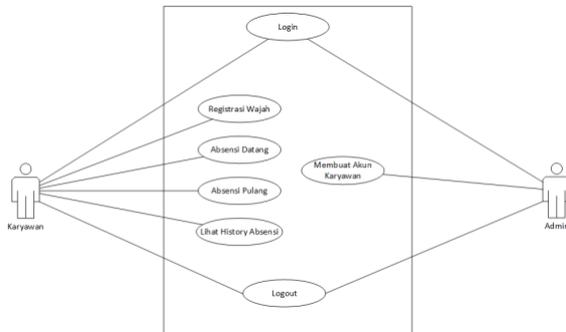
Pada tahap ini, desain program yang disetujui dikembangkan dan diuji untuk memastikan tidak ada kesalahan. Sebelum diterapkan program diuji.

### 2.4. Perancangan Sistem

Dalam penelitian ini, peneliti membuat rancangan Use Case Diagram, Class Diagram, Entity Relationship Diagram (ERD), dan Activity Diagram untuk menggambarkan sistem dari segi fungsionalitas dan alur kerja yang akan dikembangkan.

### 2.4.1. Use Case Diagram

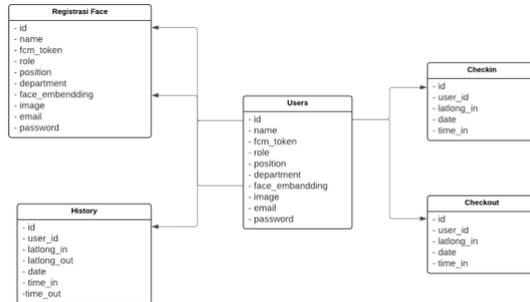
Dengan merancang Use Case Diagram, akan membantu dalam menggambarkan interaksi antara pengguna dan sistem secara visual, memudahkan pemahaman tentang fungsionalitas yang tersedia dalam sistem yang akan dikembangkan. Use Case Diagram dari sistem dapat dilihat pada Gambar 2.



Gambar 2. Use Case Diagram

### 2.4.2. Class Diagram

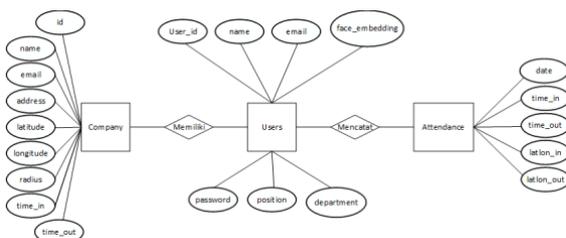
Dengan merancang Class Diagram, akan membantu dalam menggambarkan struktur statis dari sistem, termasuk kelas-kelas, atribut, metode, serta hubungan antar kelas yang ada dalam sistem yang akan dikembangkan. Class Diagram dari sistem dapat dilihat pada Gambar 3.



Gambar 3. Class Diagram

### 2.4.3. Entity Relationship Diagram

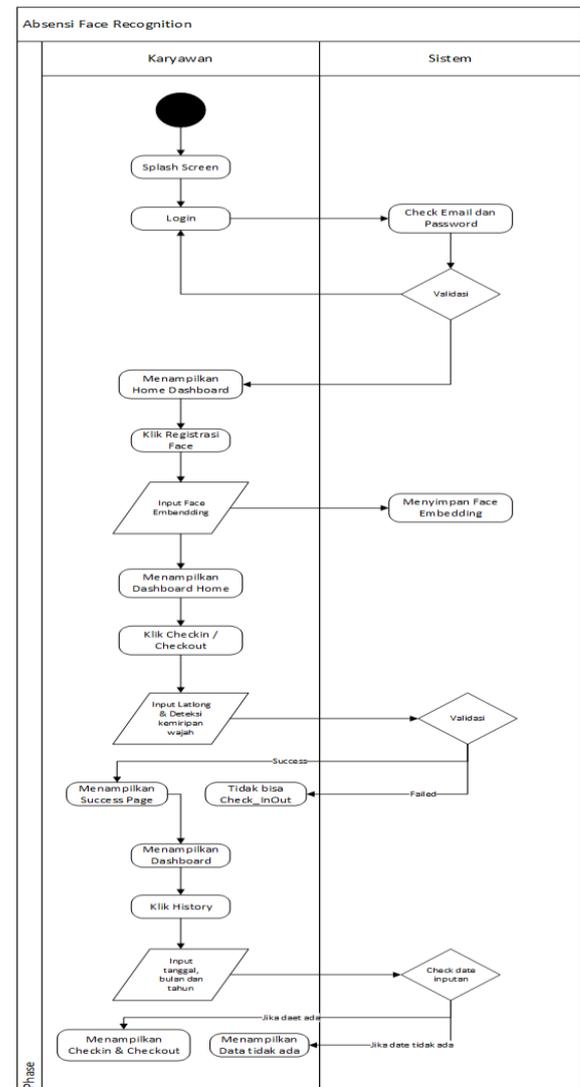
Dengan merancang Entity Relationship Diagram (ERD), akan membantu dalam menggambarkan hubungan antar entitas dalam sistem, serta atribut dan relasi yang ada di antara entitas-entitas tersebut. ERD dari sistem dapat dilihat pada Gambar 4.



Gambar 4. Entity Relationship Diagram

### 2.4.4. Activity Diagram

Dengan merancang Activity Diagram, akan membantu dalam menggambarkan alur aktivitas atau proses yang terjadi dalam sistem, menunjukkan urutan langkah-langkah yang dilakukan oleh pengguna atau sistem dalam mencapai tujuan tertentu. Activity Diagram dari sistem dapat dilihat pada Gambar 5.



Gambar 5. Activity Diagram

## 3. HASIL DAN PEMBAHASAN

### 3.1. Implementasi

#### 1. Implementasi Halaman *Splash Screen*

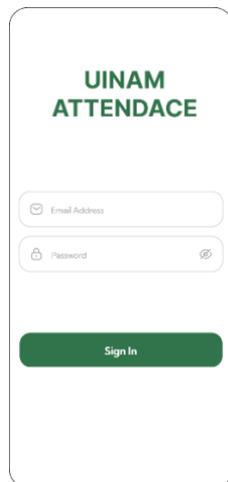
Halaman ini menampilkan tampilan awal sebelum menuju halaman utama. Pada splash screen, aplikasi akan melakukan inisialisasi sistem, seperti memuat data dan mempersiapkan komponen yang diperlukan untuk proses absensi. Tampilan splash screen dapat dilihat pada Gambar 9.



Gambar 9. Halaman *Splash Screen*

## 2. Implementasi Halaman *Login*

Halaman *login* digunakan masuk ke dalam sistem dengan memasukkan username dan password. Informasi login ini akan dikirimkan ke API, yang kemudian memproses dan memvalidasi data yang dikirim untuk memastikan keabsahannya. Setelah berhasil login, pengguna akan diarahkan ke halaman utama aplikasi. Tampilan halaman login dapat dilihat pada Gambar 10.



Gambar 10. Halaman *Login*

## 3. Implementasi *Home Screen*

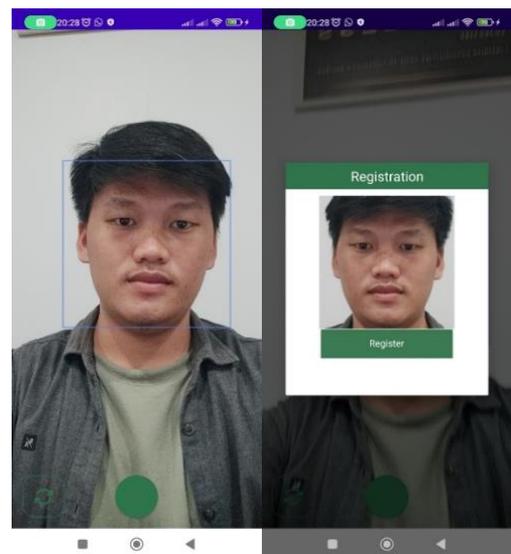
Halaman *Home Screen* ini digunakan oleh pengguna untuk melakukan beberapa tindakan penting, seperti registrasi wajah, absensi datang, dan absensi pulang. Tampilan *Home Screen* dapat dilihat pada Gambar 11.



Gambar 11. *Home Screen*

## 4. Implementasi Halaman Registrasi Wajah

Halaman registrasi wajah digunakan untuk mendaftarkan wajah pengguna ke dalam sistem, sehingga saat melakukan absensi, wajah pengguna dapat dikenali oleh aplikasi menggunakan teknologi face recognition. Pada halaman ini, pengguna diminta untuk menghadap kamera dan mengikuti instruksi untuk memastikan kualitas pendaftaran wajah yang optimal. Setelah berhasil, data wajah akan disimpan dan digunakan untuk keperluan absensi. Tampilan halaman registrasi wajah dapat dilihat pada Gambar 12.

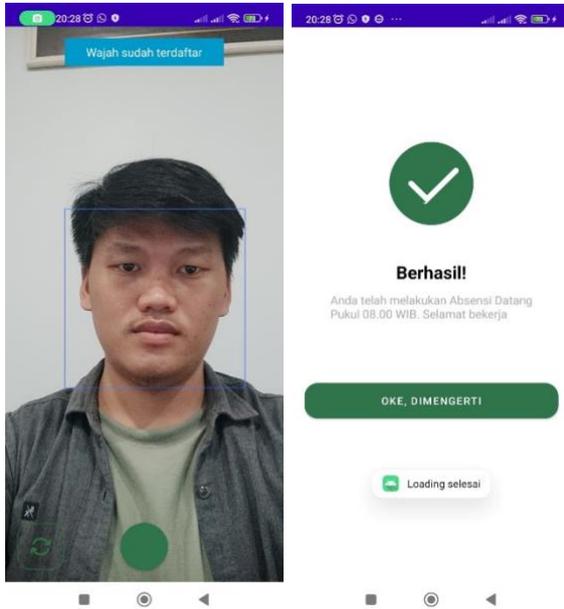


Gambar 12. Registrasi Wajah

## 5. Absensi Datang

Halaman absensi datang digunakan untuk memindai wajah pengguna dan membandingkannya

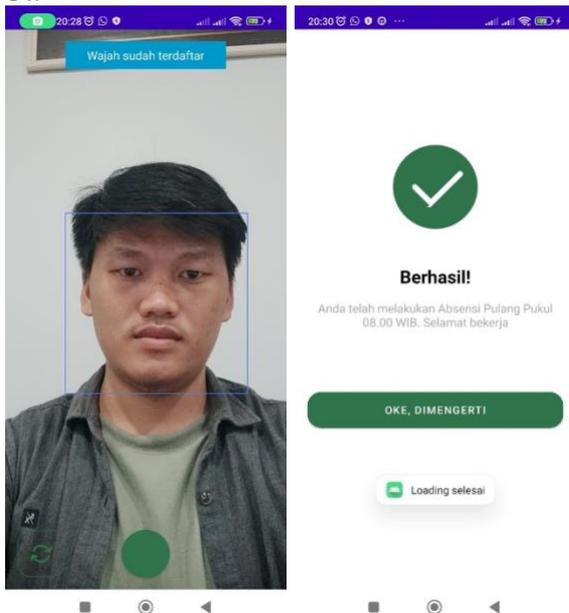
dengan data wajah yang sudah terdaftar di sistem. Ketika sistem mendeteksi kemiripan wajah yang cukup tinggi, absensi kedatangan akan tercatat secara otomatis. Proses ini menggunakan teknologi face recognition untuk memastikan keakuratan identifikasi. Tampilan halaman absensi datang dapat dilihat pada Gambar 13.



Gambar 13. Absensi Datang

#### 6. Implementasi Absensi Pulang

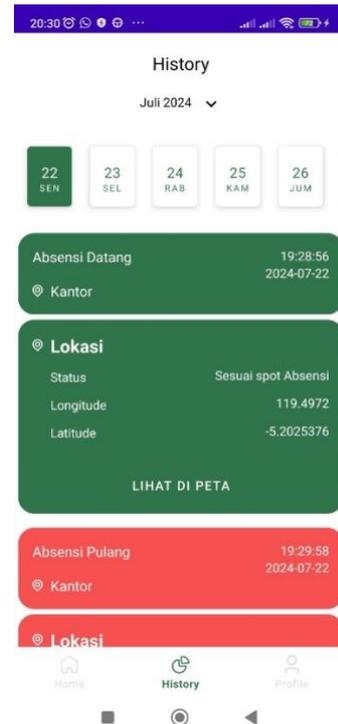
Halaman absensi pulang berfungsi untuk memindai wajah pengguna dan mencocokkannya dengan data wajah yang terdaftar di sistem. Tampilan halaman absensi pulang dapat dilihat pada Gambar 14.



Gambar 14. Absensi Pulang

#### 7. Implementasi History

Halaman History memungkinkan pengguna untuk melihat riwayat absensi mereka dengan memilih tanggal, bulan, dan tahun tertentu. Pada halaman ini, karyawan dapat memeriksa data absensi mereka, baik kedatangan maupun kepulangan, yang tercatat sebelumnya. Fitur ini memberikan kemudahan bagi pengguna untuk melacak kehadiran mereka secara detail. Tampilan halaman History dapat dilihat pada Gambar 15.



Gambar 15. History

### 3.2. Pengujian

Pengujian dalam penelitian ini terbagi menjadi dua jenis, yaitu fungsional testing dan pengujian sistem aplikasi.

#### 3.2.1. Pengujian Fungsional

Fungsional testing akan dilakukan untuk memastikan bahwa setiap fungsi dan fitur dalam sistem berjalan sesuai dengan spesifikasi yang telah ditentukan. Pengujian ini bertujuan untuk mengevaluasi kinerja setiap komponen sistem, seperti registrasi wajah, absensi kedatangan, dan absensi kepulangan. Skenario pengujiannya dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian

NO.	Kasus	Hasil yang diinginkan	Hasil Uji
1	Membuka aplikasi	Menampilkan <i>splashscreen</i> dan <i>loginscreen</i>	Berhasil
2	Memasukkan <i>email</i> dan <i>password</i> dengan benar	Menampilkan pesan berhasil dan berpindah ke halaman utama	Berhasil

3	Memasukkan <i>email</i> dan <i>password</i> dengan salah	Menampilkan pesan gagal <i>login</i>	Berhasil
4	Menekan tombol <i>Home</i> pada halaman utama	Menampilkan Halaman <i>Home</i>	Berhasil
5	Menekan tombol <i>registrasi</i> wajah pada halaman <i>Home</i>	Berpindah kehalaman <i>Registrasi</i> dan mengaktifkan kamera	Berhasil
6	Menekan tombol kamera	Akan menampilkan modal <i>register</i>	Berhasil
7	Menekan tombol <i>register</i> pada modal	Menampilkan pesan <i>register</i> berhasil dan akan berpindah halaman ke <i>Home</i>	Berhasil
8	Menekan tombol Absensi Datang pada halaman <i>Home</i>	Berpindah halaman ke Halaman Absensi Datang dan akan mengaktifkan kamera	Berhasil
9	Menekan tombol ketika wajah terdeteksi sudah terdaftar	Akan beralih ke halaman absensi berhasil	Berhasil
10	Menekan tombol ketika wajah terdeteksi belum terdaftar	Tombol <i>camera</i> tidak akan aktif	Berhasil
	Menekan tombol Oke pada halaman absensi berhasil	Berpindah kembali ke halaman utama	Berhasil
	Menekan tombol absensi pulang	Berpindah ke halaman Absensi pulang dan akan mengaktifkan kamera	Berhasil
	Menekan tombol ketika wajah terdeteksi sudah terdaftar	Akan beralih ke halaman absensi berhasil	Berhasil
	Menekan tombol ketika wajah terdeteksi belum terdaftar	Tombol <i>camera</i> tidak akan aktif	Berhasil
	Menekan tombol Oke pada halaman absensi berhasil	Berpindah kembali ke halaman utama	Berhasil
	Menekan tombol <i>history</i> pada halaman menu utama	Berpindah ke halaman <i>History</i>	Berhasil
	Menekan tanggal bulan dan tahun pada halaman <i>history</i>	Akan menampilkan <i>History</i> Absensi datang dan pulang pada tanggal,	Berhasil

		bulan dan tahun yang terpilih	
	Menekan tombol <i>profile</i>	Berpindah ke halaman <i>Profile</i>	Berhasil
	Menekan tombol <i>Logout</i> pada halaman <i>Profile</i>	Menampilkan pesan <i>Logout</i> berhasil dan akan beralih ke halaman <i>login</i>	Berhasil

### 3.2.2. Pengujian Fungsional

Sementara itu, pengujian sistem aplikasi dilakukan untuk mengukur performa aplikasi, termasuk penggunaan RAM dan CPU, serta kecepatan kompilasi file. Pengujian ini bertujuan untuk menilai seberapa efisien aplikasi dalam mengelola sumber daya perangkat dan seberapa cepat aplikasi dapat dibangun atau di-compile menjadi file APK.

#### 1. CPU Native App

**Tabel 2.** CPU Native App

	Fitur	Jumlah	Total	Rata Rata Tertimbang
1	<i>Login</i>	19	89	
2	<i>Register Face</i>	46	710	
3	Absensi Datang	31	461	
4	Absensi Pulang	31	445	
5	<i>History</i>	7	43	
6	<i>Logout</i>	3	21	
Total		137	1769	12.91%

#### 2. CPU Flutter App

**Tabel 3.** CPU Flutter App

	Fitur	Jumlah	Total	Rata Rata Tertimbang
1	<i>Login</i>	58	554	
2	<i>Register Face</i>	99	2445	
3	Absensi Datang	71	1571	
4	Absensi Pulang	69	1682	
5	<i>History</i>	15	165	
6	<i>Logout</i>	8	88	
Total		319	6517	20.43%

Berdasarkan tabel di atas penggunaan CPU pada *kotlin native* lebih kecil dari pada *framework flutter*.

#### 3. RAM Native App

**Tabel 4.** RAM Native App

	Fitur	Jumlah	Total	Rata Rata Tertimbang
1	<i>Login</i>	15	1513.5	
2	<i>Register Face</i>	31	6094.9	
3	Absensi Datang	15	3052.1	
4	Absensi Pulang	18	3834	
5	<i>History</i>	3	616.8	
6	<i>Logout</i>	2	413.2	
Total		84	15524.5	184.815 mb

#### 4. RAM Flutter App

**Tabel 5. RAM Flutter App**

	Fitur	Jumlah	Total	Rata Rata Tertimbang
1	Login	37	17763.6	
2	Register Face	47	29500.8	
3	Absensi Datang	32	21544.8	
4	Absensi Pulang	36	24590.6	
5	History	8	5074.4	
6	Logout	5	3163.5	
Total		165	101637.7	615.987 mb

Berdasarkan tabel RAM diatas penggunaan kapasitas *memory* lebih tinggi pada *kotlin native* jika di bandingkan dengan *framework flutter*.

#### 5. Request Data Native

**Tabel 6. Request Data Native**

No.	Fitur	Time (ms)
1	Login	1209
2	Register Face	542
3	Absensi Datang	498
4	Absensi Pulang	398
5	History	350
6	Logout	357
Rata Rata		559

#### 6. Request Data Flutter

**Tabel 7. Request Data Flutter**

No.	Fitur	Time (ms)
1	Login	1033
2	Register Face	1800
3	Absensi Datang	1653
4	Absensi Pulang	1564
5	History	719
6	Logout	571
Rata Rata		1223.333

Berdasarkan tabel Request Data *kotlin native* jauh lebih cepat bila di bandingkan dengan *framework flutter*.

#### 7. Kecepatan Compile .apk

**Tabel 8. Kecepatan Compile .apk**

No.	Uji	Time
1	Native	19s
2	Flutter	108s

Pada tabel *kompilasi .apk* juga menunjukkan jarak yang sangat jauh antar *kotlin native* dan *framework flutter* dalam hal *kompilasi .apk*.

#### 4. KESIMPULAN

Berdasarkan hasil penelitian Analisis Perbandingan Antara *Framework Flutter* Dengan *Native* Untuk Pengembangan Aplikasi *Mobile Absensi Face Recognition*, aplikasi *Native* menunjukkan performa yang lebih baik dalam

*request* data dan kecepatan kompilasi di bandingkan dengan *flutter*. Penggunaan *CPU* dan *RAM* pada aplikasi *native* juga lebih rendah dari pada *framework flutter*, ini menunjukkan efisiensi *CPU* dan *RAM* lebih baik pada aplikasi *native*. Dari hal tersebut diketahui bahwa keunggulan *native* di karenakan pada *flutter* perlu menerjemahkan kodenya kedalam *native* menggunakan mesin rendering sendiri yang disebut *Dart Virtual Machine (Dart VM)*, proses ini menggunakan lebih banyak siklus *CPU* dan *RAM* karena adanya lapisan abstraksi tambahan yang menyebabkan penambahan beban peforma, begitu juga pada hal *request* data dimana *native* memiliki integrasi yang lebih dalam dengan sistem jaringan di platform, memungkinkan proses permintaan dan respons data berjalan lebih cepat dan efisien. Oleh karna itu, pengembang aplikasi dapat mempertimbangkan penggunaan *framework Native* untuk mengoptimalkan performa aplikasi secara keseluruhan, terutama untuk aplikasi yang memerlukan waktu respons cepat, efisiensi *CPU*, dan penggunaan memori yang lebih rendah.

Perancangan sistem ini masih jauh dari sempurna, oleh karena itu penulis mengajukan beberapa saran untuk pengembangan lebih lanjut. Pertama, diperlukan penyelidikan mengenai kinerja jangka panjang dari aplikasi *Native* dan *Flutter*, dengan melihat bagaimana masing-masing *framework* menangani tugas berulang dan penggunaan berkelanjutan, yang dapat memberikan wawasan tambahan terkait efisiensi dan stabilitas. Selain itu, perlu dilakukan eksplorasi lebih mendalam tentang dampak penggunaan *framework* terhadap proses pengembangan dan pemeliharaan aplikasi, dengan mempertimbangkan faktor-faktor seperti kemudahan debugging, ketersediaan dokumentasi, dukungan komunitas pengembang, serta dukungan jangka panjang dari *framework* tersebut. Terakhir, pengujian pada berbagai kondisi beban dan skala pengguna perlu dilakukan untuk mengetahui bagaimana aplikasi *Native* dan *Flutter* berperforma dalam situasi yang lebih variatif, sehingga dapat memberikan gambaran yang lebih komprehensif mengenai keandalan dan efisiensi kedua *framework* dalam lingkungan yang dinamis.

#### 5. DAFTAR PUSTAKA

- [1] Ceme, D., Faizah, N., & Koryanto, L. (2023). Aplikasi Presensi Kehadiran Guru Di Smkn Pakisjaya Berbasis Face Recognition Menggunakan Opencv. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, 4(1), 1–7. <https://doi.org/10.35870/jimik.v4i1.104>
- [2] Saraiva, J. A. (2024). A large-scale empirical study on mobile performance: energy, run-time and memory.
- [3] Dwi Kurniawan, F., & Nur Laila, S. (2023). Implementasi Teknologi Geolocation Pada Aplikasi Presensi Karyawan IIB Darmajaya

- Menggunakan Metode SCRUM Berbasis Mobile. 93, 700261.
- [4] Haris, Z. A. (2021). Komparasi Flutter Dan React Native Dalam Pengembangan Perangkat Bergerak.
- [5] Olsson, M. (2020). A Comparison of Performance and Looks Between Flutter and Native Applications When to prefer Flutter over native in mobile application development. [www.bth.se/dipt](http://www.bth.se/dipt)
- [6] Pradana, I. A. (2020). Analisis Perbandingan Antara Framework Flutter Dengan Framework React Native Untuk Pengembangan Aplikasi Mobile.
- [7] Rodríguez-Sánchez Guerra, M. (2018). Cross-Platform Development Frameworks For The Development Of Hybrid Mobile Applications: Implementations And Comparative Analysis.
- [8] Salengke, M. R. (2023). Aplikasi Monitoring Presensi Menggunakan Teknologi Face Recognition Berbasis Android.
- [9] Wibawanto, N. F., Astuti, Y. P., Anisa, N., Winarsih, S., Saraswati, G. W., & Rohman, M. S. (2023). Sistem Permohonan Ijin Berbasis Website Menggunakan Framework Laravel Dengan Metodologi Scrum. *Jurnal Manajemen Informatika & Sistem Informasi (MISI)*, 6(1). <https://doi.org/10.36595/misi.v5i2>
- [10] Faisal, F. (2022). Game Visual Novel dengan Metode Rapid Application Development (RAD) Berbasis Mobile Learning (Doctoral dissertation, Prodi Teknik Informatika).
- [11] Christanto, E., & Wibowo, T. (2020, November). Analisis Komparasi Performa Web Application: Studi Kasus Asp. Net Mvc Dan Asp. Net Core. In *Conference on Business, Social Sciences and Innovation Technology* (Vol. 1, No. 1, pp. 459-466).
- [12] Sinlae, F. (2021). Prototipe Sistem Absensi Berbasis Web Dan Mobile Dengan Metode Rapid Application Development (Rad). *Prosiding SISFOTEK*, 5(1), 107-112.
- [13] Hardianto, F. (2022). Aplikasi Presensi Online WFH Menggunakan Sidik Jari Dan Face Recognition Berbasis Mobile (Studi Kasus: PT Toko Damai Sejahtera Kota Rengat) (Doctoral dissertation, Universitas Islam Riau).