

**ANALISIS PERBANDINGAN *FRAMEWORK FLUTTER*
DAN *REACT NATIVE* UNTUK PENGEMBANGAN
APLIKASI *MOBILE* YANG BERSIFAT *CROSS-
PLATFORM***

SKRIPSI

Disusun Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Komputer
dari Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang



Oleh:

ADAM MUKTI WIBISONO

1810631170103

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SINGAPERBANGSA KARAWANG
KARAWANG**

2022

LEMBAR PENGESAHAN

**ANALISIS PERBANDINGAN *FRAMEWORK FLUTTER*
DAN *REACT NATIVE* UNTUK PENGEMBANGAN APLIKASI *MOBILE*
YANG BERSIFAT *CROSS-PLATFORM***

SKRIPSI

Disusun Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Komputer
dari Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang

Oleh:

**ADAM MUKTI WIBISONO
1810631170103**

Disetujui oleh:

Pembimbing 1

Pembimbing 2

**Aji Primajaya, S.Si., M.Kom.
NIDN. 0026048706**

**Kamal Prihandani, M.Kom.
NIDN. 0002069106**

Karawang, Agustus 2022

Diketahui dan disahkan

oleh:

Dekan Fakultas Ilmu Komputer

**Dr. Mayasari, S.Si., M.Hum.
NIDN. 0426097905**

LEMBAR PENGESAHAN PENGUJI

**ANALISIS PERBANDINGAN *FRAMEWORK FLUTTER*
DAN *REACT NATIVE* UNTUK PENGEMBANGAN APLIKASI *MOBILE*
YANG BERSIFAT *CROSS-PLATFORM***

SKRIPSI

Disusun Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Komputer
dari Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang

Oleh:

**ADAM MUKTI WIBISONO
1810631170103**

Disetujui oleh:

Penguji 1

Penguji 2

**Arip Solehudin, M.Kom.
NIDN. 0016048902**

**Oman Komarudin, S.Si., M.Kom.
NIDN. 0406047702**

Karawang, Agustus 2022

Diketahui dan disahkan
oleh:

Dekan Fakultas Ilmu Komputer

**Dr. Mayasari, M.Hum.
NIDN. 0426097905**

HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan dibawah ini :

Nama : Adam Mukti Wibisono

NPM : 1810631170103

Judul Skripsi : Analisis Perbandingan *Framework Flutter* dan *React Native*
untuk Pengembangan Aplikasi Mobile yang Bersifat *Cross-Platform*

Menyatakan dengan sebenarnya bahwa penulisan skripsi ini adalah hasil pekerjaan saya dan seluruh ide, pendapat, atau materi dari sumber lain telah dikutip dengan cara penulisan referensi yang sesuai.

Pernyataan ini saya buat dengan sebenar-benarnya dan jika pernyataan ini tidak sesuai dengan kenyataan, maka saya bersedia menanggung sanksi yang akan dikenakan kepada saya, termasuk pencabutan gelar Sarjana Komputer yang nanti saya dapatkan.

Karawang, Agustus 2022

Adam Mukti Wibisono

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI KARYA
ILMIAH UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Singaperbangsa Karawang, saya yang bertandatangan di bawah ini:

Nama : Adam Mukti Wibisono
NPM : 1810631170103
Program Studi : Teknik Informatika
Fakultas : Ilmu Komputer
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Singaperbangsa Karawang **Hak Bebas Royalti Non Eksklusif (Non Exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul:

**“ANALISIS PERBANDINGAN *FRAMEWORK FLUTTER*
DAN *REACT NATIVE* UNTUK PENGEMBANGAN APLIKASI *MOBILE*
YANG BERSIFAT *CROSS-PLATFORM*”**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif ini, Universitas Singaperbangsa Karawang berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Karawang
Pada Tanggal : Agustus 2022

Yang menyatakan

(Adam Mukti Wibisono)

PERSEMBAHAN

“Skripsi ini saya selesaikan atas izin Allah SWT dan saya persembahkan untuk kedua orang tua saya (Bpk. Mukhlisin dan Ibu Titi) yang selalu mendukung, memberikan doa, moril, materil serta kepercayaannya sehingga saya mampu untuk menyelesaikan perkuliahan dengan baik dan tepat waktu. Tiada terhingga rasa syukur serta ucapan terima kasih untuk kedua orang tua.”

MOTTO

“Konsistenlah dalam melakukan sesuatu. Sebab, semua orang akan menjadi lebih baik jika terus menekuninya”

ABSTRAK

ANALISIS PERBANDINGAN *FRAMEWORK FLUTTER* DAN *REACT NATIVE* UNTUK PENGEMBANGAN APLIKASI *MOBILE* YANG BERSIFAT *CROSS-PLATFORM*

¹Adam Mukti Wibisono, ²Aji Primajaya, S.Si., M.Kom.,
³Kamal Prihandani, M.Kom.

¹adam.mukti18103@student.unsika.ac.id,
²aji.primajaya@staff.unsika.ac.id, ³kamal.prihandani@unsika.ac.id
Program Studi Teknik Informatika
Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang

Abstrak - Jumlah pengguna *smartphone* di seluruh dunia mencapai 6 miliar pengguna dan diperkirakan akan terus bertambah dalam beberapa tahun mendatang. Oleh karena jumlah pengguna *mobile phone* semakin meningkat dari tahun ke tahun, maka kebutuhan akan aplikasi berbasis *mobile* sangatlah tinggi. Sistem operasi *mobile* yang sangat populer saat ini salah satunya adalah Android dan juga iOS. Dikarenakan perbedaan sistem operasi inilah seorang pengembang harus mengembangkan aplikasi yang dapat berjalan pada masing-masing *platform*. Hal ini tentu akan membutuhkan waktu, biaya, dan usaha yang lebih besar. Untuk mengatasi hal tersebut dikembangkan sebuah *framework* pengembangan aplikasi *mobile* yang bersifat *cross-platform* seperti *Flutter* dan *React Native*. Pada penelitian ini penulis menganalisis perbandingan antara kedua *framework* tersebut, tentunya dengan kriteria pengukuran yang sudah ditentukan. Untuk tahap pengembangan aplikasi penelitian ini menggunakan metode *Rapid Application Development* (RAD) karena cepat dan cocok untuk aplikasi yang tidak terlalu kompleks. Hasil pengujian dari perbandingan ukuran aplikasi *Flutter* 48% lebih kecil dibandingkan aplikasi *React Native*, untuk pengujian kompleksitas kode keduanya memberikan hasil yang cukup baik dengan nilai *cyclomatic complexity* sama-sama bernilai 2, dan pada pengujian perbandingan performa pada penggunaan CPU aplikasi *Flutter* lebih rendah 4% dari aplikasi *React Native*, kemudian konsumsi memori aplikasi *React Native* lebih rendah 7,5% dari aplikasi *Flutter* dan untuk pengujian *framerate* diantara kedua aplikasi nilai rata-rata nya hanya terpaut 0,86% dimana aplikasi *Flutter* sedikit lebih unggul dari aplikasi *React Native*.

Kata Kunci : Analisis perbandingan, *Cross-platform*, *Flutter*, *Mobile*, *React Native*

ABSTRACT

COMPARISON ANALYSIS OF FLUTTER AND REACT NATIVE FRAMEWORK FOR CROSS-PLATFORM MOBILE APPLICATION DEVELOPMENT

**¹Adam Mukti Wibisono, ²Aji Primajaya, S.Si., M.Kom.,
³Kamal Prihandani, M.Kom.**

¹adam.mukti18103@student.unsika.ac.id,

²aji.primajaya@staff.unsika.ac.id, ³kamal.prihandani@unsika.ac.id

Informatics Engineering Study Program

Faculty of Computer Science University Singaperbangsa Karawang

Abstract - The number of smartphone users worldwide has reached 6 billion users and is expected to continue to grow in the next few years. Because the number of mobile phone users is increasing from year to year, the need for mobile-based applications is very high. One of the most popular mobile operating systems today is Android and iOS. Due to differences in operating systems, a developer must develop applications that can run on each platform. This will certainly require more time, cost, and effort. To overcome this, a cross-platform mobile application development framework such as Flutter and React Native was developed. In this study, the authors analyze the comparison between the two frameworks, of course with predetermined measurement criteria. For the application development stage, this research uses the Rapid Application Development (RAD) method because it is fast and suitable for applications that are not too complex. The result The Flutter application size comparison test is 48% smaller than the React Native application, for code complexity testing both of them give quite good results with the cyclomatic complexity value being equal to 2, and in the performance comparison test on CPU usage the Flutter application is 4% lower than React Native applications, then the memory consumption of React Native applications is 7.5% lower than Flutter applications and for testing the framerate between the two applications the average value is only 0.86% which Flutter applications are slightly superior to React Native applications.

Keywords : Comparison analysis, Cross-platform, Flutter, Mobile, React Native

KATA PENGANTAR



Puji dan Syukur bagi Allah SWT, atas rahmat dan hidayah-Nya, peneliti dapat menyelesaikan karya tulis yang berbentuk proposal skripsi dengan judul “Analisis Perbandingan *Framework Flutter* Dan *React Native* Untuk Pengembangan Aplikasi *Mobile* Yang Bersifat *Cross-Platform*” sesuai dengan waktu yang telah direncanakan.

Shalawat serta salam semoga senantiasa tercurah limpahkan kepada Nabi Besar Muhammad SAW beserta keluarga dan sahabatnya. Adapun proposal skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan program studi S1 Teknik Informatika dan memperoleh gelar Sarjana Komputer di Universitas Singaperbangsa Karawang

Pada penyusunan proposal ini, peneliti tentunya mendapat bimbingan, dukungan, kritik dan saran yang membangun dari berbagai pihak. Oleh karena itu, peneliti ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Ibu Prof. Dr. Sri Mulyani, AK., CA, selaku Rektor Universitas Singaperbangsa Karawang.
2. Ibu Dr. Mayasari, M.Hum., selaku Dekan Fakultas Ilmu Komputer.
3. Bapak Garno, S.Kom., M.Kom., selaku Wakil Dekan Bidang Akademik dan Kemahasiswaan Fakultas Ilmu Komputer.
4. Bapak Muhammad Jajuli, S.Si, M.Si., selaku Wakil Dekan bidang umum dan keuangan Fakultas Ilmu Komputer.
5. Ibu Betha Nurina Sari, M.Kom., selaku Koordinator Program Studi Teknik Informatika Fakultas Ilmu Komputer.
6. Bapak Aji Primajaya, S.Si., M.Kom., selaku dosen pembimbing yang telah memberikan arahan, saran, serta motivasi
7. Seluruh Dosen dan Staf Tata Usaha Fakultas Ilmu Komputer

8. Yang teristimewa kepada kedua orang tua saya, Bapak Mukhlisin dan Ibu Titi Sumanti, serta kedua Adik saya Camelya Dwi Nusantara dan Adnan Pandu Nugraha, yang senantiasa banyak banyak memberikan kasih sayang, dukungan moril maupun materil, nasihat dan do'a, sehingga perkuliahan dan penyusunan penelitian ini dapat terlaksana dengan baik.
9. Teman-teman mahasiswa Fakultas Ilmu Komputer angkatan 2018 terkhusus kelas D yang telah berjuang bersama melewati masa kuliah dengan penuh suka dan duka.
10. Semua pihak lainnya yang telah membantu dalam menyelesaikan proposal skripsi ini yang tidak dapat disebutkan satu per satu.

Semoga amal baik dari semua pihak yang telah membantu dalam penelitian skripsi ini mendapat balasan yang terbaik di sisi Allah SWT. Peneliti menyadari bahwa masih banyak kekurangan pada proposal skripsi ini, dari segi penulisan maupun pengetahuan yang terbatas, maka saran dan kritik yang membangun sangat diharapkan sehingga proposal skripsi ini dapat bermanfaat bagi peneliti pada khususnya dan pembaca pada umumnya. Aamiin Yaa Rabbal Alamiin.

Karawang, Agustus 2022

Adam Mukti Wibisono

DAFTAR ISI

ABSTRAK	vii
ABSTRACT	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.5.1 Manfaat Teoritis	4
1.5.2 Manfaat Praktis	4
1.6 Metodologi Penelitian.....	4
1.7 Sistematika Penulisan.....	5
BAB II LANDASAN TEORI	1
2.1 <i>Cross-platform framework</i>	1
2.1.1 <i>Progressive Web Apps (PWA)</i>	1
2.1.2 <i>Hybrid Application</i>	2
2.1.3 <i>Interpreted Application</i>	2
2.1.4 <i>Cross-compiled Application</i>	2
2.2 <i>Flutter</i>	3
2.2.1 <i>Dart</i>	4
2.2.2 <i>Widget</i>	4
2.3 <i>React Native</i>	5
2.3.1 <i>Javascript</i>	5
2.3.2 <i>JSX</i>	6

2.3.3	<i>Component</i>	7
2.4	<i>Rapid Application Development (RAD)</i>	7
2.5	<i>Android Package Kit (APK)</i>	8
2.6	Kompleksitas Kode	9
2.7	<i>Central Processing Unit (CPU)</i>	9
2.8	<i>Random Access Memory (RAM)</i>	10
2.9	<i>Frame Rate</i>	10
2.10	Penelitian Sebelumnya	11
2.11	Penelitian Sekarang	15
BAB III OBJEK DAN METODOLOGI PENELITIAN		16
3.1	Objek Penelitian	16
3.2	Metodologi Penelitian	16
3.3	Rancangan Penelitian	16
3.3.1	Rancangan Pengembangan Aplikasi	16
3.3.1.1	Perencanaan Kebutuhan	17
3.3.1.2	Desain Sistem	17
3.3.1.3	Proses Pengembangan	17
3.3.1.4	Implementasi	17
3.3.2	Rancangan Pengujian Aplikasi	17
3.3.2.1	Analisis Objek Pengujian	18
3.3.2.2	Penentuan Skenario Pengujian	18
3.3.2.3	Tahap Pengujian	18
3.3.2.4	Kesimpulan	19
BAB IV HASIL DAN PEMBAHASAN		20
4.1	Hasil Penelitian	20
4.1.1	Pengembangan Aplikasi	20
4.1.1.1	Perencanaan Kebutuhan	20
4.1.1.2	Desain Sistem	22
4.1.1.3	Proses Pengembangan	29
4.1.1.4	Implementasi	33
4.1.2	Pengujian Aplikasi	34

4.1.2.1	Analisis Objek Pengujian.....	35
4.1.2.2	Skenario Pengujian	35
4.1.2.3	Tahap Pengujian	36
4.1.2.4	Kesimpulan	48
BAB V KESIMPULAN DAN SARAN		50
5.1	Kesimpulan	50
5.2	Saran.....	50
DAFTAR PUSTAKA		52
LAMPIRAN.....		54

DAFTAR GAMBAR

Gambar 3.1 Diagram Rapid Application Development (RAD)	16
Gambar 4. 1 Use Case Diagram	22
Gambar 4. 2 Activity Diagram Lihat Daftar Mahasiswa.....	23
Gambar 4. 3 Activity Diagram Cari Data Mahasiswa.....	23
Gambar 4. 4 Activity Diagram Lihat Detail Mahasiswa	24
Gambar 4. 5 Desain Halaman Utama (Flutter).....	25
Gambar 4. 6 Desain Halaman Daftar Mahasiswa (Flutter)	25
Gambar 4. 7 Desain Halaman Detail Mahasiswa (Flutter).....	26
Gambar 4. 8 Desain Halaman Not Found (Flutter)	26
Gambar 4. 9 Desain Halaman Utama	27
Gambar 4. 10 Desain Halaman Daftar Mahasiswa (React Native)	27
Gambar 4. 11 Desain Halaman Detail Mahasiswa (React Native).....	28
Gambar 4. 12 Desain Halaman Not Found (React Native)	28
Gambar 4. 13 Halaman Utama (Flutter).....	29
Gambar 4. 14 Halaman Daftar Mahasiswa (Flutter)	29
Gambar 4. 15 Halaman Daftar Mahasiswa (Pencarian) (Flutter).....	30
Gambar 4. 16 Halaman Detail Mahasiswa (Flutter).....	30
Gambar 4. 17 Halaman Not Found (Flutter)	31
Gambar 4. 18 Halaman Utama (React Native).....	31
Gambar 4. 19 Halaman Daftar Mahasiswa (React Native)	32
Gambar 4. 20 Halaman Daftar Mahasiswa (Pencarian) (React Native).....	32
Gambar 4. 21 Halaman Detail Mahasiswa	33
Gambar 4. 22 Halaman Not Found (React Native)	33

Gambar 4. 23 Hasil Build Apk Aplikasi Flutter.....	36
Gambar 4. 24 Hasil Build Apk Aplikasi React Native.....	37
Gambar 4. 25 Perbandingan Ukuran Aplikasi.....	37
Gambar 4. 26 Flowgraph Function runfilter Pada Aplikasi Flutter Dan React Native	41
Gambar 4. 27 Persentase Penggunaan CPU	43
Gambar 4. 28 Penggunaan CPU Aplikasi Flutter.....	44
Gambar 4. 29 Penggunaan CPU Aplikasi React Native.....	44
Gambar 4. 30 Konsumsi Memori	45
Gambar 4. 31 Penggunaan Memori Aplikasi Flutter.....	45
Gambar 4. 32 Penggunaan Memori Aplikasi React Native.....	46
Gambar 4. 33 Performance Overlay Aplikasi Flutter.....	47
Gambar 4. 34 Performance Overlay Aplikasi React Native.....	47
Gambar 4. 35 Grafik Framerate Aplikasi	48

DAFTAR TABEL

Tabel 2.1 Perbandingan JSX dan Javascript	6
Tabel 2.2 Penelitian sebelumnya.....	11
Tabel 4. 1 Analisis kebutuhan fungsional	20
Tabel 4. 2 Analisis kebutuhan non fungsional	21
Tabel 4. 3 Analisis kebutuhan lingkungan	22
Tabel 4. 4 Black Box Testing.....	34
Tabel 4. 5 Klasifikasi Studi Kasus Pengujian	35
Tabel 4. 6 Skenario Pengujian.....	35
Tabel 4. 7 Package yang di Import.....	38
Tabel 4. 8 Kode Function runfilter Pada Aplikasi Flutter	40
Tabel 4. 9 Kode Function runfilter Pada Aplikasi React Native.....	40
Tabel 4. 10 Hasil Analisis Kompleksitas Kode Pada Function runfilter.....	41
Tabel 4. 11 Hasil Pengujian CPU.....	42

BAB I

PENDAHULUAN

1.1 Latar Belakang

Mobile phone menjadi salah satu perangkat yang cukup penting bagi sebagian besar masyarakat. Apalagi di tengah pandemi *Covid-19* dimana masyarakat diharapkan untuk membatasi ataupun meminimalisir kegiatan di luar rumah. Tidak sekedar untuk sarana komunikasi kegunaan *mobile phone* ini menjadi sangat luas, sebagai contoh dalam menunjang pekerjaan *mobile phone* dapat digunakan untuk melakukan *meeting* dengan rekan kerja, selain itu *mobile phone* digunakan juga untuk aktivitas perbelanjaan, seperti membeli barang ataupun memesan makanan, dan yang paling sering *mobile phone* digunakan untuk sarana hiburan di tengah pandemi yang mengharuskan masyarakat untuk tetap dirumah saja.

Jumlah pengguna *smartphone* di seluruh dunia mencapai 6 miliar pengguna dan diperkirakan akan terus bertambah dalam beberapa tahun mendatang. China, India dan Amerika Serikat merupakan negara dengan jumlah pengguna *smartphone* terbanyak (*Smartphone Users 2026 | Statista, n.d.*). Sedangkan di Indonesia sendiri pengguna *Smartphone* diperkirakan mencapai 199,2 juta pengguna pada tahun 2021, Indonesia merupakan negara keempat dengan pengguna *smartphone* terbanyak di dunia setelah China, India, dan Amerika Serikat. (*Indonesia: Smartphone Users 2026 | Statista, n.d.*).

Jumlah pengguna *mobile phone* semakin meningkat dari tahun ke tahun, sehingga kebutuhan pada aplikasi berbasis *mobile* sangatlah tinggi. Sistem operasi *mobile* yang sangat populer saat ini salah duanya adalah Android dan juga iOS, sistem operasi Android memegang 71,08% sedangkan iOS hanya 28,23% *market share* di seluruh dunia (*Mobile Operating System Market Share Worldwide | Statcounter Global Stats, n.d.*). Dikarenakan perbedaan sistem operasi inilah seorang pengembang harus mengembangkan aplikasi yang dapat berjalan pada masing-masing *platform*, sementara untuk mengembangkan aplikasi pada *platform* tersebut dibutuhkan *tool* dan bahasa pemrograman yang berbeda, dimana dalam pengembangan aplikasi android membutuhkan bahasa pemrograman Java/Kotlin

dan Android Studio sebagai IDE nya, sedangkan untuk pengembangan aplikasi berbasis iOS menggunakan bahasa pemrograman Swift/Objective-c (Anggit et al., 2020). Hal ini tentu akan membutuhkan waktu, biaya, dan usaha yang lebih besar dalam mengembangkan aplikasi untuk masing-masing *platform*.

Mengatasi hal tersebut dikembangkan sebuah *framework* pengembangan aplikasi *mobile* yang bersifat *cross-platform* seperti *React Native*. Dengan menggunakan *React Native* pengembang hanya perlu membuat satu basis kode yang kemudian dapat dijalankan pada sistem operasi Android maupun iOS. Di bulan Mei 2017 *Google* meluncurkan sebuah *cross-platform framework* dengan nama *Flutter* dan kemudian *release* versi stabil nya pada 4 Desember 2018. Baik *React Native* ataupun *Flutter* menjanjikan performa yang setara dengan aplikasi *native*.

Hal yang melatarbelakangi penelitian untuk membandingkan *framework* *Flutter* dan *React Native* adalah karena kedua *framework* ini yang paling banyak digunakan, menurut data dari *statista.com* pada tahun 2021 jumlah pengguna *framework* *Flutter* sebanyak 42% dan *React Native* sebanyak 38%, kedua *cross-platform mobile framework* ini adalah yang paling banyak digunakan oleh *developer* di seluruh dunia dalam rentang tahun 2019-2021 (*Cross-Platform Mobile Frameworks Used by Global Developers 2021* | *Statista*, n.d.).

Flutter adalah *framework open source* yang dikembangkan oleh *Google* dan digambarkan sebagai “UI *toolkit* untuk membangun aplikasi yang indah dan dikompilasi secara *native* untuk perangkat *mobile*, web, *desktop*, dari satu basis kode (*Beautiful Native Apps in Record Time* | *Flutter*, n.d.). *Flutter* pertama kali dirilis pada tahun 2017, namun untuk versi stabil nya dirilis pada Desember 2018. Dalam waktu yang singkat *Flutter* menjadi salah satu alternatif yang cukup populer untuk mengembangkan aplikasi *cross-platform*.

React Native adalah sebuah *Javascript framework* yang dibuat oleh *Facebook* untuk mengembangkan aplikasi *mobile*. Berbasiskan *React*, sebuah *Javascript library* untuk membangun antarmuka pengguna yang juga dibuat oleh *Facebook*. Sedang *React* digunakan untuk membuat antarmuka pada web, *React*

Native ditargetkan untuk *mobile platform* dengan mengkompilasi kode untuk masing-masing *platform*.

Dari kedua *framework* yang sangat populer ini manakah yang paling cocok digunakan untuk mengembangkan aplikasi *mobile* lintas *platform*, karena setiap *framework* pastinya memiliki kelebihan dan kekurangan. Oleh karena itu dalam penelitian ini penulis akan menganalisis perbandingan antara kedua *framework* tersebut, tentunya dengan kriteria pengukuran yang sudah ditentukan. Dengan adanya penelitian ini diharapkan dapat membantu perusahaan maupun pengembang dalam menentukan *cross-platform mobile application framework* yang paling tepat sesuai dengan kebutuhan proyek yang dikembangkan.

1.2 Rumusan Masalah

Dari pemaparan latar belakang diatas yang menjadi rumusan masalah pada penelitian ini adalah:

1. Bagaimana mengembangkan aplikasi *mobile* menggunakan *framework Flutter* dan *React Native*?
2. Bagaimana melakukan analisis perbandingan diantara kedua *framework*?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah antara lain:

1. *Mobile application framework* yang digunakan pada penelitian ini adalah *Flutter* dan *React Native*.
2. *Platform* yang digunakan pada penelitian ini hanya pada sistem operasi Android.
3. Pengujian menggunakan aplikasi *prototype* dengan studi kasus aplikasi sistem informasi tugas akhir.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Mengembangkan aplikasi *mobile* menggunakan *framework Flutter* dan *React Native*.

2. Membandingkan *framework Flutter* dan *React Native* berdasarkan parameter ukuran aplikasi, kompleksitas kode, dan performa.

1.5 Manfaat Penelitian

Adapun manfaat yang diperoleh dari penelitian ini adalah:

1.5.1 Manfaat Teoritis

Hasil dari penelitian ini diharapkan dapat memberikan informasi dan pengetahuan tentang kinerja dan juga performa yang diberikan oleh *framework Flutter* dan juga *React Native*.

1.5.2 Manfaat Praktis

Hasil dari penelitian ini juga diharapkan dapat menjadi gambaran bagi perusahaan maupun pengembang dalam menentukan *cross-platform mobile application framework* yang paling tepat sesuai dengan kebutuhan proyek yang dikembangkan.

1.6 Metodologi Penelitian

Langkah – langkah metodologi pada penelitian ini adalah sebagai berikut

1. Metode Pengembangan Aplikasi

Metode pengembangan aplikasi yang digunakan pada penelitian ini adalah metode *Rapid Application Development* (RAD). Adapun tahapan-tahapannya adalah sebagai berikut:

- a. Perencanaan Kebutuhan
- b. Desain Sistem
- c. Proses Pengembangan
- d. Implementasi atau penyelesaian produk

2. Metode Komparasi *Framework*

Untuk melakukan tahap komparasi, setidaknya berikut ini adalah tahapan-tahapan yang dilakukan pada penelitian ini:

- a. Penentuan objek yang akan uji
- b. Penentuan parameter pengujian
- c. Penentuan skenario pengujian
- d. Tahap pengujian

- e. Analisis
- f. kesimpulan

1.7 Sistematika Penulisan

Sistematika penulisan skripsi pada penelitian ini meliputi lima bagian yaitu sebagai berikut:

BAB 1 : PENDAHULUAN

Bab 1 pendahuluan berisi pembahasan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian (teoritis dan praktis), metodologi penelitian, sistematika penelitian dan jadwal penelitian.

BAB 2 : LANDASAN TEORI

Bab 2 landasan teori berisikan teori yang dibutuhkan dan berhubungan dengan penelitian yang akan dilakukan. Pada akhir bab ini akan dipaparkan beberapa rangkuman dari jurnal/karya tulis ilmiah dari penelitian sebelumnya dan juga mengemukakan penelitian yang akan dilakukan.

BAB 3 : OBJEK DAN METODOLOGI PENELITIAN

Bab 3 objek dan metodologi penelitian membahas objek penelitian, metodologi penelitian yang digunakan dan rancangan penelitian.

BAB 4 : HASIL DAN PEMBAHASAN

Bab 4 hasil dan pembahasan berisi tentang hasil analisis perbandingan *framework Flutter* dan *React Native* dan pembahasan dari pengujian yang diperoleh.

BAB 5 : KESIMPULAN DAN SARAN

Bab 5 kesimpulan dan saran berisi tentang kesimpulan atas pengujian pada bab sebelumnya serta saran sebagai bahan perbaikan untuk penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 *Cross-platform framework*

Dimasa sekarang, hadir berbagai jenis *smartphone* dan juga tablet yang berjalan pada *platform* yang beragam, *platform* ini memiliki bahasa pemrograman, kumpulan *API* dan juga lingkungan pengembangannya masing-masing. Pengembangan aplikasi secara spesifik untuk *individual platform* membutuhkan sumber daya yang memiliki kompetensi untuk *platform* tertentu. Sebuah bisnis yang mengharapkan dapat menjangkau berbagai *platform* mengharuskan memiliki banyak sumber daya dengan kompetensi yang spesifik pada satu *platform* untuk mengembangkan aplikasi yang sama (Fentaw, 2020).

Cross-platform mobile application development merupakan sebuah pendekatan untuk mengembangkan aplikasi *mobile* dalam satu basis kode untuk masing-masing *platform*. Keuntungan utama pada pendekatan ini adalah untuk menekan biaya dan juga waktu yang dibutuhkan pada proses pengembangan aplikasi (Ebone et al., 2018). *Cross-platform development* pada *mobile* sudah ada sejak kehadiran *smartphone* itu sendiri. Perangkat *iPhone* pertama kali dirilis pada tahun 2007 dan disaat itu juga *cross-platform framework* untuk *mobile* dirilis pertama kali pada awal tahun 2008 (Kuitunen, 2019).

Cross-platform application dapat dikategorisasikan berdasarkan bagaimana hal tersebut dikembangkan, antara lain dalam hal ini adalah *Progressive Web Apps* (PWA), *hybrid apps*, *interpreterred apps*, dan *cross-compiled apps*.

2.1.1 *Progressive Web Apps (PWA)*

Sebuah *mobile web application* yang pada dasarnya adalah sebuah aplikasi web yang dikembangkan menggunakan *HTML*, *CSS*, dan *Javascript*. Dioptimalisasi untuk resolusi layar *smartphone* dan *tablet*. Dengan meningkatnya perkembangan dan juga dukungan dari beragam *API* pada lingkungan *mobile browser*, memungkinkan untuk mengakses fitur dari perangkat seperti lokasi dan penyimpanan (Biørn-Hansen et al., 2020). *PWA* dapat ditambahkan pada *home screen* dan juga beberapa fitur seperti *offline service* dan *push notification* yang

diakses melalui *web API*. *PWA* dapat meniru fungsional dari *native* tetapi tidak berkomunikasi langsung dengan *API* dari Android atau *iOS*, semuanya terjadi didalam *browser* (Hjort, 2020).

2.1.2 *Hybrid Application*

Hybrid web application merupakan kumpulan teknologi pengembangan berbasis web, perbedaan utama antara web aplikasi murni dengan *hybrid application* adalah web aplikasi hanya dapat digunakan melalui web *browser*, sedangkan *hybrid application* dapat dipasang pada perangkat *smartphone* (Kuitunen, 2019).

Hybrid web application membawa teknologi web dan juga *native development*. hal ini memanfaatkan *browser engine* pada perangkat untuk mensinkronkan konten *HTML* ke dalam *native web containers* seperti *WebView* pada android dan *UIWebView* pada *iOS*. *Web container* ini memiliki akses kepada fungsionalitas tertentu melalui *API*. *Hybrid app* menyediakan kembali kode untuk berbagai *platform* dan menyediakan akses pada kapabilitas perangkat. Namun untuk pengalaman pengguna mungkin akan kurang menyenangkan karena kurangnya penggunaan *native UI component* dan memperlambat aplikasi karena harus memuat *web container* (Fentaw, 2020).

2.1.3 *Interpreted Application*

Interpreted application mengandalkan sebuah *bridge* untuk mengakses *API* dari *platform* tujuan. Perbedaannya dengan *hybrid application* adalah *bridge* digunakan untuk merender langsung komponen antarmuka *native* untuk *platform* yang dituju, jadi tidak lagi menggunakan *web view* dan memberikan pengalaman penggunaan yang lebih baik dibandingkan *hybrid application* (Hjort, 2020).

2.1.4 *Cross-compiled Application*

Cross-compiled application memungkinkan pengembang untuk menulis kode menggunakan bahasa pemrograman yang umum digunakan, yang kemudian dikompilasi ke dalam aplikasi *native* menggunakan *cross-compiler*. *Cross-compiler*

ini bertanggung jawab untuk menggambarkan komponen antarmuka *native* yang mana dan juga fitur *native* yang dibutuhkan (Kuitunen, 2019).

Cross-compiled application mampu memberikan pengalaman dan performa layaknya *native* karena ia memiliki akses ke semua elemen UI *native* dan juga *API*. Meskipun begitu untuk *platform* tertentu perlu dilakukan ubahan dan konfigurasi untuk mengakses kemampuan perangkat seperti audio, video, *native calendar application*, *device information*, dsb (Fentaw, 2020).

2.2 *Flutter*

Flutter adalah *framework open source* yang dikembangkan oleh *Google* dan digambarkan sebagai “UI *toolkit* untuk membangun aplikasi yang indah dan dikompilasi secara *native* untuk perangkat *mobile*, web, *desktop*, dari satu basis kode (*Beautiful Native Apps in Record Time* | *Flutter*, n.d.). *Flutter* pertama kali dirilis pada tahun 2017, namun untuk versi stabil nya dirilis pada Desember 2018. Dalam waktu yang singkat *Flutter* menjadi salah satu alternatif yang cukup populer untuk mengembangkan aplikasi *cross-platform*.

Aplikasi *Flutter* ditulis dalam bahasa *Dart*, sebuah bahasa pemrograman yang dikembangkan oleh *Google*. Dikarenakan kode dikompilasi ke dalam bahasa *native*, *Flutter* memiliki performa yang cukup bagus, karena ia tidak menggunakan komponen antarmuka *native*, melainkan *render* komponen menggunakan *rendering engine* nya sendiri (Hjort, 2020). Oleh karena itulah aplikasi yang dibangun dengan *Flutter* memiliki performa yang sama cepatnya dengan aplikasi *native*.

Flutter menyediakan fitur *hot-reload* yang merupakan salah satu fitur unggulan dalam mendorong siklus pengembangan dalam pembuatan *user interface*, penambahan fitur, dan perbaikan *bugs*. Cara kerja dari *hot-reload* adalah dengan memasukkan file kode yang telah diperbarui ke dalam *Dart Virtual Machine* (VM) yang sedang berjalan. Setelah VM memperbarui *class* dengan versi yang baru. *Flutter framework* secara otomatis akan melakukan *build* ulang seluruh *widget*, dan memungkinkan untuk menampilkan perubahan secara langsung (*Beautiful Native Apps in Record Time* | *Flutter*, n.d.).

2.2.1 *Dart*

Dart adalah sebuah bahasa pemrograman yang rilis untuk produksi pada tahun 2013, dikembangkan dan dipelihara oleh *Google*. Digunakan secara luas oleh *Google* dan terbukti kemampuannya untuk pengembangan aplikasi web berskala besar, seperti contohnya *AdWords* (Wu, 2018)

Dart menganut *type safe language*, dimana kompiler akan melakukan *static type checking* untuk memastikan apakah nilai yang ada pada variabel cocok dengan tipe data nya. Tetapi sistem *typing* pada *Dart* juga fleksibel, memungkinkan untuk menggunakan tipe data *dynamic*, yang sangat berguna pada tahap percobaan ataupun untuk kode yang membutuhkan tipe data *dynamic* (*Dart Overview* | *Dart*, n.d.).

2.2.2 *Widget*

Widget menjadi komponen yang sangat penting dalam aplikasi *Flutter*. *Widget* haruslah menarik dan responsif karena inilah yang dilihat dan dirasakan langsung oleh pengguna. *Widget* tidak hanya mengontrol dan mempengaruhi perilaku tampilan, tetapi juga merespon aksi yang dilakukan pengguna. Oleh karena itu sebuah *widget* haruslah memiliki performa yang cepat, termasuk pada proses *rendering* dan *animating* (Wu, 2018).

Widget didefinisikan dengan *state* dan juga UI aplikasi yang strukturnya seperti pohon atau disebut juga *widget tree*. Kapanpun terjadi perubahan pada *widget*, *Flutter* merender ulang tampilan antarmuka dan menerapkan perubahan dari terakhir kali *widget tree* di *render*. Perenderan ulang ini efisien karena hanya ditargetkan pada komponen yang mengalami perubahan pada *widget tree*. *Flutter* hadir dengan banyak *widget* bawaan yang tentunya juga mudah untuk dikustomisasi (Fentaw, 2020).

Semua tampilan antarmuka yang ada pada *Flutter* adalah *widget*, tidak hanya elemen struktural seperti buttons dan views, tetapi juga pada *padding*, *alignment*, dan tata letak lainnya yang memiliki karakteristik serupa didefinisikan oleh *widget*. Sebuah aplikasi *Flutter* memiliki sebuah *root widget* yang berisi widget lainnya. Properti diturunkan dari *parent widget* kepada *child widget*. (Hjort, 2020).

Widget sendiri bersifat *immutable*, artinya properti yang ada pada nya bersifat final dan hanya dapat diatur saat inisialisasi. Namun dapat juga menjadi *stateful* ataupun *stateless*. Sebuah *stateful widget* memiliki *mutable state* dimana objek dapat berubah selama masa hidup *widget*. Sedangkan *stateless widget* tidak memiliki *mutable state* oleh karenanya ia bersifat statis (Hjort, 2020).

2.3 *React Native*

React Native adalah sebuah *Javascript framework* yang dibuat oleh *Facebook* untuk mengembangkan aplikasi *mobile*. Berbasiskan *React*, sebuah *Javascript library* untuk membangun antarmuka pengguna yang juga dibuat oleh *Facebook*. Sedangkan *React* digunakan untuk membuat antarmuka pada web, *React Native* ditargetkan untuk *mobile platform* dengan mengkompilasi kode untuk masing-masing *platform*.

Proses kompilasi kode ini dilakukan oleh sesuatu yang disebut dengan *bridge*. Sebuah rangkaian *Javascript* yang mengirim pesan *asynchronous JSON* yang berisi instruksi untuk menjalankan perintah *native*. Pesan ini kemudian diuraikan oleh sebuah compiler yang memproduksi *native code* untuk masing-masing *platform*. Pada dasarnya *bridge* adalah sebuah antrian pesan dimana penerima akan mengeksekusi perintah ketika memungkinkan dan dengan demikian dapat mempertahankan *frame rate* 60 *fps* pada perangkat (Tysk & Kling, 2021).

2.3.1 *Javascript*

Javascript dikembangkan oleh *Netscape* pada tahun 1995 dan sekarang menjadi salah satu bahasa pemrograman yang paling banyak digunakan pada aplikasi web. Hampir semua web *browser* pada perangkat *desktop*, *tablet*, dan *smartphone* menjalankan *Javascript*. *Javascript* menganut *dynamic language* yang mana variabel tidak harus dideklarasikan tipe data nya, melainkan didefinisikan dari nilai yang ada pada variabel. Di padukan dengan *HTML* dan *CSS* ketiganya menjadi *tools* utama bagi *web developer* untuk mengembangkan aplikasi web. Sejak kemunculannya sampai sekarang komunitas *open source* membuat *library* dan juga *framework*, yang memungkinkan *Javascript* agar dapat digunakan lebih efisien dan memperluas fungsionalitas dari *Javascript* itu sendiri (Tysk & Kling, 2021).

2.3.2 JSX

JSX adalah sebuah ekstensi sintaks *javascript* yang digunakan oleh *React* untuk menggambarkan tampilan antarmuka (*Introducing JSX – React*, n.d.). Sintaks *JSX* yang digunakan *React* mirip seperti *XML/HTML*, sintaks ini nantinya akan digunakan oleh *preprocessor* (contohnya, *transpiler* seperti Babel) untuk kemudian teks seperti *HTML* yang ada pada *Javascript* ini di transformasikan ke dalam *standard Javascript objects* (*JSX - What Is a JSX? & Introduction to Advanced*, n.d.).

Pada dasarnya dengan menggunakan *JSX* Anda dapat menuliskan struktur kode seperti *HTML/XML* pada file yang sama dengan kode *javascript*, kemudian Babel mentransformasikan *expressions* ke dalam kode *Javascript* sesungguhnya (*JSX - What Is a JSX? & Introduction to Advanced*, n.d.). Dapat dilihat pada **Tabel 2.1**.

Tabel 2.1 Perbandingan *JSX* dan *Javascript*

<i>JSX</i>	<pre> var nav = (<ul id="nav"> Home About Contact Us); </pre>
<i>Javascript</i>	<pre> var nav = React.createElement("ul", { id: "nav" }, React.createElement("li", null, React.createElement("a", { href: "#" }, "Home")), React.createElement("li", null, React.createElement("a", { href: "#" }, "About"))); </pre>

	<pre>)), React.createElement("li", null, React.createElement("a", { href: "#" }, "Contact Us"))); </pre>
--	---

2.3.3 Component

Dalam *React Native*, *component* biasa digunakan untuk membangun tampilan antarmuka seperti *Button* ataupun *TextField*, sama seperti ketika membangun tampilan antarmuka web dengan *HTML*. *Component* ini akan diterjemahkan ke dalam *native element* untuk masing-masing sistem operasi. Setiap *component* yang ada pada *React Native* dapat diberikan banyak properti sekaligus, atau yang lebih sering dikenal sebagai *props* digunakan untuk menspesifikasikan perilaku dari *component* atau untuk mengirimkan data tambahan. *Component* juga memiliki sebuah *local state* di dalamnya, dimana ia tidak dapat dimodifikasi dari luar. Perubahan yang terjadi pada *state* atau *props* akan memicu *component* untuk di *render* ulang (Tysk & Kling, 2021).

2.4 Rapid Application Development (RAD)

RAD (*Rapid Application Development*) adalah metode gabungan dari beberapa pendekatan terstruktur. Pendekatan metode RAD memakai pendekatan iteratif untuk mengembangkan suatu sistem, dimana model kerja sistem dibangun pada awal tahap pengembangan dengan tujuan untuk mengidentifikasi kebutuhan pengguna. Metode RAD memfokuskan pada ruang lingkup pemodelan bisnis, pemodelan data, pemodelan proses, pembuatan aplikasi, dan pengujian (Karim & Santoso, 2019). Metode ini telah banyak digunakan di jurnal penelitian, dikarenakan kecepatannya dalam membangun sistem dan aplikasi *prototype* tidak memakan banyak waktu, sehingga tujuan utamanya dapat dicapai secepat mungkin. Karena waktu pengembangan yang singkat pada metode RAD, pengembang

biasanya selalu berkomunikasi secara intensif dengan *client* untuk menganalisis kebutuhan yang dapat beradaptasi dengan situasi atau kondisi yang berubah-ubah (Martin & Tanaamah, 2018).

Ada 4 tahapan yang harus dilalui ketika menggunakan metode RAD, adapun diantaranya adalah perencanaan kebutuhan, desain sistem, proses pengembangan dan implementasi.

Kelebihan dan kekurangan metode RAD ini menurut Maikel Bolung dalam (Murdiani & Hermawan, 2022) adalah sebagai berikut:

Kelebihan *Rapid Application Development* (RAD) adalah:

1. Waktu pengembangan yang singkat
2. Bisa dilakukan perubahan apabila diperlukan oleh *user*
3. Penggunaan *tools* yang bisa digunakan kembali untuk pengerjaan yang lebih cepat
4. Hanya memerlukan sedikit tim pengembang

Kekurangan *Rapid Application Development* (RAD) adalah:

1. Perlu adanya manajemen tim yang baik
2. Dalam penyusunan sebuah sistem keterlibatan *user* sangat diperlukan
3. Membutuhkan seorang yang profesional dan berpengalaman dalam penggunaan metode ini
4. Kemampuan pada *prototype* sangat bergantung untuk pengembangan metode ini

2.5 *Android Package Kit* (APK)

Android Package Kit (APK) adalah sebuah format file yang digunakan untuk mendistribusikan dan menginstalasi aplikasi *android*. APK adalah sebuah arsip file yang di dalamnya berisi beragam file dan beberapa *metadata* mengenai aplikasi tersebut. Seorang *developer* yang mengembangkan aplikasi *mobile* harus mengkompilasi aplikasi tersebut kedalam format APK untuk dapat mengunggahnya di *Google Play*. File APK juga dapat langsung di distribusikan ke pengguna android dengan mengizinkan untuk pemasangan aplikasi tidak dikenal (ben Stegner, 2022).

2.6 Kompleksitas Kode

Kompleksitas kode menurut IEEE mendefinisikan kompleksitas sebagai sejauh mana desain atau kode sistem sulit untuk dipahami karena banyak komponen atau hubungan antar komponen atau sejauh mana sistem atau komponen memiliki desain atau implementasi yang sulit untuk dipahami dan diverifikasi (Hourani et al., 2019). Pengukuran kompleksitas kode dapat memfasilitasi dan memudahkan pemeliharaan, ekstensibilitas, keterbacaan, konsistensi, dan pemahaman kode pemrograman. Oleh karena itu, metrik kompleksitas kode dianggap sebagai pendorong kualitas utama untuk setiap pengembang perangkat lunak. Mengukur kompleksitas kode akan menyusun bagaimana pengembang mendesain kode mereka dan bagaimana membuatnya tidak terlalu rumit.

Metode pengukuran kompleksitas kode tradisional seperti *Cyclomatic Complexity*, digunakan untuk mengevaluasi kompleksitas kode suatu algoritma dalam suatu *method*. *Cyclomatic Complexity* mengukur jumlah jalur independen linier melalui kode fungsi atau *method*. Ini adalah salah satu metode pengukuran yang paling banyak digunakan di antara metode lainnya

2.7 Central Processing Unit (CPU)

Central Processing Unit (CPU) adalah komponen sistem komputer yang menjalankan instruksi program komputer, dan merupakan elemen utama yang menjalankan fungsi komputer. CPU membawa serangkaian instruksi program, menjalankan fungsi logika dan aritmatika, dan menangani operasi input/output dari sistem (Guadaña et al., 2013). Agar dapat berjalan dengan baik, sebuah CPU memiliki komponen pendukung didalamnya. Berikut adalah komponen tersebut diantaranya :

a. Control Unit (CU)

Komponen ini merupakan bagian dari prosesor yang bertugas untuk mengatur jalannya program. Tugas *control unit* adalah mengambil instruksi dari memori utama dan menentukan jenis instruksi yang ada. Pengolahan data yang dilakukan oleh *control unit* kemudian dibawa ke memori utama untuk disimpan. Kemudian ketika saatnya tiba, data akan

disajikan melalui perangkat output. *Control unit* memiliki kontrol penuh atas komputer, memungkinkan sinkronisasi antara komponen untuk dapat melakukan setiap fungsinya.

b. Register

Register adalah perangkat penyimpanan yang memiliki ukuran kecil tapi cukup cepat. Fungsi register adalah untuk menyimpan data atau instruksi yang sedang diproses. Memori ini bersifat sementara, sehingga hanya menyimpan data yang sedang diproses atau akan digunakan untuk pemrosesan lebih lanjut.

c. *Aritmathic Logic Unit* (ALU)

Aritmathic Logic Unit (ALU) adalah bagian dari CPU yang tugasnya melakukan operasi aritmatika dan operasi logika sesuai dengan instruksi. ALU terdiri dari dua bagian: unit operasi dan unit logika. Unit logika memiliki spesifikasi tugas sendiri. Tugas utama ALU adalah melakukan semua perhitungan aritmatika sesuai dengan instruksi program.

2.8 *Random Access Memory* (RAM)

Random Access Memory (RAM) komputer pada dasarnya adalah memori jangka pendek di mana data disimpan sesuai kebutuhan prosesor. Ini berbeda dengan data jangka panjang yang disimpan pada hard drive, yang tetap ada bahkan ketika komputer dimatikan. Ketika komputer dihidupkan data dan perintah dari hard drive disimpan pada RAM. CPU menggunakan data ini untuk melakukan tugas yang diperlukan. Segera setelah komputer dimatikan, RAM kehilangan data. Jadi, data tetap berada di RAM selama komputer menyala dan hilang saat komputer dimatikan. Manfaat memuat data ke dalam RAM adalah membaca data dari RAM jauh lebih cepat daripada membaca dari hard drive (*What Is RAM | Random Access Memory - Javatpoint*, n.d.).

2.9 *Frame Rate*

Frame Rate adalah frekuensi atau kecepatan gambar berurutan (*frame*) ditangkap atau ditampilkan. Dinyatakan dengan satuan *frame per second* (fps)

untuk *frame* yang direkam atau ditampilkan dalam satu detik. *Frame rate* berlaku pada komputer, kamera, video dan sistem penangkap gerak. *Frame rate* juga disebut frekuensi *frame* dan dinyatakan dengan Hertz (Hz) (*Frame Rate - Everything You Need to Know - NFI*, n.d.).

2.10 Penelitian Sebelumnya

Tabel 2.2 Penelitian sebelumnya

Penelitian ke-1	
Judul	<i>Cross-platform Framework Comparison Flutter & React Native</i>
Tahun	2020
<i>Authors</i>	Simon Stender & Hampus Akesson
Ringkasan	Penelitian ini dilakukan untuk menguji kedua framework yaitu <i>Flutter</i> dan <i>React Native</i> , pengujian dilakukan dengan membuat dua buah aplikasi yang serupa pada <i>platform</i> android, alasan tidak membuat pada <i>platform</i> iOS dikarenakan keterbatasan sumber daya penulis. Pengujian dilakukan menggunakan <i>Android Profiler</i> dan perangkat android. Pengukuran yang dilakukan pada pengujian ini antara lain penggunaan CPU, waktu eksekusi, dan ukuran aplikasi. Selain itu penelitian ini melakukan survei terhadap pendapat <i>developer</i> tentang <i>framework Flutter</i> dan <i>React Native</i> .
Kesimpulan	Hasilnya dapat disimpulkan bahwa performa yang diberikan oleh <i>Flutter</i> lebih baik dibandingkan dengan <i>React Native</i> . Namun perbedaan ini tidaklah signifikan, karena secara keseluruhan keduanya cukup mirip. Kemudian untuk survei yang dilakukan kepada <i>developer</i> hasilnya <i>React Native</i> memberikan persentase yang lebih tinggi dibandingkan <i>Flutter</i> , hal ini dapat terjadi karena <i>React Native</i> sudah ada sejak lama sebelum kehadiran

	<i>Flutter</i> dan cukup dikenal dan banyak digunakan di kalangan <i>developer</i> .
Penelitian ke-2	
Judul	<i>Cross Platform Mobile Application Development: A Comparison Study of React Native Vs Flutter</i>
Tahun	2020
<i>Authors</i>	Awel Eshetu Fentaw
Ringkasan	Tujuan utama pada penelitian ini adalah menyediakan studi perbandingan diantara dua aplikasi pengembangan <i>cross-platform</i> yang paling banyak digunakan yaitu <i>Flutter</i> dan <i>React Native</i> . Penelitian ini mengarahkan untuk mengetahui penggunaan CPU, GPU dan memori pada aplikasi yang dikembangkan dengan <i>React Native</i> dan <i>Flutter</i> yang berjalan pada <i>platform</i> iOS dan android.
Kesimpulan	<i>React Native</i> secara konsisten menggunakan CPU lebih banyak dibandingkan <i>Flutter</i> dalam semua kasus pengujian. Alasan terjadi nya hal ini adalah karena <i>React Native</i> menggunakan <i>Javascript bridge</i> untuk melakukan komunikasi dengan modul <i>native</i> . Hasil pengujian penggunaan memori, <i>React Native</i> memberikan performa yang lebih baik dibandingkan <i>Flutter</i> . Hal ini terjadi karena aplikasi <i>React Native</i> mengimplementasi <i>library</i> <i>Redux</i> yang berguna sebagai <i>state management</i> . Terakhir untuk pengujian GPU, secara umum selama pengujian <i>Scrolling cards</i> dan <i>Opening webviews</i> kedua nya memberikan hasil yang serupa. Disisi lain <i>React Native</i> memberikan penggunaan GPU yang lebih baik saat pengetesan <i>Rendering listview</i> . Dan <i>Flutter</i> memberikan penggunaan GPU yang lebih baik pada pengetesan <i>Filtering</i> .

Penelitian ke-3	
Judul	<i>Evaluation of React Native and Flutter For Cross-Platform Mobile Application Development</i>
Tahun	2020
<i>Authors</i>	Elin Hjort
Ringkasan	Penelitian ini mengevaluasi <i>React Native</i> dan <i>Flutter</i> , sebagai <i>cross-platform framework</i> paling modern, untuk menentukan manakah yang paling tepat untuk menjadi langkah awal dalam pengembangan aplikasi <i>mobile</i> . Evaluasi dilakukan pada pengembangan untuk aplikasi android dan iOS, mengingat kedua sistem operasi ini yang paling digunakan pada <i>platform mobile</i> .
Kesimpulan	Evaluasi dilakukan berdasarkan kriteria yang telah ditentukan dari kebutuhan proses pengembangan aplikasi <i>mobile</i> . Hasilnya <i>React Native</i> adalah <i>framework</i> yang paling tepat. <i>React Native</i> mendapat skor tertinggi pada banyak kriteria, oleh karena itu dinilai paling tepat untuk berbagai jenis aplikasi. Disisi lain <i>Flutter</i> menjadi pilihan yang baik untuk sebagian aplikasi yang mengedepankan performa dan mengesampingkan tampilan dan pengalaman <i>native</i> .
Penelitian ke-4	
Judul	<i>Cross-Platform Mobile Application Development with React Native</i>
Tahun	2018
<i>Authors</i>	Mika Kuitunen
Ringkasan	Penelitian ini berfokus pada teknologi pengembangan aplikasi <i>mobile</i> yang bersifat <i>cross-platform</i> . Yang secara spesifik mampu memberikan fungsionalitas dan pengalaman serupa seperti aplikasi <i>native</i> untuk masing-

	masing <i>platform</i> . Tujuan penelitian ini adalah untuk mengetahui apakah teknologi <i>cross-platform</i> dapat menjadi pilihan yang layak untuk pengembangan aplikasi modern.
Kesimpulan	Secara keseluruhan hasil dari penelitian ini menyebutkan bahwa <i>React Native</i> cukup menjanjikan, untuk menjadi pilihan yang layak dalam pengembangan aplikasi <i>mobile</i> . Mengembangkan aplikasi menggunakan teknologi <i>cross-platform</i> membuat proses pengembangan lebih mudah dan lebih singkat dan juga memudahkan perawatan aplikasi.
Penelitian ke-5	
Judul	<i>Flutter Native Performance and Expressive UI/UX</i>
Tahun	2020
<i>Authors</i>	Thanh Tran
Ringkasan	Penelitian ini mengeksplorasi teknologi yang ada di belakang <i>Flutter</i> dalam pengembangan aplikasi performa tinggi dan antarmuka dinamis. Tujuan utama penelitian ini untuk membandingkan tampilan dan performa aplikasi <i>Flutter</i> dengan teknologi pengembangan aplikasi yang sudah ada. Proyek yang ada pada penelitian ini merupakan kolaborasi dengan Blidz, sebuah perusahaan <i>e-commerce</i> yang ingin melakukan peningkatan dengan migrasi dari Cordova ke <i>Flutter</i> . Namun sebelum itu perlu dilakukan pengujian untuk membandingkan aplikasi yang dibuat dengan <i>Flutter</i> dan juga aplikasi dengan Cordova yang sudah ada.
Kesimpulan	Selama proses pengembangan yang sesungguhnya dan migrasi aplikasi pada level produksi, <i>Flutter</i> memberikan dampak positif selama masa pengembangan. <i>Flutter</i>

	<p>lebih cepat saat di <i>debug</i>, lebih cepat untuk dikembangkan, dan lebih cepat untuk di tes. <i>Flutter</i> memiliki keunggulan dibandingkan <i>Cordova</i>, karena penggunaan sumber daya lebih di optimal, <i>frame rate</i> konsisten pada 60fps dan tetap mempertahankan konsumsi energi yang baik. Animasi yang ditampilkan juga sangat halus. Lebih dari itu dukungan <i>widget</i> yang luas dari <i>Flutter</i> SDK memungkinkan untuk mendesain aplikasi lebih leluasa.</p>
--	--

2.11 Penelitian Sekarang

Penelitian yang akan dilakukan saat ini adalah analisis perbandingan penggunaan *framework Flutter* dan *React Native* untuk pengembangan aplikasi *mobile* yang bersifat *cross-platform* dengan studi kasus Sistem Informasi Tugas Akhir, yang data nya diperoleh melalui data *local json*. Untuk melakukan uji perbandingan diantara kedua *framework* tersebut dilakukan beberapa pengujian diantaranya membandingkan ukuran aplikasi, kompleksitas kode, dan performa (CPU, memori, *frame rate*) menggunakan *Android Profiler* bawaan *Android Studio*.

BAB III

OBJEK DAN METODOLOGI PENELITIAN

3.1 Objek Penelitian

Penelitian ini akan menganalisis perbandingan *framework Flutter* dan *React Native* dalam pengembangan aplikasi *mobile* yang bersifat *cross-platform*. Objek yang menjadi bahan pengujian pada penelitian ini yaitu mengenai perbandingan ukuran aplikasi, kompleksitas kode, dan performa diantaranya penggunaan CPU, penggunaan memori, dan *frame rate*.

3.2 Metodologi Penelitian

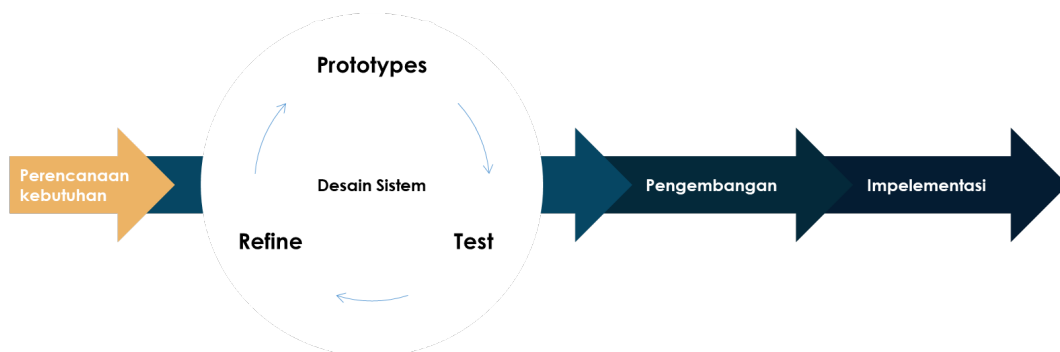
Metodologi pada penelitian ini menggunakan metode *Rapid Application Development* (RAD) untuk metode pengembangan aplikasi nya. Tahapan dari metodologi tersebut yaitu perencanaan kebutuhan, desain sistem, proses pengembangan, dan implementas. Kemudian, untuk metodologi pengujian aplikasi tahapan nya antara lain penentuan objek yang akan diuji, penentuan parameter pengujian, penentuan skenario pengujian, tahap pengujian, analisis dan kesimpulan.

3.3 Rancangan Penelitian

Adapun untuk rancangan penelitian ini dibagi menjadi dua hal yaitu:

3.3.1 Rancangan Pengembangan Aplikasi

Rancangan pengembangan aplikasi yang akan dilaksanakan terdiri dari beberapa tahapan yaitu perencanaan kebutuhan, desain sistem, proses pengembangan, dan implementas.



Gambar 3.1 Diagram *Rapid Application Development* (RAD)

3.3.1.1 Perencanaan Kebutuhan

Tahap ini merupakan tahap awal dalam pengembangan sistem, dimana pada tahap ini akan dilakukan identifikasi masalah dan mengumpulkan data yang didapat dari pengguna atau *stakeholder* pengguna yang memiliki tujuan untuk mengidentifikasi tujuan akhir dari sistem yang diinginkan.

Studi kasus yang digunakan pada penelitian ini adalah Sistem Informasi Tugas Akhir dimana sistem ini dapat menjadi bank data dari mahasiswa yang sedang melaksanakan tugas akhir. Studi kasus ini menjadi bahan untuk pengujian diantara kedua *framework Flutter* dan *React Native*, yang mana akan membandingkan kompleksitas kode, ukuran aplikasi, dan performa.

3.3.1.2 Desain Sistem

Pada tahap ini akan dibuat sebuah prototipe secara cepat dengan fitur-fitur dan fungsi yang dibutuhkan, setelah ini prototipe akan dievaluasi apakah sudah sesuai dengan kebutuhan *software* yang ingin dikembangkan. Pada tahap ini tentu hasil yang diberikan belum sempurna sama sekali, diperlukan proses desain dan perbaikan desain secara berulang-ulang apabila masih ada ketidaksesuaian antara desain dengan kebutuhan pengguna yang telah didefinisikan sebelumnya.

3.3.1.3 Proses Pengembangan

Dalam tahap ini dilakukan pengkodean untuk mengembangkan fitur, fungsi, dan tampilan antarmuka dengan tujuan untuk mengimplementasikan hasil perancangan menjadi sebuah sistem dari desain sistem yang telah disetujui pada tahap sebelumnya.

3.3.1.4 Implementasi

Tahap ini merupakan tahapan dimana pengembang mengimplementasikan desain dari sistem yang telah disetujui pada tahap sebelumnya. Sebelum sistem diterapkan, sistem terlebih dulu dilakukan proses pengujian terhadap program untuk mendeteksi kesalahan yang terdapat pada sistem yang dikembangkan.

3.3.2 Rancangan Pengujian Aplikasi

Untuk melakukan pengujian diantara kedua *framework Flutter* dan *React Native* dilakukan beberapa skema pengujian diantaranya adalah:

3.3.2.1 Analisis Objek Pengujian

Pada tahap ini yaitu menentukan objek yang akan diuji, dalam hal ini adalah *framework Flutter* dan *React Native* untuk pengembangan aplikasi *mobile* bersifat *cross-platform*. Selain itu untuk objek yang menjadi studi kasus nya yaitu sistem informasi tugas akhir

3.3.2.2 Penentuan Skenario Pengujian

Pada tahap ini yaitu menentukan skenario pengujian aplikasi, skenario tersebut diperlukan untuk mengidentifikasi bagaimana pengujian pada aplikasi dilakukan.

3.3.2.3 Tahap Pengujian

Pada tahap ini dilakukan pengujian perbandingan *framework Flutter* dan *React Native* dengan melaksanakan skenario yang telah didefinisikan sebelumnya yaitu membandingkan ukuran aplikasi, kompleksitas kode, dan performa.

a. Ukuran Aplikasi

Pada tahap ini kode program yang telah selesai ditulis menggunakan *Flutter* dan *React Native* dilakukan *build* menjadi aplikasi (*apk*). Pengujian yang dilakukan adalah dengan membandingkan ukuran penyimpanan hasil jadi aplikasinya.

b. Kompleksitas Kode

Pada tahap pengujian kompleksitas kode dibuat sebuah fungsi untuk menjalankan tugas tertentu yang ditulis menggunakan bahasa *Dart* dan *Javascript* kemudian dibandingkan manakah yang secara struktur kode lebih sederhana atau lebih kompleks. Proses pengujian kompleksitas kode ini menggunakan metode *Cyclomatic Complexity*

c. Performa

Pada tahap pengujian performa terdapat beberapa aspek pengujian, diantaranya pengujian penggunaan CPU, penggunaan memori, dan *framerate*. Pengujian ini dilakukan menggunakan *tools* bawaan Android Studio yaitu *Android Profiler*

3.3.2.4 Kesimpulan

Pada tahap ini dapat ditarik sebuah kesimpulan dari hasil analisis perbandingan yang dilakukan diantara kedua framework Flutter dan React Native, dan juga menjadi hasil dari penelitian yang dilakukan

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Hasil penelitian ini adalah membandingkan dua *framework* pengembangan aplikasi *mobile* yang bersifat *cross-platform* yaitu *framework Flutter* dan *React Native*. Studi kasus aplikasi yang digunakan pada penelitian ini adalah sebuah sistem informasi tugas akhir yang fungsinya sebagai bank data bagi mahasiswa yang sedang menyelesaikan tugas akhir.

4.1.1 Pengembangan Aplikasi

Berikut ini adalah hasil dari tahapan pengembangan aplikasi dengan menerapkan metode pengembangan *Rapid Application Development (RAD)* dengan 4 tahapannya yaitu perencanaan kebutuhan, desain sistem, proses pengembangan dan implementasi

4.1.1.1 Perencanaan Kebutuhan

Tahap perencanaan kebutuhan yang dilakukan bertujuan untuk menganalisis apa saja yang dibutuhkan untuk mengembangkan aplikasi sistem informasi tugas akhir, analisis tersebut antara lain analisis kebutuhan fungsional dan analisis kebutuhan non fungsional

1. Analisis Kebutuhan Fungsional

Aplikasi Sistem Informasi Tugas Akhir adalah aplikasi yang digunakan untuk melihat data-data dari mahasiswa yang sedang melaksanakan tugas akhir, mulai dari biodata mahasiswa, status tugas akhir, judul proposal, nama dosen pembimbing, judul skripsi, nama dosen penguji, dll. Pengembangan aplikasi ini menggunakan *Flutter* dan *React Native* secara bersamaan. Proses akuisisi pengetahuan untuk tahap analisis kebutuhan fungsional dilakukan dengan metode observasi, adapun hasil analisis kebutuhan nya adalah sebagai berikut:

Tabel 4. 1 Analisis kebutuhan fungsional

Aktor	Kebutuhan Fungsional	Deskripsi
Mahasiswa	Lihat daftar mahasiswa	Menampilkan daftar nama-nama mahasiswa dan npm

	Pencarian mahasiswa	Mencari data mahasiswa pada kolom pencarian berdasarkan nama mahasiswa
	Lihat detail mahasiswa	Menampilkan data detail dari mahasiswa seperti nama, npm, email, dosen pembimbing dan judul proposal

2. Analisis Kebutuhan Non Fungsional

Pada tahap ini dilakukan analisis kebutuhan perangkat yang digunakan pada proses pengembangan aplikasi sistem informasi manajemen tugas akhir. Perangkat yang dimaksud pada penelitian ini yaitu perangkat untuk kebutuhan pengembangan dan perangkat lingkungan. Penggunaan perangkat yang berbeda mungkin saja dapat memberikan hasil yang berbeda pula. Berikut ini adalah perangkat yang digunakan:

Tabel 4. 2 Analisis kebutuhan *non fungsional*

No.	Nama Perangkat	Spesifikasi	Jenis
1	Asus X505ZA	<ul style="list-style-type: none"> Prosesor: AMD Ryzen 3 2200U RAM: 12 GB Penyimpanan: 256 GB SSD 	Perangkat keras
2	Sistem Operasi	Windows 11 Home	Perangkat lunak
3	<i>Flutter</i>	Versi 2.8.1	Perangkat lunak
4	<i>React Native</i>	Versi 0.68.1	Perangkat lunak
5	Android Studio	Versi 2021.2.1 Patch 1	Perangkat lunak
6	Visual Studio Code	Versi 1.70.0	Perangkat lunak

Tabel 4. 3 Analisis kebutuhan lingkungan

No.	Nama Perangkat	Spesifikasi	Jenis
1	Redmi Note 10 Pro	<ul style="list-style-type: none"> • Prosesor: Snapdragon 732G • RAM: 8 GB • Penyimpanan: 128 GB 	Perangkat keras
2	Sistem Operasi	Android 12	Perangkat lunak

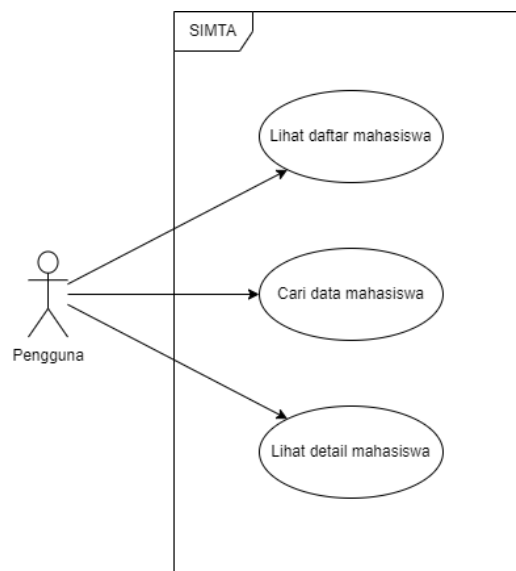
4.1.1.2 Desain Sistem

Pada tahap analisis desain sistem digunakan metode *Unified Modelling Language* (UML) yang bertujuan agar mudah memahami, menganalisis, dan mempermudah pengembangan suatu aplikasi dengan mengimplementasikan rancang bangun ke dalam sebuah diagram visual. Selain itu dilakukan proses desain *interface* untuk merancang tampilan aplikasi.

a. Unified Modelling Language (UML)

1) Use case diagram

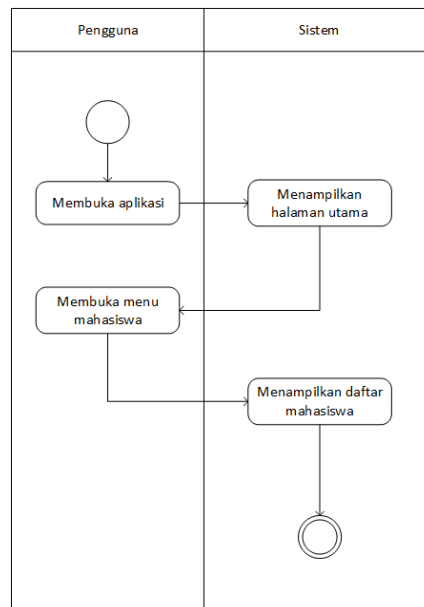
Use case diagram digunakan untuk menggambarkan interaksi yang dapat dilakukan pengguna dengan sistem aplikasi

**Gambar 4. 1** Use Case Diagram

2) *Activity diagram*

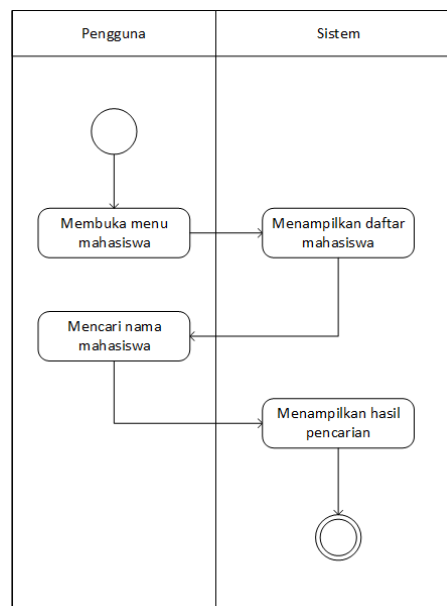
Activity diagram merupakan diagram untuk memodelkan proses yang terjadi pada sistem secara runtut dan divisualisasikan secara vertikal

a) Lihat daftar mahasiswa



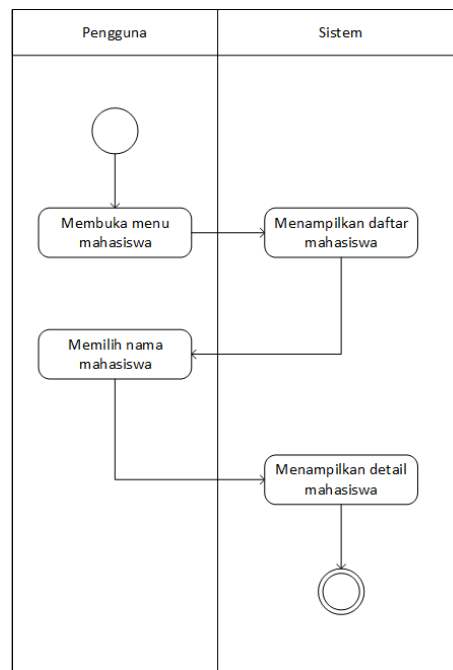
Gambar 4. 2 Activity Diagram Lihat Daftar Mahasiswa

b) Cari data mahasiswa



Gambar 4. 3 Activity Diagram Cari Data Mahasiswa

c) Lihat detail mahasiswa



Gambar 4. 4 Activity Diagram Lihat Detail Mahasiswa

b. Desain Interface

Terdapat 4 buah halaman pada aplikasi diantaranya yaitu, halaman utama, halaman daftar mahasiswa, halaman detail mahasiswa, dan halaman *not found*. Berikut adalah penjelasan mengenai kegunaan dari halaman yang ada pada aplikasi:

1) Halaman utama (*Dashboard* aplikasi)

Halaman ini menjadi tampilan awal ketika membuka aplikasi, pada halaman ini menampilkan data statistik dari mahasiswa, dosen, dan tugas akhir

2) Halaman daftar mahasiswa

Halaman ini menampilkan daftar mahasiswa yang sedang melaksanakan tugas akhir. Data ini diambil dari dari *file JSON* yang disimpan didalam folder proyek

3) Halaman detail mahasiswa

Halaman ini menyajikan detail informasi mahasiswa mulai dari biodata mahasiswa, status tugas akhir, judul proposal, nama dosen pembimbing, judul skripsi, nama dosen penguji, dll..

4) Halaman *not found*

Halaman ini digunakan sebagai halaman *default* ketika *user* membuka menu yang tidak tersedia

Selain itu desain aplikasi dibuat ke dalam dua versi yaitu untuk aplikasi yang di bangun menggunakan *framework Flutter* dan *framework React Native*.

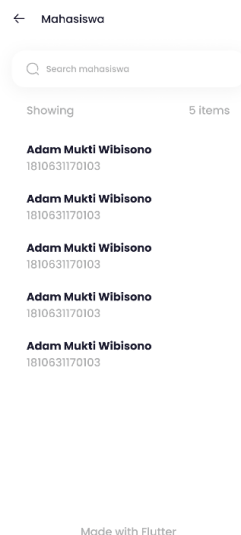
1) *Flutter*

a) Halaman utama



Gambar 4. 5 Desain Halaman Utama (*Flutter*)

b) Halaman daftar mahasiswa

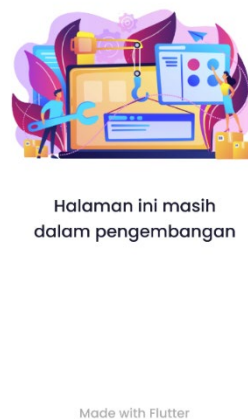


Gambar 4. 6 Desain Halaman Daftar Mahasiswa (*Flutter*)

c) Halaman detail mahasiswa



Gambar 4. 7 Desain Halaman Detail Mahasiswa (*Flutter*)

d) Halaman *not found*

Gambar 4. 8 Desain Halaman *Not Found* (*Flutter*)

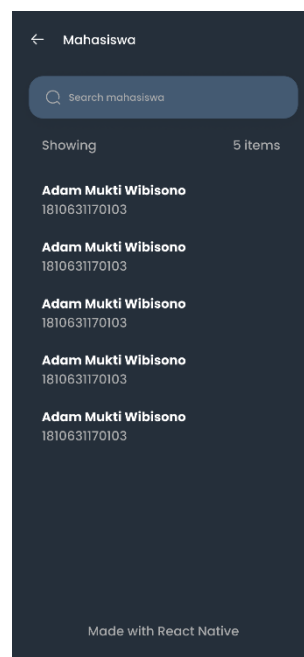
2) *React Native*

a. Halaman utama



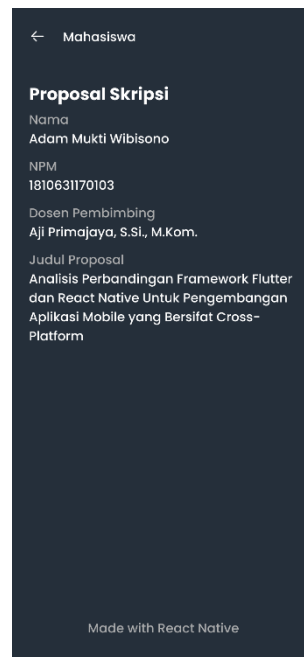
Gambar 4. 9 Desain Halaman Utama

b. Halaman daftar mahasiswa

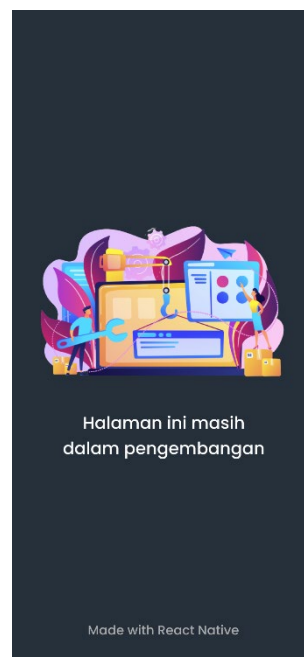


Gambar 4. 10 Desain Halaman Daftar Mahasiswa (*React Native*)

c. Halaman detail mahasiswa

**Gambar 4. 11** Desain Halaman Detail Mahasiswa (*React Native*)

d. Halaman not found

**Gambar 4. 12** Desain Halaman Not Found (*React Native*)

4.1.1.3 Proses Pengembangan

Pada tahap pengembangan, desain yang telah dibuat pada tahap sebelumnya di implementasikan ke dalam aplikasi yang sesungguhnya. Aplikasi tersebut dikembangkan menggunakan *framework Flutter* dengan bahasa pemrogramannya yaitu *Dart* dan juga *framework React Native* dengan bahasa pemrograman *Javascript*. Berikut adalah hasil dari pengembangan aplikasi tersebut:

a. *Flutter*

1) Halaman utama



Gambar 4. 13 Halaman Utama (*Flutter*)

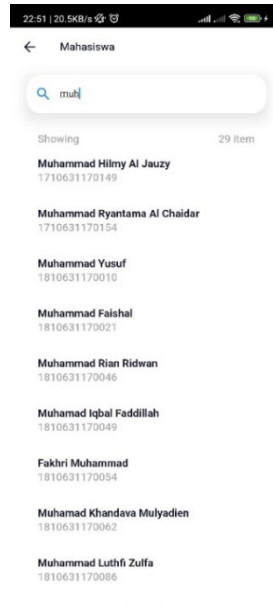
2) Halaman daftar mahasiswa



Gambar 4. 14 Halaman Daftar Mahasiswa (*Flutter*)

3) Halaman daftar mahasiswa (pencarian)

Halaman daftar mahasiswa ketika melakukan pencarian dengan memasukkan nama pada *search bar*



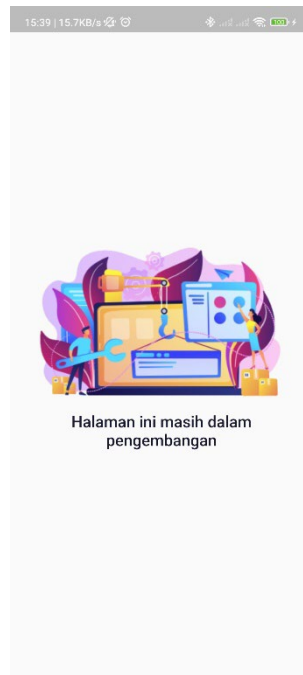
Gambar 4. 15 Halaman Daftar Mahasiswa (Pencarian) (*Flutter*)

4) Halaman detail mahasiswa



Gambar 4. 16 Halaman Detail Mahasiswa (*Flutter*)

5) Halaman *not found*



Gambar 4. 17 Halaman *Not Found* (*Flutter*)

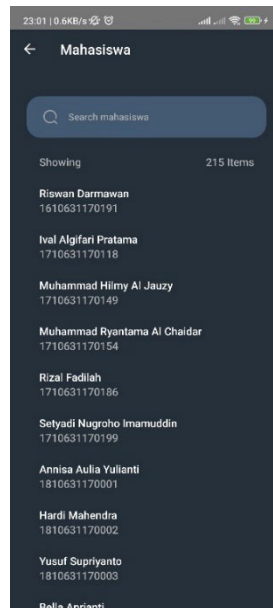
b. *React Native*

1) Halaman utama



Gambar 4. 18 Halaman Utama (*React Native*)

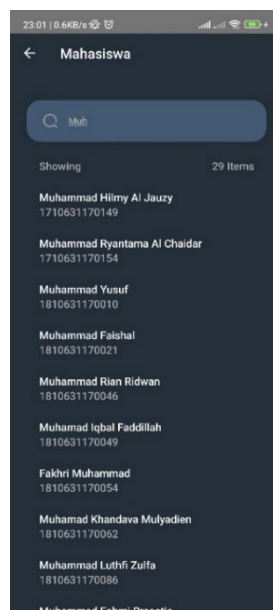
2) Halaman daftar mahasiswa



Gambar 4. 19 Halaman Daftar Mahasiswa (*React Native*)

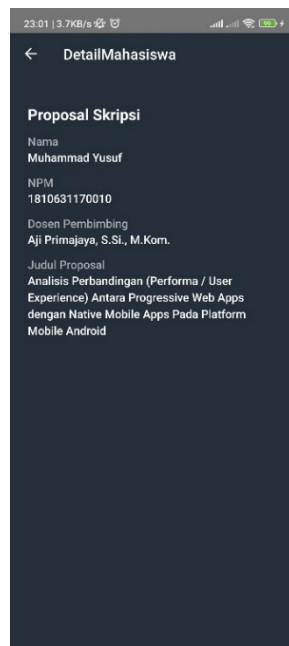
3) Halaman daftar mahasiswa (pencarian)

Halaman daftar mahasiswa ketika melakukan pencarian dengan memasukkan nama pada *search bar*



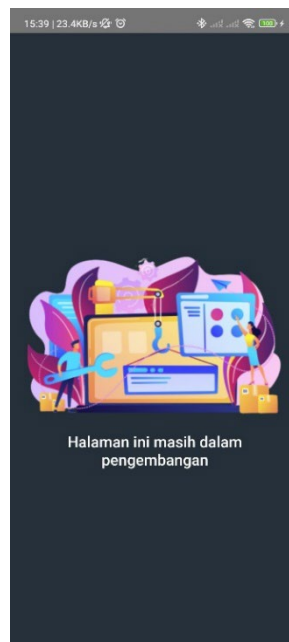
Gambar 4. 20 Halaman Daftar Mahasiswa (Pencarian) (*React Native*)

4) Halaman detail mahasiswa



Gambar 4. 21 Halaman Detail Mahasiswa

5) Halaman *not found*



Gambar 4. 22 Halaman Not Found (*React Native*)

4.1.1.4 Implementasi

Dalam tahap implementasi ini dilakukan pengujian untuk melakukan uji coba terhadap aplikasi dan juga fungsi-fungsi yang terdapat di dalamnya, untuk

memastikan bahwa aplikasi dapat berjalan dengan baik. Pengujian ini dilakukan dengan metode *black-box testing*

Tabel 4. 4 *Black Box Testing*

Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Menekan <i>icon</i> aplikasi	Aplikasi menampilkan halaman utama	Sesuai harapan	Diterima
Menekan <i>banner</i> menu mahasiswa	Menampilkan daftar semua mahasiswa	Sesuai harapan	Diterima
Memasukkan huruf di kolom pencarian menu mahasiswa	Menampilkan daftar mahasiswa dengan nama tersebut	Sesuai harapan	Diterima
Menekan salah satu nama mahasiswa	Membuka halaman detail dari mahasiswa tersebut	Sesuai harapan	Diterima
Menekan menu dosen	Menampilkan halaman masih dalam tahap pengembangan	Sesuai harapan	Diterima
Menekan menu proposal	Menampilkan halaman masih dalam tahap pengembangan	Sesuai harapan	Diterima
Menekan menu kolokium	Menampilkan halaman masih dalam tahap pengembangan	Sesuai harapan	Diterima
Menekan menu yudisium	Menampilkan halaman masih dalam tahap pengembangan	Sesuai harapan	Diterima

4.1.2 Pengujian Aplikasi

Berikut ini adalah hasil dari tahap metode pengujian aplikasi dengan tahapannya yaitu analisis objek pengujian, skenario pengujian dan tahap pengujian.

4.1.2.1 Analisis Objek Pengujian

Agar memudahkan dalam memahami studi kasus, penulis mengklasifikasikan penelitian ini seperti pada **Tabel 4. 5**. Hanya beberapa fitur yang diuji pada penelitian ini karena fitur-fitur tersebut dirasa sudah cukup mewakili dalam studi kasus ini.

Tabel 4. 5 Klasifikasi Studi Kasus Pengujian

Kriteria	SIMTA
Fitur yang diuji	Menu Mahasiswa: <ul style="list-style-type: none"> - Daftar nama mahasiswa - Pencarian nama mahasiswa - Detail mahasiswa
Sumber data	<i>Local JSON file</i>
Kerangka kerja	<ul style="list-style-type: none"> - <i>Flutter (Dart)</i> - <i>React Native (Javascript)</i>
Parameter pengujian	<ul style="list-style-type: none"> - Ukuran aplikasi - Kompleksitas kode - Penggunaan CPU, konsumsi memori, <i>framerate</i>

4.1.2.2 Skenario Pengujian

Berikut ini merupakan skenario yang digunakan untuk melakukan pengujian aplikasi Flutter dan React Native

Tabel 4. 6 Skenario Pengujian

No	Skenario Pengujian
1	Klik <i>banner</i> Mahasiswa untuk menuju halaman daftar mahasiswa
2	Menggulirkan daftar mahasiswa
3	Buka detail mahasiswa
4	Kembali ke daftar mahasiswa
5	Klik kolom pencarian nama mahasiswa
6	Ketik untuk mencari nama mahasiswa

No	Skenario Pengujian
7	Buka detail mahasiswa hasil pencarian
8	Kembali ke menu awal

4.1.2.3 Tahap Pengujian

Berikut ini adalah hasil pada tahap pengujian perbandingan aplikasi Flutter dan React Native dengan parameter pengujian pada ukuran aplikasi, kompleksitas kode dan performa.

a. Ukuran Aplikasi

Pada tahap pengujian ini akan membandingkan ukuran aplikasi dari proyek yang dibuat menggunakan *Flutter* dan *React Native*, kedua aplikasi tersebut di *build* dalam mode *release* dengan kode program yang meng *import package* seperti pada pada **Tabel 4. 7**. Berikut ini hasil ukuran aplikasi (APK) yang dibangun menggunakan *framework Flutter* dan *React Native*

```

Warning: Mapping new ns http://schemas.android.com/repository/android/generic/02 to old ns http://schemas.android.com/repository/android/generic/01
Warning: Mapping new ns http://schemas.android.com/sdk/android/repo/addon2/02 to old ns http://schemas.android.com/sdk/android/repo/addon2/01
Warning: Mapping new ns http://schemas.android.com/sdk/android/repo/repository2/02 to old ns http://schemas.android.com/sdk/android/repo/repository2/01
Warning: Mapping new ns http://schemas.android.com/sdk/android/repo/sys-img2/02 to old ns http://schemas.android.com/sdk/android/repo/sys-img2/01
Parameter format not correct -
Running Gradle task 'assembleRelease'... 63.4s
✓ Built build/app/outputs/flutter-apk/app-release.apk (16.9MB).
PS C:\Users\adamm\Documents\Project Skripsi\flutter_project> cd build/app/outputs/flutter-apk
PS C:\Users\adamm\Documents\Project Skripsi\flutter_project\build\app\outputs\flutter-apk> ls -l app-release.apk

Directory: C:\Users\adamm\Documents\Project Skripsi\flutter_project\build\app\outputs\flutter-apk

Mode                LastWriteTime         Length Name
----                -
-a-----       7/15/2022   9:42 PM      17733808 app-release.apk

```

Gambar 4. 23 Hasil *Build* Apk Aplikasi *Flutter*

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  GITLENS
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\adamm\Documents\Project Skripsi\ReactNativeProject> cd android
PS C:\Users\adamm\Documents\Project Skripsi\ReactNativeProject\android> ./gradlew assembleDebug

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

See https://docs.gradle.org/7.3.3/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 1m 55s
92 actionable tasks: 17 executed, 75 up-to-date
PS C:\Users\adamm\Documents\Project Skripsi\ReactNativeProject\android> cd app/build/outputs/apk/release
PS C:\Users\adamm\Documents\Project Skripsi\ReactNativeProject\android\app\build\outputs\apk\release> ls -l app-release.apk

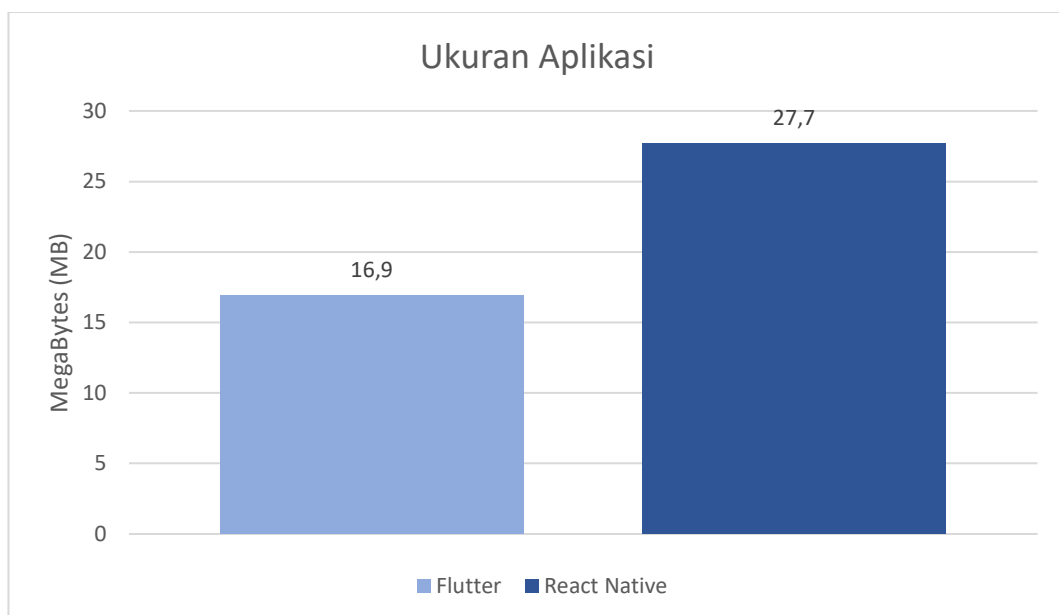
Directory: C:\Users\adamm\Documents\Project Skripsi\ReactNativeProject\android\app\build\outputs\apk\release

Mode                LastWriteTime         Length Name
----                -
-a-----          7/12/2022  11:28 PM      29082204 app-release.apk

```

Gambar 4. 24 Hasil *Build* Apk Aplikasi *React Native*

Seperti yang dapat dilihat pada gambar **Gambar 4. 23** dan **Gambar 4. 24** hasil *build* APK antara aplikasi yang dibangun dengan *framework Flutter* dan *React Native* menunjukkan perbedaan ukuran yang cukup jauh, dimana ukuran aplikasi *Flutter* sebesar 17733808 Bytes (B) atau sama dengan 16,9 Megabytes (MB) sedangkan ukuran aplikasi *React Native* sebesar 29082204 Bytes (B) atau sama dengan 27,7 Megabytes (MB).



Gambar 4. 25 Perbandingan Ukuran Aplikasi

Tabel 4. 7 *Package yang di Import*

<i>Flutter</i>	
<i>File</i>	<i>Code</i>
homepage.dart	<pre>import 'package:flutter/material.dart'; import 'package:flutter_project/src/utility/theme_color.dart'; import 'package:flutter_project/src/utility/theme_text.dart'; import 'package:flutter_project/src/views/listMahasiswa.dart'; import 'package:flutter_project/src/views/notFoundPage.dart';</pre>
listMahasiswa.dart	<pre>import 'dart:convert'; import 'package:flutter/material.dart'; import 'package:flutter/services.dart'; import 'package:flutter_project/src/utility/theme_color.dart'; import 'package:flutter_project/src/utility/theme_text.dart'; import 'package:flutter_project/src/views/detailMahasiswa.dart';</pre>
detailMahasiswa .dart	<pre>import 'dart:collection'; import 'package:flutter/material.dart'; import 'package:flutter_project/src/utility/theme_color.dart'; import 'package:flutter_project/src/utility/theme_text.dart';</pre>
notFoundPage .dart	<pre>import 'package:flutter/material.dart'; import 'package:flutter_project/src/utility/theme_text.dart';</pre>
<i>React Native</i>	
File	Code
Homepage.js	<pre>import {View, Text, StyleSheet, Image, TouchableOpacity} from 'react-native'; import React from 'react'; import ThemeText from '../assets/style/ThemeText'; import Styles from '../assets/style/Styles'; import ThemeColor from '../assets/style/ThemeColor';</pre>
Mahasiswa.js	<pre>import { View, Text, StyleSheet, TextInput, Image, FlatList, TouchableOpacity, } from 'react-native';</pre>

	<pre>import React, {useState, useEffect} from 'react'; import Style from '../assets/style/Styles'; import SearchIcon from '../assets/icon/search.png'; import ThemeText from '../assets/style/ThemeText'; import ThemeColor from '../assets/style/ThemeColor';</pre>
DetailMahasiswa.js	<pre>import {Text, View} from 'react-native'; import React from 'react'; import ThemeText from '../assets/style/ThemeText'; import Style from '../assets/style/Styles';</pre>
NotFoundPage.js	<pre>import {Text, View, Image} from 'react-native'; import React from 'react'; import maintenance from '../assets/image/maintenance.png'; import Styles from '../assets/style/Styles'; import ThemeText from '../assets/style/ThemeText';</pre>

b. Kompleksitas Kode

Pada tahap pengujian ini membandingkan kompleksitas kode *function* pada aplikasi *Flutter* dan *React Native*, *function* yang digunakan untuk pengujian memiliki tugas untuk melakukan pencarian mahasiswa berdasarkan nama mahasiswa. Metode yang digunakan untuk menguji ataupun mengukur kompleksitas kode menggunakan metode yang disebut *Cyclomatic Complexity*.

Cyclomatic Complexity menghitung tingkat linearitas jalur independen kode, *cyclomatic complexity* digunakan untuk mengukur tingkat kompleksitas kode. Semakin rendah nilai *cyclomatic complexity* maka semakin rendah juga tingkat kompleksitas kode nya.

Metode kalkulasi *Cyclomatic Complexity* kode dapat dihitung menggunakan rumus berikut:

$$CC = E - N + 2P$$

dimana,

E = jumlah edge

N = jumlah node

P = jumlah komponen terhubung = 1

Berikut ini adalah *function* untuk melakukan *filter* mahasiswa pada aplikasi *Flutter* dan *React Native*

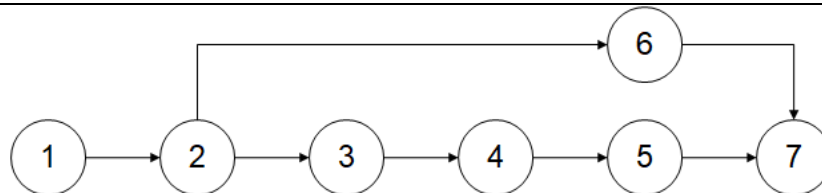
Tabel 4. 8 Kode *Function runfilter* Pada Aplikasi *Flutter*

Ln	Code	Node
1	<code>void _runFilter(String keyword) {</code>	1
2	<code>List results = [];</code>	
3	<code>if (keyword.isNotEmpty) {</code>	2
4	<code>String textData = keyword.toLowerCase();</code>	3
5	<code>List filteredData = _mahasiswa.where((user) {</code>	4
6	<code>return user["nama"].toLowerCase()</code> <code>.contains(textData);</code>	
7	<code>}).toList();</code>	
8	<code>results = filteredData;</code>	5
9	<code>} else {</code>	6
10	<code>results = _mahasiswa;</code>	
11	<code>}</code>	
12	<code>setState(() => _mahasiswalist = results);</code>	7
13	<code>};</code>	

Tabel 4. 9 Kode *Function runfilter* Pada Aplikasi *React Native*

Ln	Code	Node
1	<code>const runFilter = keyword => {</code>	1
2	<code>let result;</code>	
3	<code>if (keyword) {</code>	2
4	<code>const textData = keyword.toLowerCase();</code>	3
5	<code>const filteredData = mahasiswa.filter(function (item) {</code>	4
6	<code>return item.nama.toLowerCase().indexOf(textData) > -1;</code>	
7	<code>});</code>	
8	<code>result = filteredData;</code>	5
9	<code>} else {</code>	6

Ln	Code	Node
10	<code>result = mahasiswa;</code>	
11	<code>}</code>	
12	<code>setMahasiswaList(result);</code>	7
13	<code>};</code>	



Gambar 4. 26 Flowgraph Function runfilter Pada Aplikasi Flutter Dan React Native

Seperti yang dapat dilihat pada **Tabel 4. 8** dan

Tabel 4. 9 merupakan sebuah *function* yang memiliki tugas yang sama, yaitu melakukan pencarian data mahasiswa berdasarkan *keyword* yang dimasukkan. Namun menggunakan dua bahasa pemrograman yang berbeda yaitu *Dart* dan juga *Javascript*, berikut ini hasil analisis kompleksitas kode untuk *function* diatas.

Tabel 4. 10 Hasil Analisis Kompleksitas Kode Pada *Function runfilter*

	<i>Flutter</i>	<i>React Native</i>
Jumlah baris	13	13
Node	7	7
Edge	7	7
<i>Cyclomatic complexity</i>	$= E - N + 2P$ $= 7 - 7 + 2(1)$ $= 2$	$= E - N + 2P$ $= 7 - 7 + 2(1)$ $= 2$

Karena nilai *cyclomatic complexity* pada *function runFilter* bernilai 2 maka terdapat 2 kemungkinan eksekusi program, yaitu:

1. Path 1 = 1-2-3-4-5-7
2. Path 2 = 1-2-6-7

c. Performa

Tahap pengujian ini memberikan detail informasi mengenai performa pada aplikasi yang dibangun menggunakan *framework Flutter* dan *React Native*. Seperti yang sudah dijelaskan sebelumnya kedua aplikasi tersebut dibuat persis sama agar menghindari perbedaan hasil pengujian yang diakibatkan oleh perbedaan fungsional aplikasi. Namun begitu tetap ada sedikit perbedaan pada penggunaan elemen UI, dimensi teks, *styling* dan lainnya, yang seharusnya tidak memberikan dampak yang cukup besar pada performa aplikasi.

Tujuan yang ingin dicapai pada tahap ini adalah mengetahui apakah aplikasi yang dibangun menggunakan *framework Flutter* dan *React Native* memiliki perbedaan performa yang signifikan, dengan membandingkan penggunaan CPU, konsumsi memori dan *frame rate* aplikasi. Pengujian ini dibatasi hanya pada *platform* Android dan untuk mensimulasikan penggunaan nyata dari aplikasi, kedua aplikasi tersebut dijalankan pada *real device*, adapun perangkat yang digunakan adalah Xiaomi Redmi Note 10 Pro.

Pengujian dilakukan dengan mengambil nilai puncak penggunaan CPU, memori, dan *frame rate* ketika menggunakan aplikasi, adapun skenario penggunaan aplikasi dijelaskan pada **Tabel 4. 6**.

1) Penggunaan CPU

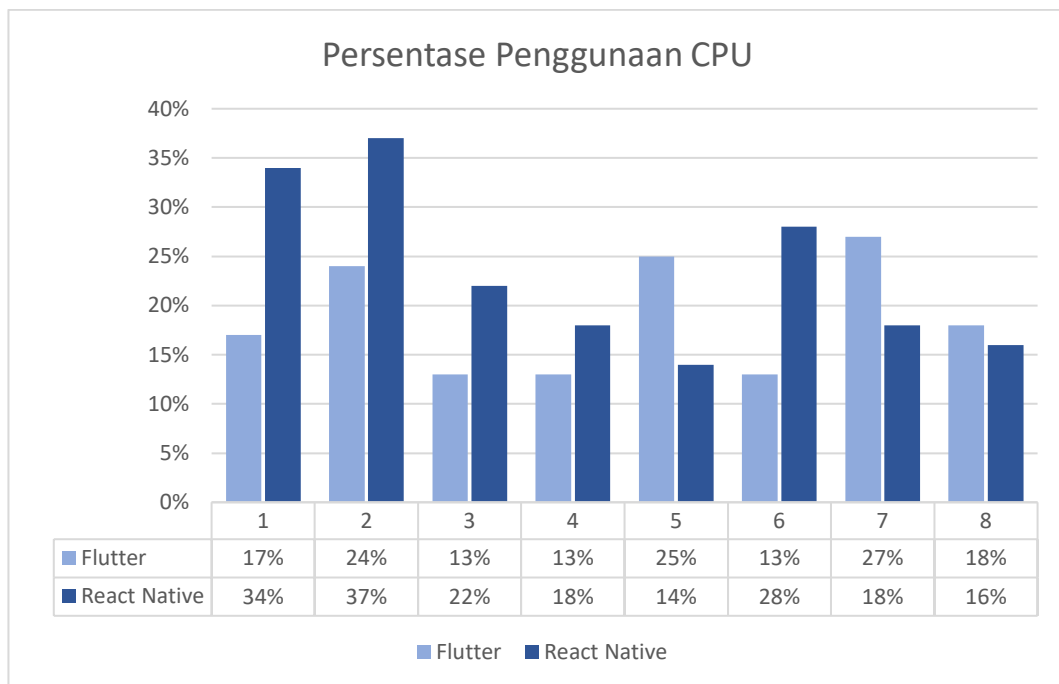
Berdasarkan hasil pengujian penggunaan CPU pada aplikasi *Flutter* dan *React Native* dengan menggunakan *tools* Android Profiler pada Android Studio menunjukkan bahwa penggunaan CPU pada aplikasi *Flutter* lebih rendah dibandingkan pada aplikasi *React Native*.

Tabel 4. 11 Hasil Pengujian CPU

Nilai	<i>Flutter</i>	<i>React Native</i>
AVG	19%	23%
MAX	27%	37%
MIN	13%	14%

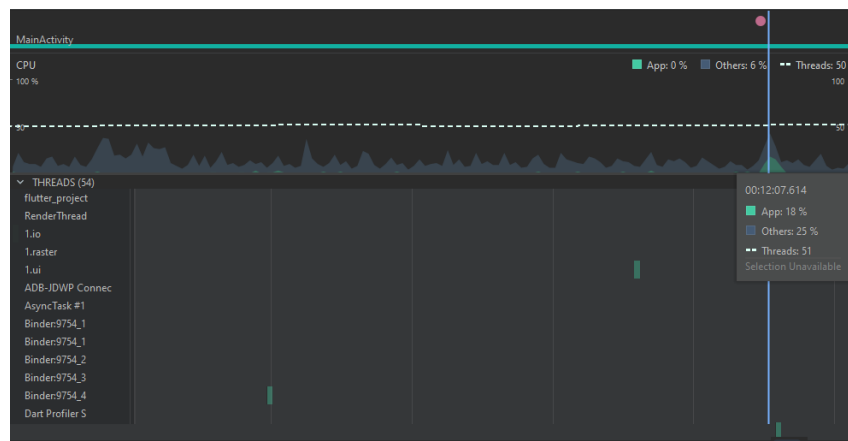
Seperti yang dapat dilihat pada **Tabel 4. 11** pengujian dilakukan dengan menjalankan skenario secara berurutan dimana hasilnya adalah penggunaan CPU rata-rata pada aplikasi *Flutter* lebih rendah 4% jika dibandingkan aplikasi

React Native, selain itu penggunaan CPU tertinggi selama pengujian menunjukkan aplikasi *Flutter* menggunakan 27% sedangkan *React Native* mencapai 37% penggunaan CPU dan penggunaan CPU minimal aplikasi *Flutter* menggunakan sebanyak 13% dibandingkan aplikasi *React Native* menggunakan 14% penggunaan CPU. Hal ini dapat disimpulkan bahwa secara keseluruhan aplikasi *Flutter* menggunakan CPU lebih sedikit dibandingkan aplikasi *React Native*.

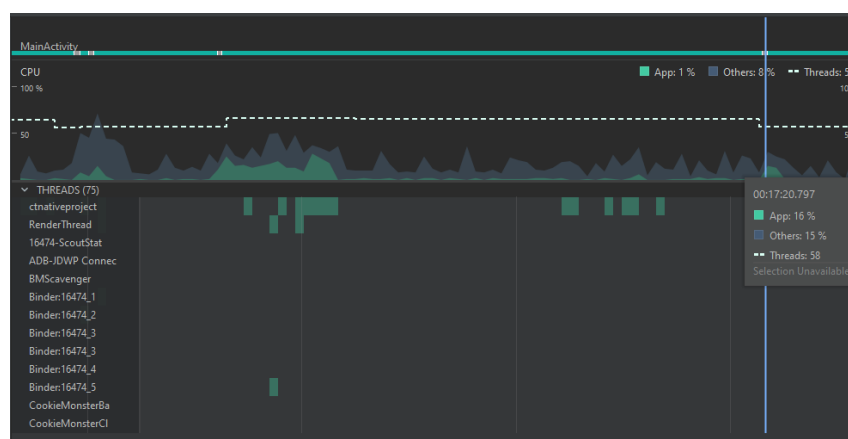


Gambar 4. 27 Persentase Penggunaan CPU

Pada **Gambar 4. 27** dapat dilihat penggunaan CPU untuk masing-masing skenario dimana aplikasi *React Native* menunjukkan penggunaan CPU yang lebih rendah untuk beberapa skenario



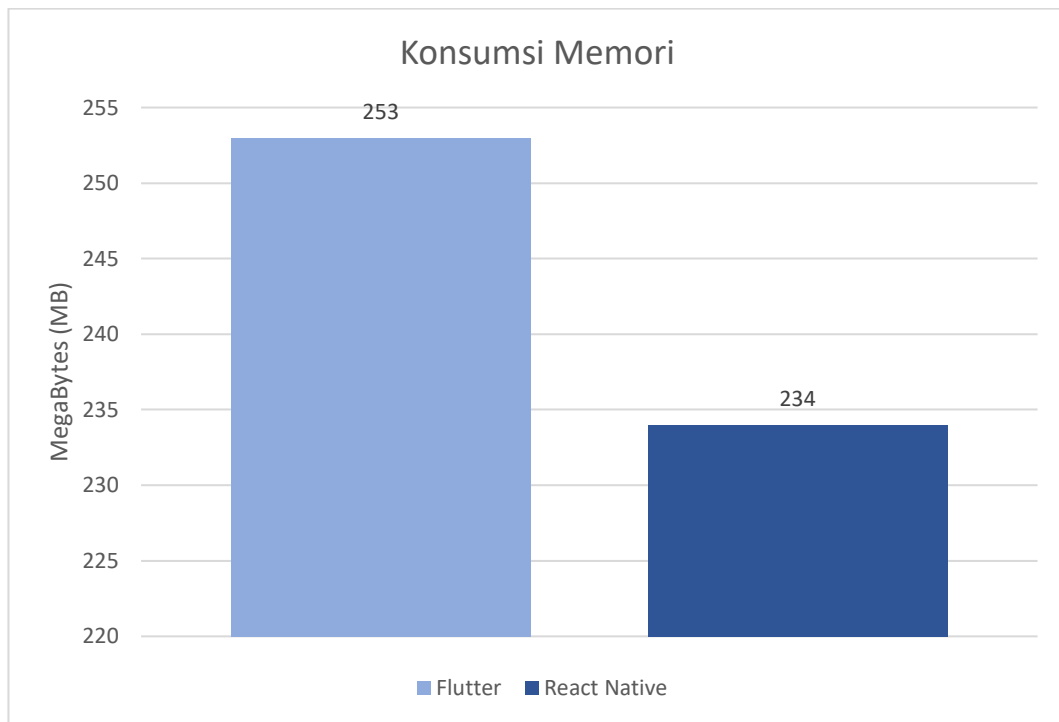
Gambar 4. 28 Penggunaan CPU Aplikasi *Flutter*



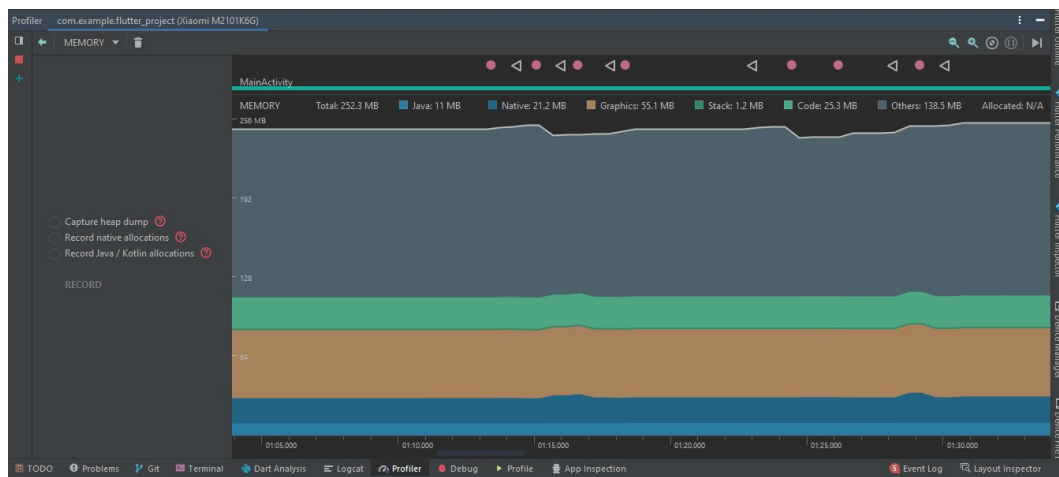
Gambar 4. 29 Penggunaan CPU Aplikasi *React Native*

2) Konsumsi Memori

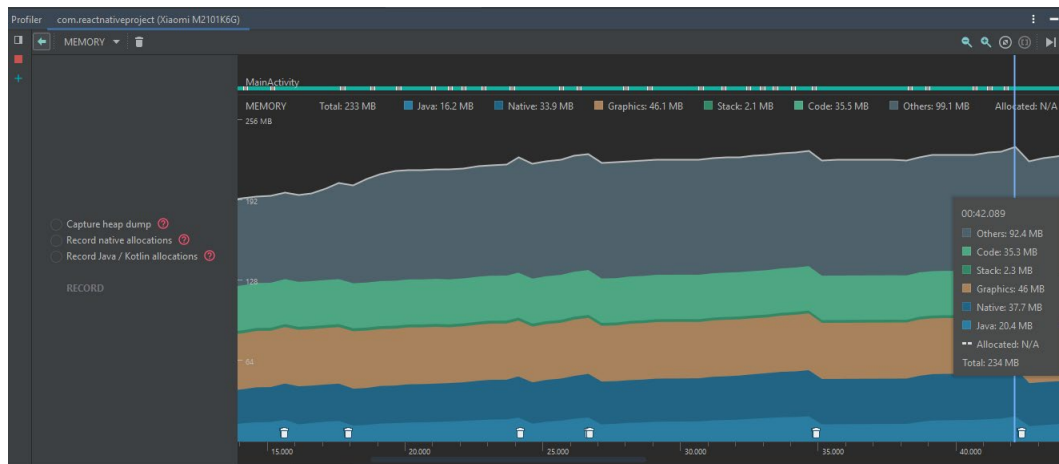
Pengujian konsumsi memori pada aplikasi *Flutter* dan *React Native* menggunakan *tools* *Android Profiler* dari *Android Studio*, pengujian dilakukan dengan menjalankan skenario 1 – 8 selama kurang lebih 1 menit. Hasilnya adalah aplikasi *React Native* mengkonsumsi memori lebih rendah 7,5% atau 19 MB dibandingkan aplikasi *Flutter*.



Gambar 4. 30 Konsumsi Memori



Gambar 4. 31 Penggunaan Memori Aplikasi *Flutter*

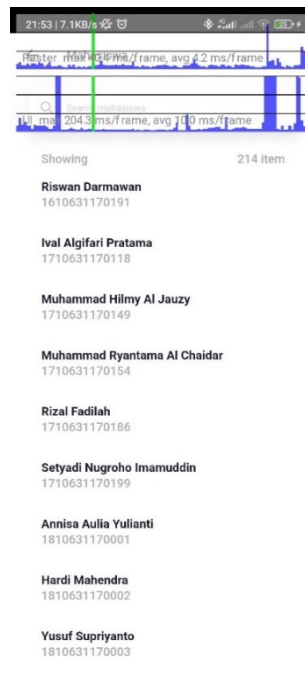


Gambar 4. 32 Penggunaan Memori Aplikasi *React Native*

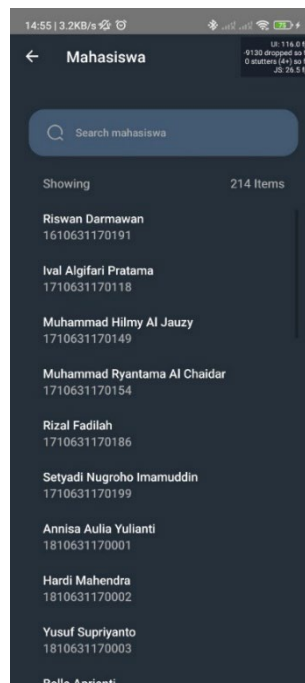
3) *Frame Rate*

Pengujian untuk mengukur *frame rate* pada aplikasi menggunakan *tools performance overlay* yang telah disediakan oleh masing-masing *framework*, sama seperti tahap sebelumnya pengujian dilakukan dengan mengikuti skenario secara berurutan.

Seperti yang dapat dilihat pada Gambar *Flutter* tidak menggunakan satuan *fps* melainkan *ms/frame* untuk menampilkan nilai *frame rate* nya. Maka dari itu untuk menyamakan satuan yang digunakan (*fps*) nilai *frame rate* pada aplikasi *Flutter* dikonversi ke dalam *fps* dengan membagi nilai *ms/frame* dengan 1000



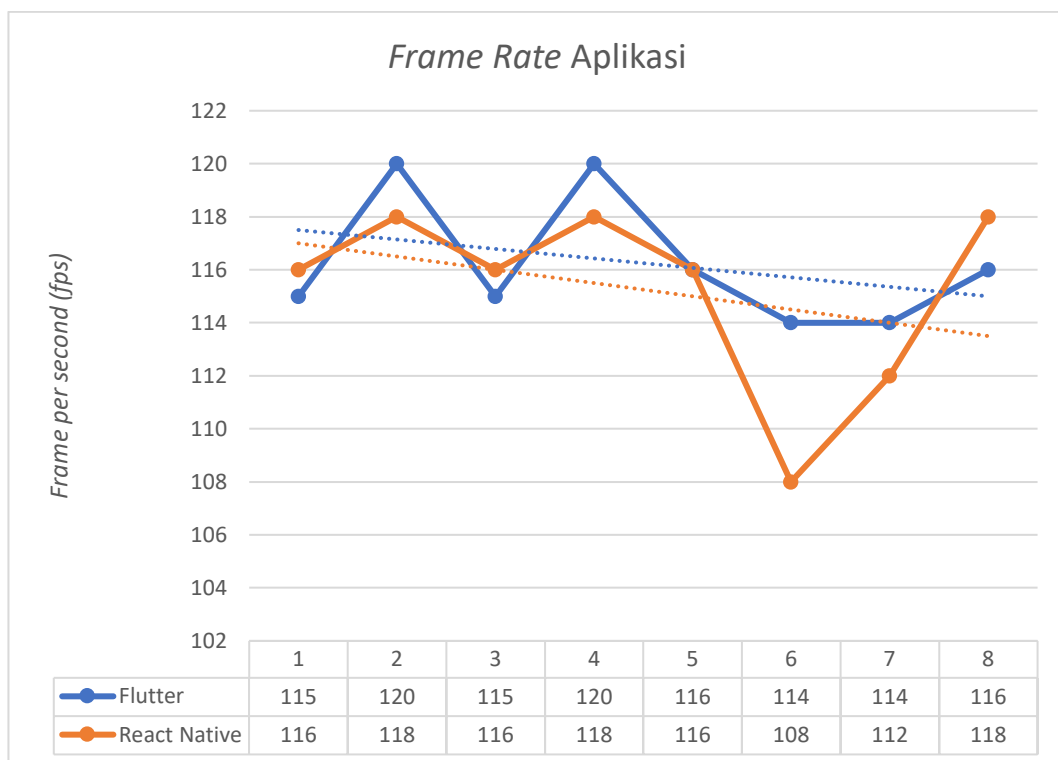
Gambar 4. 33 *Performance Overlay* Aplikasi *Flutter*



Gambar 4. 34 *Performance Overlay* Aplikasi *React Native*

Berbeda dengan pengujian penggunaan CPU dan konsumsi memori dimana nilai yang lebih rendah adalah yang lebih baik, pengujian pada *frame rate* dinilai lebih baik apabila mendekati atau sama dengan nilai *refresh rate* yang didukung oleh perangkat tersebut, dalam studi kasus ini perangkat yang digunakan adalah

ponsel Xiaomi Redmi Note 10 Pro yang sudah mendukung *refresh rate* tinggi sampai dengan 120 Hz. Semakin tinggi nilai *frame rate* aplikasi atau mendekati *refresh rate* perangkat maka tampilan yang dihasilkan lebih lancar (*smooth*). Rata-rata *frame rate* yang diperoleh pada aplikasi *Flutter* adalah 116,5 *fps* sedangkan pada aplikasi *React Native* 115,25, artinya *framerate* aplikasi *Flutter* 0,86% lebih tinggi dari aplikasi *React Native*.



Gambar 4. 35 Grafik *Framerate* Aplikasi

4.1.2.4 Kesimpulan

Pengujian perbandingan ukuran aplikasi menunjukkan bahwa aplikasi yang dibangun menggunakan Flutter 48% lebih kecil dibandingkan aplikasi React Native, untuk pengujian kompleksitas kode keduanya memberikan hasil yang cukup baik dengan nilai cyclomatic complexity sama-sama bernilai 2, hal ini dikarenakan dukungan function built-in yang ada pada bahasa Dart dan Javascript cukup membantu dalam penulisan kode agar menjadi lebih sederhana. Pengujian performa membandingkan penggunaan CPU, konsumsi memori dan framerate, dimana pada penggunaan CPU aplikasi Flutter lebih rendah 4% dari aplikasi React

Native, kemudian konsumsi memori aplikasi React Native lebih rendah 7,5% dari aplikasi Flutter dan untuk pengujian framerate diantara kedua aplikasi nilai rata-rata nya hanya terpaut 0,86% yang mana aplikasi Flutter sedikit lebih unggul dari aplikasi React Native.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan dari penelitian yang telah dilakukan maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Penelitian ini berhasil melakukan perancangan dan pengembangan aplikasi *mobile* berbasis android dengan menggunakan *cross-platform application framework* yaitu *Flutter* dan *React Native* dengan menggunakan metode pengembangan *Rapid Application Development* (RAD). Metode ini memiliki empat tahapan yaitu perencanaan kebutuhan, desain sistem, proses pengembangan, dan implementasi.
2. Baik *Flutter* dan *React Native* memiliki kelebihan dan kekurangan masing-masing dan pilihan terbaik kembali lagi bergantung pada kemampuan dan kebutuhan dari proyek. *Flutter* akan cocok untuk aplikasi yang mengedepankan performa dan stabilitas, *user interface* yang lebih kompleks dan sudah terbiasa menggunakan *Java* (*Dart* akan mudah dipahami apabila sudah memahami *Java/Kotlin*). *React Native* cocok digunakan apabila memiliki website dan komponennya ingin digunakan kembali untuk aplikasi *mobile* (*single tech stack*), menginginkan *user interface* layaknya *native* dan sudah terbiasa menggunakan bahasa *Javascript*.

5.2 Saran

Dari hasil penelitian yang telah dilakukan ada beberapa saran sebagai bahan referensi untuk pengembangan penelitian yang akan dilakukan selanjutnya:

1. Disarankan tidak hanya membandingkan antar *cross platform application framework* seperti *Flutter* dan *React Native* tetapi dibandingkan juga dengan aplikasi *native* yang dibuat menggunakan *Java/Kotlin*.
2. Dilakukan juga pengujian pada *platform* *iOS* untuk mengetahui bagaimana *framework* *Flutter* dan *React Native* berjalan pada *platform* tersebut.

3. Menggunakan metode pengujian yang berbeda untuk membandingkan kedua *framework* tersebut, sebagai contoh pengujian untuk menampilkan visual yang lebih dinamis atau melakukan komputasi yang berat.

DAFTAR PUSTAKA

- Anggit, L., Pamungkas, B., & Imrona, D. M. (2020). *Analisa Perbandingan Kinerja Cross Platform Mobile Framework React Native dan Flutter*. 7(1), 2195.
- Beautiful native apps in record time | Flutter*. (n.d.). Retrieved November 14, 2021, from <https://flutter.dev/>
- ben Stegner. (2022). *What Is an APK File and What Does It Do? Explained*. <https://www.makeuseof.com/tag/what-is-apk-file/>
- Biørn-Hansen, A., Rieger, C., Grønli, T. M., & ... (2020). An empirical investigation of performance overhead in cross-platform mobile development frameworks. *Empirical Software* <https://doi.org/10.1007/s10664-020-09827-6>
- Cross-platform mobile frameworks used by global developers 2021 | Statista*. (n.d.). Retrieved August 19, 2022, from <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>
- Dart overview | Dart*. (n.d.). Retrieved November 12, 2021, from <https://dart.dev/overview>
- Ebone, A., Tan, Y., & Jia, X. (2018). A performance evaluation of cross-platform mobile application development approaches. *2018 IEEE/ACM 5th International* <https://ieeexplore.ieee.org/abstract/document/8543442/>
- Fentaw, A. E. (2020). *Cross platform mobile application development: a comparison study of react native vs flutter*. [jyx.jyu.fi. https://jyx.jyu.fi/handle/123456789/70969](https://jyx.jyu.fi/handle/123456789/70969)
- Frame Rate - Everything You Need to Know - NFI*. (n.d.). Retrieved August 21, 2022, from <https://www.nfi.edu/frame-rate/>
- Guadaña, R. R., Perez, M. R., & Rutaquio Jr, L. (2013). A comprehensive review for central processing unit scheduling algorithm. *International Journal of Computer Science Issues (IJCSI)*, 10(1), 353.
- Hjort, E. (2020). *Evaluation of React Native and Flutter for cross-platform mobile application development*.
- Hourani, H., Wasmi, H., & Alrawashdeh, T. (2019). A code complexity model of object oriented programming (OOP). *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 560–564.
- Indonesia: smartphone users 2026 | Statista*. (n.d.). Retrieved November 3, 2021, from <https://www.statista.com/statistics/266729/smartphone-users-in-indonesia/#statisticContainer>

- Introducing JSX – React.* (n.d.). Retrieved November 13, 2021, from <https://reactjs.org/docs/introducing-jsx.html>
- JSX - What Is a JSX? & Introduction to Advanced.* (n.d.). Retrieved November 13, 2021, from <https://www.reactenlightenment.com/react-jsx/5.1.html>
- Karim, D., & Santoso, H. B. (2019). Perancangan Dan Usability Evaluation Prototipe Informasi Akademik Menggunakan Metode Rapid Application Development. *Jurnal Ilmiah ILKOMINFO-Ilmu Komputer & Informatika*, 2(2).
- Kuitunen, M. (2019). *Cross-Platform Mobile Application Development with React Native*. trepo.tuni.fi. <https://trepo.tuni.fi/handle/123456789/27139>
- Martin, J., & Tanaamah, A. R. (2018). Perancangan Dan Implementasi Sistem Informasi Penjualan Berbasis Desktop Website Menggunakan Framework Bootstrap Dengan Metode Rapid Application Development, Studi Kasus Toko Peralatan Bayi ‘Eeng Baby Shop.’ *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 5(1), 57–68.
- Mobile Operating System Market Share Worldwide | Statcounter Global Stats.* (n.d.). Retrieved November 3, 2021, from <https://gs.statcounter.com/os-market-share/mobile/>
- Murdiani, D., & Hermawan, H. (2022). Perbandingan Metode Waterfall dan RAD (Rapid Application Development) Pada Pengembangan Sistem Informasi. (*JurTI*) *Jurnal Teknologi Informasi*, 6(1), 14–23.
- Smartphone users 2026 | Statista.* (n.d.). Retrieved November 3, 2021, from <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
- Tysk, A. C., & Kling, E. (2021). *An evaluation of MobileCross-platform Developmentusing React Native and NativeAndroid*. diva-portal.org. <https://www.diva-portal.org/smash/record.jsf?pid=diva2:1585650>
- What is RAM | Random Access Memory - javatpoint.* (n.d.). Retrieved August 21, 2022, from <https://www.javatpoint.com/ram>
- Wu, W. (2018). *React Native vs Flutter, Cross-platforms mobile application frameworks*. theseus.fi. <https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf?sequence=1>

LAMPIRAN

Lampiran 1. Biodata Penulis**BIODATA PENULIS**

Nama : Adam Mukti Wibisono
Tempat Tanggal Lahir : Cirebon, 15 Agustus 2000
Jenis Kelamin : Laki-laki
Alamat : Blok Kr. Anyar RT09/RW03 Desa Jangga, Kec. Losarang,
Kab. Indramayu, 45253
Agama : Islam
Email : adam.mukti18103@student.unsika.ac.id
No. Telepon : 08997584110

Pendidikan Formal :

1. 2006 – 2012 : SD Negeri Losarang
2. 2012 – 2015 : SMP Negeri Unggulan Sindang
3. 2015 – 2018 : SMA Negeri 1 Sindang
4. 2018 – 2022 : Universitas Singaperbangsa Karawang

Lampiran 2. *Source code main.dart (Flutter)*

```
import 'package:flutter/material.dart';
import 'package:flutter_project/src/views/homepage.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        scaffoldBackgroundColor: const Color(0xFFFFFFFF),
        home: HomePage(),
      );
    }
  }
```

Lampiran 3. Source code homepage.dart (Flutter)

```
import 'package:flutter/material.dart';
import 'package:flutter_project/src/utility/theme_color.dart';
import 'package:flutter_project/src/utility/theme_text.dart';
import 'package:flutter_project/src/views/listMahasiswa.dart';
import 'package:flutter_project/src/views/notFoundPage.dart';

class HomePage extends StatelessWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            SizedBox(
              height: 40,
            ),
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 16),
              child: Text(
                'Halo, Selamat datang di',
                style: textMuted,
              ),
            ),
            SizedBox(
              height: 12,
            ),
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 16),
              child: Text(
                'SISTEM\nINFORMASI\nTUGAS AKHIR',
                style: textHeading,
              ),
            ),
            SizedBox(
              height: 24,
            ),
            GestureDetector(
              behavior: HitTestBehavior.translucent,
              onTap: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => ListMahasiswa(),
                  ),
                );
              },
            ),
            child: Container(
              margin: EdgeInsets.symmetric(horizontal: 16),
              height: 150,
              width: MediaQuery.of(context).size.width,
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(20),
                color: primary,
                boxShadow: [
                  BoxShadow(
```



```

child: Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    GestureDetector(
      behavior: HitTestBehavior.translucent,
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => NotFoundPage(),
          ),
        );
      },
    ),
    child: Container(
      height: 160,
      width: MediaQuery.of(context).size.width * 0.42,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20),
        color: Color(0xfffff5e9),
      ),
    ),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            SizedBox(
              height: 24,
            ),
            Padding(
              padding: const EdgeInsets.only(left: 24),
              child: Text(
                'Dosen',
                style: textMuted.copyWith(
                  color: Color(0xffd9c0ab),
                ),
              ),
            ),
            SizedBox(
              height: 12,
            ),
            Padding(
              padding: const EdgeInsets.only(left: 24),
              child: Text(
                '1234',
                style: textHeading.copyWith(
                  color: Color(0xffd29b74),
                  fontSize: 24,
                ),
              ),
            ),
          ],
        ),
        Column(
          mainAxisAlignment: MainAxisAlignment.end,
          children: [
            Padding(
              padding: const EdgeInsets.only(
                right: 24, bottom: 24),
            ),
          ],
        ),
      ],
    ),
  ],
)

```



```

        color: Color(0xffBEB3CE),
      ),
    ),
  ),
  SizedBox(
    height: 12,
  ),
  Padding(
    padding: const EdgeInsets.only(left: 24),
    child: Text(
      '1234',
      style: textHeading.copyWith(
        color: Color(0xff9073B4),
        fontSize: 24,
      ),
    ),
  ),
],
),
Column(
  mainAxisAlignment: MainAxisAlignment.end,
  children: [
    Padding(
      padding: const EdgeInsets.only(
        right: 24, bottom: 24),
      child: Image.asset(
        'assets/icon/arrow_right_dark.png',
      ),
    ),
  ],
),
],
),
),
),
GestureDetector(
  behavior: HitTestBehavior.translucent,
  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => NotFoundPage(),
      ),
    );
  },
  child: Container(
    height: 160,
    width: MediaQuery.of(context).size.width * 0.42,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(20),
      color: Color(0xffF4FFFB),
    ),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            SizedBox(

```


Lampiran 4. Source code listMahasiswa.dart (Flutter)

```
import 'dart:convert';

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:flutter_project/src/utility/theme_color.dart';
import 'package:flutter_project/src/utility/theme_text.dart';
import 'package:flutter_project/src/views/detailMahasiswa.dart';

class ListMahasiswa extends StatefulWidget {
  const ListMahasiswa({Key? key}) : super(key: key);

  static TextEditingController searchBar = TextEditingController();

  @override
  State<ListMahasiswa> createState() => _ListMahasiswaState();
}

class _ListMahasiswaState extends State<ListMahasiswa> {
  List _mahasiswa = [];
  List _mahasiswaList = [];

  Future<void> readJson() async {
    final String response = await rootBundle.loadString('api/mahasiswa.json');
    final data = await json.decode(response);
    setState(() {
      _mahasiswa = data;
      _mahasiswaList = _mahasiswa;
    });
  }

  @override
  void initState() {
    readJson();
    super.initState();
  }

  void _runFilter(String keyword) {
    List results = [];
    if (keyword.isNotEmpty) {
      String textData = keyword.toLowerCase();
      List filteredData = _mahasiswa.where((user) {
        return user["nama"].toLowerCase().contains(textData);
      }).toList();
      results = filteredData;
    } else {
      results = _mahasiswa;
    }
    setState(() => _mahasiswaList = results);
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        appBar: AppBar(
          title: Text(
            'Mahasiswa',
            style: textTitle,

```

```

    ),
    elevation: 0,
    backgroundColor: white,
    foregroundColor: dark,
  ),
  body: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 24),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        SizedBox(
          height: 12,
        ),
        Container(
          // margin: EdgeInsets.symmetric(horizontal: 16),
          height: 60,
          width: MediaQuery.of(context).size.width,
          decoration: BoxDecoration(
            boxShadow: [
              BoxShadow(
                color: Color(0xFF000000).withOpacity(0.1),
                blurRadius: 15,
                offset: Offset(0, 4), // changes position of shadow
              ),
            ],
          ),
          child: TextField(
            onChanged: (value) => _runFilter(value),
            controller: ListMahasiswa.searchBar,
            decoration: InputDecoration(
              hintText: 'Search mahasiswa',
              hintStyle: textMuted.copyWith(fontSize: 14),
              filled: true,
              fillColor: Colors.white,
              prefixIconColor: Colors.red,
              prefixIcon: Icon(Icons.search),
              border: OutlineInputBorder(
                borderSide: BorderSide.none,
                borderRadius: BorderRadius.circular(20),
              ),
            ),
          ),
        ),
        SizedBox(
          height: 24,
        ),
        Padding(
          padding: EdgeInsets.symmetric(horizontal: 16),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
              Text(
                'Showing',
                style: textMuted,
              ),
              Text('${_mahasiswaList.length} item', style: textMuted),
            ],
          ),
        ),
      ],
    ),
  ),

```

```

Expanded(
  child: ListView.builder(
    itemCount: _mahasiswaList.length,
    itemBuilder: (context, index) {
      return ListTile(
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) =>
                DetailMahasiswa(data: _mahasiswaList[index]),
            ),
          );
        },
        title: Text(
          _mahasiswaList[index]["nama"],
          style: textHeading.copyWith(fontSize: 18),
        ),
        subtitle: Text(
          _mahasiswaList[index]["npm"].toString(),
          style: textMuted,
        ),
      );
    }
  ),
),
),
),
),
);
}
}

```


Lampiran 5. Source code detailMahasiswa.dart (Flutter)

```
import 'dart:collection';

import 'package:flutter/material.dart';
import 'package:flutter_project/src/utility/theme_color.dart';
import 'package:flutter_project/src/utility/theme_text.dart';

class DetailMahasiswa extends StatelessWidget {
  final LinkedHashMap<String, dynamic> data;
  const DetailMahasiswa({Key? key, required this.data}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        appBar: AppBar(
          title: Text(
            'Detail Mahasiswa',
            style: textTitle,
          ),
          elevation: 0,
          backgroundColor: white,
          foregroundColor: dark,
        ),
        body: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 24),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Padding(
                padding: const EdgeInsets.only(top: 36, bottom: 8),
                child: Text(
                  'Proposal Skripsi',
                  style: textHeading,
                ),
              ),
              DetailInfoWidget(
                title: 'Nama',
                value: data['nama'],
              ),
              DetailInfoWidget(
                title: 'NPM',
                value: data['npm'].toString(),
              ),
              DetailInfoWidget(
                title: 'Dosen Pembimbing',
                value: data['pembimbing_proposal'],
              ),
              DetailInfoWidget(
                title: 'Judul Proposal',
                value: data['judul_proposal'],
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```

        ],
      ),
    ),
  );
}
}

class DetailInfoWidget extends StatelessWidget {
  final String title;
  final String value;
  const DetailInfoWidget({Key? key, required this.title, required
this.value})
    : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: EdgeInsets.only(top: 12),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            title,
            style: textMuted.copyWith(fontWeight: FontWeight.w500),
          ),
          Text(
            value == '' ? '-' : value,
            style: textTitle.copyWith(height: 1.5),
          ),
        ],
      ),
    );
  }
}

```

Lampiran 6. *Source code notFoundPage.dart (Flutter)*

```

import 'package:flutter/material.dart';
import 'package:flutter_project/src/utility/theme_text.dart';

class NotFoundPage extends StatelessWidget {
  const NotFoundPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Padding(
        padding: const EdgeInsets.all(24.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            Image.asset('assets/image/maintenance.png'),
            Text('Halaman ini masih dalam pengembangan',
              style: textTitle.copyWith(
                fontSize: 24,
              ),
              textAlign: TextAlign.center)
          ],
        ),
      ),
    );
  }
}

```

Lampiran 7. *Source code App.js (React Native)*

```
import * as React from 'react';
import {NavigationContainer} from '@react-navigation/native';
import {createNativeStackNavigator} from '@react-navigation/native-stack';

import Homepage from './src/pages/Homepage';
import Mahasiswa from './src/pages/Mahasiswa';
import DetailMahasiswa from './src/pages/DetailMahasiswa';
import NotFoundPage from './src/pages/NotFoundPage';

const Stack = createNativeStackNavigator();

const App = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen
          name="Homepage"
          component={Homepage}
          options={{headerShown: false}}
        />
        <Stack.Screen name="Mahasiswa" component={Mahasiswa} />
        <Stack.Screen name="DetailMahasiswa" component={DetailMahasiswa} />
        <Stack.Screen name="NotFoundPage" component={NotFoundPage} />
      </Stack.Navigator>
    </NavigationContainer>
  );
};

export default App;
```

Lampiran 8. Source code Homepage.js (React Native)

```
import {
  View,
  Text,
  StyleSheet,
  Image,
  TouchableOpacity,
  Pressable,
} from 'react-native';
import React from 'react';
import ThemeText from '../assets/style/ThemeText';
import Styles from '../assets/style/Styles';
import ThemeColor from '../assets/style/ThemeColor';
// import arrowRightWhite from '../assets/icon/arrow_right_dark.png';

const Homepage = ({navigation}) => {
  return (
    <View style={Styles.bg_dark}>
      <View style={Styles.container}>
        <Text style={[ThemeText.textMuted, style.welcomeText]}>
          Halo, Selamat datang di
        </Text>
        <Text style={[ThemeText.textHeading, style.textHeading]}>
          {`SISTEM\nINFORMASI\nTUGAS AKHIR`}
        </Text>
        <TouchableOpacity onPress={() => navigation.navigate('Mahasiswa')}>
          <View style={[style.hero, {flexDirection: 'row'}]>
            <View style={{padding: 24}}>
              <Text style={ThemeText.textTitle}>Mahasiswa</Text>
              <Text style={[ThemeText.textHeading, {fontSize: 48}]}>215</Text>
            </View>
            <Image
              source={require('../assets/icon/arrowRightLight.png')}
              style={{position: 'absolute', bottom: 24, right: 24}}
            />
          </View>
        </TouchableOpacity>
        <View style={style.menuWrapper}>
          <TouchableOpacity
            onPress={() => navigation.navigate('NotFoundPage')}
            style={[{backgroundColor: '#FFF5E9'}, style.menu]>
            <Text style={[ThemeText.textTitle, {color: '#D9C0AB'}]}>Dosen</Text>
            <Text
              style={[ThemeText.textHeading, {color: '#D29B74', fontSize: 36}]}>
              123
            </Text>
            <Image
              source={require('../assets/icon/arrowRightDark.png')}
              style={{position: 'absolute', bottom: 24, right: 24}}
            />
          </TouchableOpacity>
          <TouchableOpacity
            onPress={() => navigation.navigate('NotFoundPage')}
            style={[{backgroundColor: '#EBFAFF'}, style.menu]>
            <Text style={[ThemeText.textTitle, {color: '#B6D2D8'}]}>
              Proposal
            </Text>
            <Text
              style={[ThemeText.textHeading, {color: '#5DABC1', fontSize: 36}]}>
```

```

        123
      </Text>
      <Image
        source={require('../assets/icon/arrowRightDark.png')}
        style={{position: 'absolute', bottom: 24, right: 24}}
      />
    </TouchableOpacity>
    <TouchableOpacity
      onPress={() => navigation.navigate('NotFoundPage')}
      style={[{backgroundColor: '#F7F4FF'}, style.menu]}>
      <Text style={[ThemeText.textTitle, {color: '#BEB3CE'}]}>
        Kolokium
      </Text>
      <Text
        style={[ThemeText.textHeading, {color: '#9073B4', fontSize: 36}]}>
        123
      </Text>
      <Image
        source={require('../assets/icon/arrowRightDark.png')}
        style={{position: 'absolute', bottom: 24, right: 24}}
      />
    </TouchableOpacity>
    <TouchableOpacity
      onPress={() => navigation.navigate('NotFoundPage')}
      style={[{backgroundColor: '#F4FFFB'}, style.menu]}>
      <Text style={[ThemeText.textTitle, {color: '#B6D0C5'}]}>
        Yudisium
      </Text>
      <Text
        style={[ThemeText.textHeading, {color: '#6FABA1', fontSize: 36}]}>
        123
      </Text>
      <Image
        source={require('../assets/icon/arrowRightDark.png')}
        style={{position: 'absolute', bottom: 24, right: 24}}
      />
    </TouchableOpacity>
  </View>
</View>
</View>
);
};

export default Homepage;

const style = StyleSheet.create({
  welcomeText: {
    marginBottom: 12,
  },
  textHeading: {
    marginBottom: 24,
  },
  hero: {
    backgroundColor: ThemeColor.primary,
    height: 150,
    borderRadius: 20,
    marginBottom: 24,
    // shadowColor: '#51C3FF',
    // elevation: 5,
  },
});

```

```
    },
    menuWrapper: {
      flexDirection: 'row',
      flex: 2,
      justifyContent: 'space-between',
      flexWrap: 'wrap',
    },
    menu: {
      height: 170,
      width: '47%',
      borderRadius: 20,
      marginBottom: 24,
      padding: 24,
      position: 'relative',
    },
  },
});
```

Lampiran 9. *Source code Mahasiswa.js (React Native)*

```

import {
  View,
  Text,
  StyleSheet,
  TextInput,
  Image,
  FlatList,
  TouchableOpacity,
} from 'react-native';
import React, {useState, useEffect} from 'react';
import Style from '../assets/style/Styles';
import SearchIcon from '../assets/icon/search.png';
import ThemeText from '../assets/style/ThemeText';
import ThemeColor from '../assets/style/ThemeColor';

const Mahasiswa = ({navigation}) => {
  let [mahasiswaList, setMahasiswaList] = useState([]);
  let mahasiswa = require('../api/mahasiswa.json');
  useEffect(() => {
    setMahasiswaList(mahasiswa);
  }, []);

  React.useLayoutEffect(() => {
    navigation.setOptions({
      headerTintColor: '#ffffff',
      headerShadowVisible: false,
      headerStyle: {
        backgroundColor: '#252F3A',
      },
    });
  }, [navigation]);

  const runFilter = keyword => {
    let result;
    if (keyword) {
      const textData = keyword.toLowerCase();
      const filteredData = mahasiswa.filter(function (item) {
        return item.nama.toLowerCase().indexOf(textData) > -1;
      });
      result = filteredData;
    } else {
      result = mahasiswa;
    }
    setMahasiswaList(result);
  };

```



```

const renderListMahasiswa = ({item}) => {
  return (
    <TouchableOpacity
      onPress={() => {
        /* 1. Navigate to the Details route with params */
        navigation.navigate('DetailMahasiswa', {
          mhs: item,
        });
      }}>
      <View
        style={{
          flexDirection: 'row',
          marginBottom: 24,
          marginHorizontal: 18,
        }}>
        <View style={{flex: 1}}>
          <Text style={ThemeText.textTitle}>{item.nama}</Text>
          <Text style={ThemeText.textMuted}>{item.npm}</Text>
        </View>
      </View>
    </TouchableOpacity>
  );
};

return (
  <View style={Style.bg_dark}>
    <View style={Style.container}>
      <View style={style.searchWrapper}>
        <Image source={SearchIcon} style={style.iconStyle} />
        <TextInput
          placeholder="Search mahasiswa"
          placeholderTextColor={ThemeColor.secondary}
          style={style.textSearchStyle}
          onChangeText={text => runFilter(text)}
        />
      </View>
      <View
        style={{
          flexDirection: 'row',
          marginTop: 24,
          marginHorizontal: 18,
          marginBottom: 24,
        }}>
        <View style={{flex: 1}}>
          <Text style={[[ThemeText.textMuted]]}>Showing</Text>
        </View>
        <View style={{flex: 1, alignItems: 'flex-end'}}>

```

```

        <Text style={[ThemeText.textMuted]}>
          {mahasiswaList.length} Items
        </Text>
      </View>
    </View>
  </View>
  <FlatList
    data={mahasiswaList}
    renderItem={renderListMahasiswa}
    keyExtractor={item => item.npm.toString()}
  />
</View>
</View>
);
};

const style = StyleSheet.create({
  searchWrapper: {
    flexDirection: 'row',
    alignItems: 'center',
    height: 60,
    width: '100%',
    backgroundColor: '#445A72',
    borderRadius: 20,
    paddingLeft: 24,
  },
  iconStyle: {width: 20, height: 20, marginRight: 12},
  textSearchStyle: {
    fontSize: 16,
    fontWeight: '500',
    color: ThemeColor.secondary,
  },
});

export default Mahasiswa;

```

Lampiran 10. *Source code DetailMahasiswa.js (React Native)*

```

import {StyleSheet, Text, View} from 'react-native';
import React from 'react';
import ThemeText from '../assets/style/ThemeText';
import ThemeColor from '../assets/style/ThemeColor';
import Style from '../assets/style/Styles';

const DetailMahasiswa = ({route, navigation}) => {
  const {mhs} = route.params;
  React.useLayoutEffect(() => {
    navigation.setOptions({
      headerTintColor: '#ffffff',
      headerShadowVisible: false,
      headerStyle: {
        backgroundColor: '#252F3A',
      },
    });
  }, [navigation]);
  return (
    <View style={Style.bg_dark}>
      <View style={Style.container}>
        <Text style={ThemeText.textHeading}>Proposal Skripsi</Text>
        <DetailInfoComponent title={'Nama'} value={mhs.nama} />
        <DetailInfoComponent title={'NPM'} value={mhs.npm} />
        <DetailInfoComponent
          title={'Dosen Pembimbing '}
          value={mhs.pembimbing_proposal}
        />
        <DetailInfoComponent
          title={'Judul Proposal'}
          value={mhs.judul_proposal}
        />
      </View>
    </View>
  );
};

const DetailInfoComponent = ({title, value}) => {
  return (
    <View style={{marginTop: 12}}>
      <Text style={ThemeText.textMuted}>{title}</Text>
      <Text style={[ThemeText.textTitle, {lineHeight: 26}]}>
        {value !== '' ? value : '-'}
      </Text>
    </View>
  );
};

```

```
export default DetailMahasiswa;
```

Lampiran 11. *Source code* NotFoundPage.js (React Native)

```

import {StyleSheet, Text, View, Image} from 'react-native';
import React from 'react';
import maintenance from '../assets/image/maintenance.png';
import Styles from '../assets/style/Styles';
import ThemeText from '../assets/style/ThemeText';

const NotFoundPage = ({navigation}) => {
  React.useLayoutEffect(() => {
    navigation.setOptions({headerShown: false});
  }, [navigation]);
  return (
    <View style={Styles.bg_dark}>
      <View style={[Styles.container, {justifyContent: 'center', flex: 1}]}>
        <Image source={maintenance} style={{width: '100%'}} />
        <Text
          style={[ThemeText.textTitle, {fontSize: 24, textAlign: 'center'}]}>
          Halaman ini masih dalam pengembangan
        </Text>
      </View>
    </View>
  );
};

export default NotFoundPage;

const styles = StyleSheet.create({});

```