

Analisa Perbandingan Kinerja Cross Platform Mobile Framework React Native dan Flutter

Ludfi Anggit Brilliant Pamungkas¹
Drs. Mahmud Imrona, MT²

^{1,2}Fakultas Informatika, Universitas Telkom, Bandung

¹anggitludfi@students.telkomuniversity.ac.id, mahmudimrona²@telkomuniversity.ac.id

Abstrak

Dalam membuat aplikasi *mobile*, *developer* aplikasi harus membuat aplikasi untuk android dan ios, akan tetapi aplikasi yang dibuat untuk kedua sistem operasi tersebut memiliki bahasa dan logika pemrograman yang berbeda, sehingga *developer* memerlukan waktu, usaha, dan biaya dua kali untuk mengembangkan aplikasi yang sama. Kemudian, pada tahun 2015, facebook menyelesaikan permasalahan ini dengan mengembangkan suatu framework bernama react native, framework react native ini sehingga memungkinkan *developer* untuk mengembangkan aplikasi dalam *single code base* yang kemudian dapat berjalan di ios dan android, hingga kini *framework* react native merupakan *mobile framework* yang paling sering digunakan berdasarkan data survei dari website stackoverflow. Kemudian pada tahun 2018, google mengembangkan framework bernama flutter, sama halnya dengan react native, aplikasi yang dibuat menggunakan framework ini dapat berjalan pada ios dan android, perbedaannya adalah flutter memiliki engine buatan mereka sendiri bernama skia sementara react native menggunakan bridge untuk mengcompile code yang telah dibuat menggunakan bahasa pemrograman javascript menjadi Bahasa pemrograman swift dan c# untuk ios dan java/ kotlin untuk android. Penelitian ini menguji kinerja kedua framework pada 3 aplikasi yang telah dibuat dengan tampilan, dan fungsi yang sama lalu melihat penggunaan cpu, dan memori. Hasil akhir yang diperoleh react native unggul dalam penggunaan memori yang lebih rendah, sementara flutter unggul dalam penggunaan cpu yang lebih rendah.

Kata kunci : *cross platform mobile framework*, react native, flutter, android, *mobile programming*.

Abstract

In making mobile application, developers often have to make applications for Android and iOS, even though the applications made for the two operating systems have different programming languages and logic, so developers need time, effort and cost twice to develop the application. Then in 2015, Facebook developed a framework called react native to overcome this problem, react native enabled developers to develop applications in single code base which can run on iOS and Android, until now framework react native is the most frequent mobile framework used based on survey data from stackoverflow. Furthermore, in 2018, Google developed a framework called flutter, same as react native, applications made using this framework can run on iOS and Android, the difference is that Flutter has their own engine called Skia while react native uses bridges to compile code that has been made using javascript programming language into swift / c # language for ios and java / kotlin for android. This study examines the performance of the two frameworks in 3 applications that have been made with the same appearance, and functions then monitors CPU usage, and memory usage. The final results obtained by react native has lower memory usage than flutter, while flutter has lower cpu usage than react native.

Keywords: *cross platform mobile framework*, react native, flutter, android, *mobile programming*.

1. Pendahuluan

Latar Belakang

Smartphone kini memiliki peran penting dalam kehidupan, menurut data dari statista total pengguna *smartphone* pada tahun 2014 sebanyak 1.57 miliar dan hingga 2018 terdapat 2.53 miliar pengguna *smartphone* [1]. Banyaknya pengguna *smartphone* ini juga diikuti bertambahnya jumlah aplikasi yang diunduh dengan total sebanyak 205.4 miliar aplikasi [2]. Hal ini menunjukkan adanya kebutuhan akan penggunaan aplikasi *smartphone*. Selanjutnya, berdasarkan data dari statcounter, *mobile operating system* yang paling sering digunakan terbagi menjadi dua yaitu android sebesar 74.17% dan ios 24.75% [3], sehingga agar suatu aplikasi dapat digunakan oleh semua pengguna pada kedua platform, aplikasi perlu dibuat pada ios dan android. Sementara, untuk membuat aplikasi pada kedua platform tersebut, diperlukan keahlian, dan bahasa pemrograman yang berbeda, seperti java/kotlin untuk android dan object-c/swift untuk ios. Dengan demikian *developer* memerlukan waktu, biaya dan usaha dua kali untuk mengembangkan aplikasi.



Gambar 1 Data statistik mobile operating system market share worldwide

Untuk mengatasi permasalahan tersebut, para *developer* menggunakan *cross platform mobile framework*, teknologi *framework* ini memungkinkan *developer* aplikasi untuk membuat satu *code base* yang dapat diimplementasikan pada *platform* ios dan android, hingga saat ini terdapat banyak *framework* yang dapat digunakan, seperti ionic dan cordova dengan pendekatan *webview*, xamarin, flutter, react native, dan nativescript dengan pendekatan *native*. Akan tetapi, penelitian kali ini hanya akan membahas mengenai flutter dan react native. Flutter dipilih karena merupakan *framework* terbaru yang dikembangkan oleh google dengan *engine* buatan mereka bernama skia sementara react native adalah *framework* yang telah dikembangkan oleh facebook terlebih dahulu yang kemudian menjadi *framework* paling populer digunakan di kalangan *developer* aplikasi berdasarkan data survey dari stack overflow [4]. React native telah digunakan oleh beberapa perusahaan teknologi seperti ,facebook, instagram, bloomberg, tesla, uber, walmart, skype, pinterest [5], sementara flutter digunakan oleh google ads, Alibaba group, jd finance, grab, tencent, dan stadia.



Gambar 2 Beberapa perusahaan teknologi yang menggunakan react native

React Native dikembangkan pertama kali oleh facebook pada tahun 2015, pada awal kemunculannya, framework ini menjadi revolusioner karena pada waktu itu framework yang ada masih berbasis webview dan react native memiliki kemampuan untuk dapat membuat aplikasi dengan tampilan dan kinerja yang mirip layaknya aplikasi yang dibuat menggunakan pengembangan nativenya. React native menggunakan bahasa pemrograman javascript dan merupakan pengembangan dari framework react yang sebelumnya dikembangkan untuk web [6]. Untuk dapat menjalankan aplikasi pada ios dan android, react native memiliki bridge untuk memanggil *native rendering* APIs menggunakan objective-c untuk ios dan java untuk android, sehingga aplikasi akan di render menjadi komponen *native UI* yang memiliki tampilan dan kinerja layaknya aplikasi *native*. Sama halnya dengan flutter, aplikasi yang dibuat menggunakan flutter memiliki kinerja dan tampilan layaknya aplikasi yang dibuat menggunakan pengembangan nativenya. Flutter dikembangkan oleh google pada tahun 2017 dan dirilis pada desember 2018 lalu, flutter menggunakan bahasa pemrograman bernama dart yang juga dikembangkan oleh google. Flutter menggunakan engine buatan mereka sendiri bernama skia, dan tidak menggunakan *bridge* seperti pada react native, sehingga diklaim dapat memiliki kinerja yang lebih baik dan efisien[6].



Gambar 3 Beberapa perusahaan teknologi yang menggunakan react native

Gamebench adalah *mobile application profiling tool* yang dapat memonitor dan merekam kinerja cpu, memori, penggunaan daya, *frame per second* dan network usage pada aplikasi. Gamebench juga dapat melakukan screenshot tiap detik pada saat aplikasi berjalan, sehingga memudahkan *developer* untuk menganalisa kinerja aplikasi pada saat dilakukan skenario tertentu, Gamebench hingga saat ini digunakan oleh beberapa *developer game* dan beberapa perusahaan teknologi, seperti Samsung, Huawei, OnePlus, Rovio dan Netmarble [7].



Gambar 4 Beberapa perusahaan yang menggunakan gamebench

Penelitian kali ini akan membandingkan kinerja cpu, dan memori menggunakan gamebench *tool* pada 3 aplikasi yang dibuat menggunakan framework flutter dan react native dengan tampilan, dan fungsi yang sama. Data penggunaan cpu dan memori menunjukkan bagaimana kinerja aplikasi di suatu *device* terhadap interaksi dengan pengguna, semakin rendah penggunaan cpu dan memorinya semakin baik kinerja aplikasi, hal ini menjadi dasar untuk menentukan framework mana yang memiliki kinerja lebih baik. Aplikasi diuji dengan melakukan beberapa skenario yang telah ditentukan dan gamebench akan memonitor seluruh kinerja cpu dan memori. Hasil ini kemudian ditampilkan dalam bentuk grafik untuk dianalisa dan diambil kesimpulan.

Topik dan Batasannya

Berdasarkan latar belakang yang telah dijelaskan, penelitian ini menganalisa dan membandingkan kinerja aplikasi yang dibuat menggunakan *framework* react native dan flutter dengan metrik *cpu usage*, dan *memory usage* menggunakan *tool* bernama gamebench.

Rumusan masalah pada penelitian ini adalah untuk membandingkan kinerja pada aplikasi yang dibuat menggunakan react native dibandingkan dengan flutter, dan metrik yang akan diuji adalah *cpu usage*, dan *memory usage*.

Pada penelitian ini terdapat beberapa Batasan masalah, yaitu platform yang digunakan adalah android dan terdapat 3 aplikasi yang diuji, aplikasi pertama merupakan aplikasi kuis sederhana, aplikasi kedua perhitungan transaksi mingguan, dan ketiga aplikasi yang menampilkan resep makanan, semua aplikasi tersebut dibuat menggunakan framework react native dan flutter dan memiliki tampilan dan fungsi yang sama, sehingga nantinya dapat diperoleh hasil yang relevan.

Tujuan

Penelitian ini memiliki tujuan untuk membandingkan kinerja framework react native dan framework flutter dengan mengamati penggunaan cpu dan penggunaan memori.

2. Studi Terkait

2.1 Studi Komparatif

Berdasarkan penelitian yang dilakukan oleh wu w tahun 2019, pada studi kasus dengan membuat aplikasi *list* yang identik kemudian menghitung *dropped frames* ketika melakukan *scrolling* pada *list*, menunjukkan hasil bahwa flutter memiliki *dropped frames* yang lebih sedikit dibandingkan react native [8]. Penelitian lainnya yang dilakukan Tomas Vidhall dan Nicolas Hansson dalam penelitiannya berjudul “Effects on performance and usability for cross-platform application development using React Native” dengan studi kasus *homescreen application*, menunjukkan hasil bahwa tidak ada permasalahan kinerja yang signifikan pada aplikasi android yang dibuat menggunakan react native maupun aplikasi versi *native* nya, akan tetapi pada pengujian dengan intensitas rendah, react native menunjukkan kinerja *cpu load* yang tinggi dibandingkan dengan *native*, hal ini menjadi permasalahan ketika aplikasi menjadi lebih kompleks sehingga membutuhkan kinerja cpu lebih tinggi, pada tes penggunaan memori juga menunjukkan kemiripan hasil, kemudian dari segi ukuran besar file aplikasi, react native memiliki ukuran 24,35 MB sedangkan native android 13,39 MB [9]. Pada *tes frame per second* (FPS) react native menunjukkan kinerja yang lebih baik dengan rata-rata *dropped frame* sebesar 0.61% sementara native android memiliki rata-rata frame drop sebesar 0.89% dan pada pengujian response time react native menghasilkan kinerja yang lebih baik dibanding dengan native android [9]. Penelitian serupa yang dilakukan oleh A.Lelli dan V. Bostrand, pada aplikasi penyimpanan kontak, dan *photogrid* yang menampilkan 3 foto tiap baris, menunjukkan hasil bahwa pada aplikasi pertama react native memiliki kinerja yang kurang efisien dibandingkan native dengan perbedaan penggunaan memori 90% lebih besar, penggunaan cpu 100% lebih besar, dan gpu 20% lebih besar, sama halnya pada aplikasi kedua react native menggunakan memori 35% lebih besar, cpu 300% lebih besar dan penggunaan gpu 100% lebih besar akan tetapi fps yang dihasilkan pada react native lebih baik dengan 60fps dibandingkan dengan native android 45 fps [10]. Penelitian serupa yang dilakukan oleh Jarko Saarinen yang berjudul “Evaluating Cross Platform Mobile App Performance with Video-Based Measurements” dengan menguji kinerja pada framework react native, flutter, cordova, titanium, xamarin dan native menunjukkan hasil bahwa flutter memiliki kinerja paling baik dibandingkan dengan *cross platform mobile framework* lainnya, tetapi tidak lebih baik dibandingkan native [10].

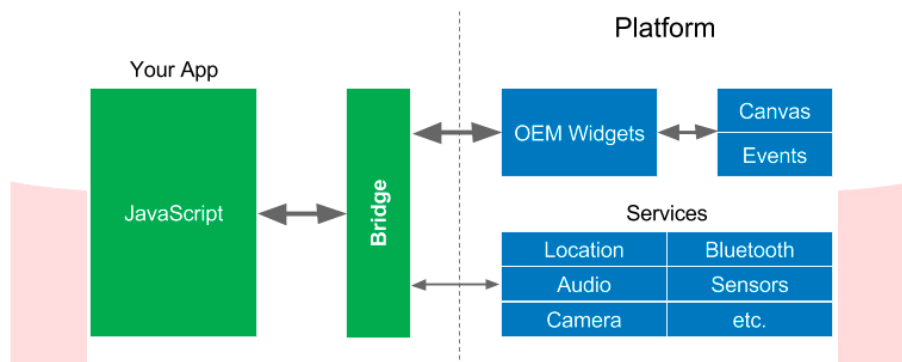
2.2 Cross Platform Mobile Framework

Cross platform mobile framework adalah framework yang memungkinkan *developer* untuk membuat aplikasi yang dapat digunakan pada platform ios dan android, sampai saat ini terdapat banyak framework seperti ionic, xamarin, adobe phone gap, cordova akan tetapi berdasarkan survey dari stackoverflow pada 2019, flutter adalah framework yang paling disukai disusul dengan react native [4].

2.3 React Native

React Native dirilis oleh facebook pada tahun 2015, pada awal kemunculannya, react native menjadi framework yang paling revolusioner karena memiliki tampilan dan kinerja layaknya aplikasi yang dikembangkan secara nativenya, hingga saat ini react native merupakan framework yang paling sering digunakan, react native menggunakan bahasa pemrograman javascript dan framework react js yang dikembangkan oleh facebook.

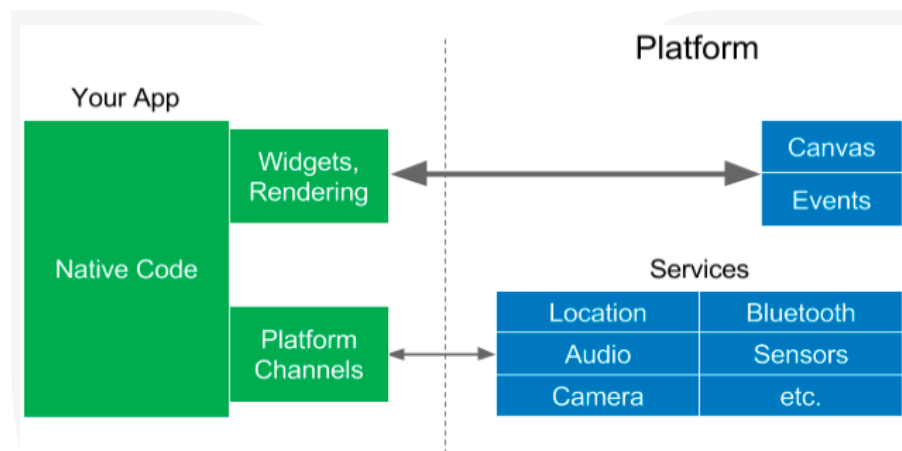
Pada React native terdapat empat bagian inti, react adalah kode yang ditulis, javascript mentafsirkan apa yang ditulis, lalu ada bridge untuk berkomunikasi dengan native code, dan bagian native. Aspek kunci pada arsitektur ini terdapat pada javascript dan native, untuk berkomunikasi mereka memerlukan JSON messages yang ditransmisikan melalui bridge. Bridge inilah yang mengkonversikan javascript code menjadi native code yang dapat dikenali oleh smartphone [6].



Gambar 5 Arsitektur React Native

2.4 Flutter

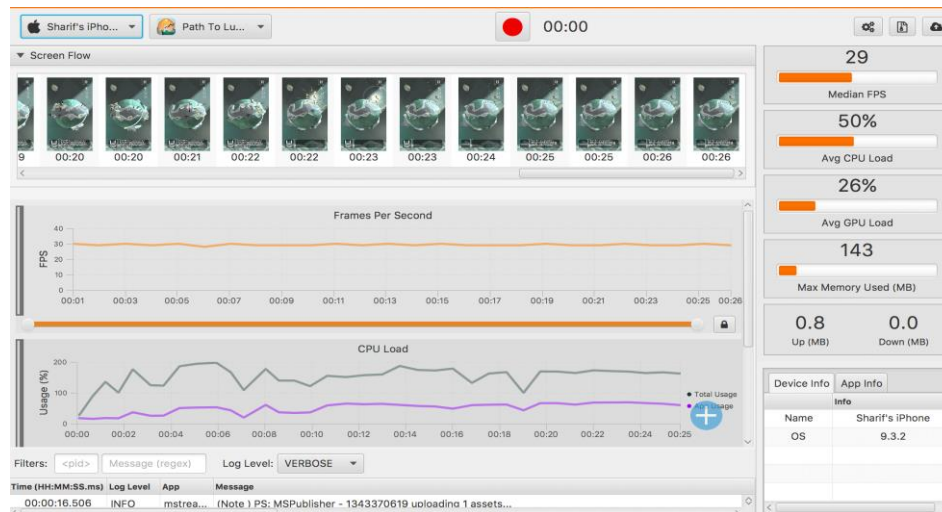
Flutter dirilis oleh google pada desember 2018, sama halnya dengan react native, aplikasi yang dibuat menggunakan framework ini dapat dijalankan pada ios dan android, perbedaannya flutter tidak menggunakan javascript bridge untuk berkomunikasi dengan native android, tetapi flutter menggunakan bahasa pemrograman dart yang dikembangkan oleh google dan di optimasi untuk dapat dijalankan langsung pada native platform, sehingga secara teori memungkinkan flutter memiliki kinerja yang lebih baik dibanding react native [6]. Hingga saat ini flutter merupakan cross platform mobile framework paling disukai developer berdasarkan data survei stackoverflow 2019 [4].



Gambar 6 Arsitektur Flutter

2.5 Gamebench

Pada penelitian kali ini *tool* yang akan digunakan bernama Gamebench (lihat Gambar 7). Gamebench adalah *mobile application profiling tool* yang dapat memonitor kinerja cpu, memori, penggunaan daya, *frame per second*, dan *network usage* smartphone pada aplikasi yang sedang dijalankan. Tool ini juga dapat melakukan screenshot tiap detik pada saat aplikasi digunakan, sehingga memudahkan developer untuk menganalisa kinerja aplikasi. Gamebench digunakan oleh beberapa developer game dan perusahaan teknologi, seperti Samsung, Huawei, OnePlus, Rovio dan Netmarble [7]. Tool ini digunakan karena dapat menampilkan penggunaan cpu, memori dan terdapat fitur screenshot tiap detik yang dapat memudahkan pada pengujian.

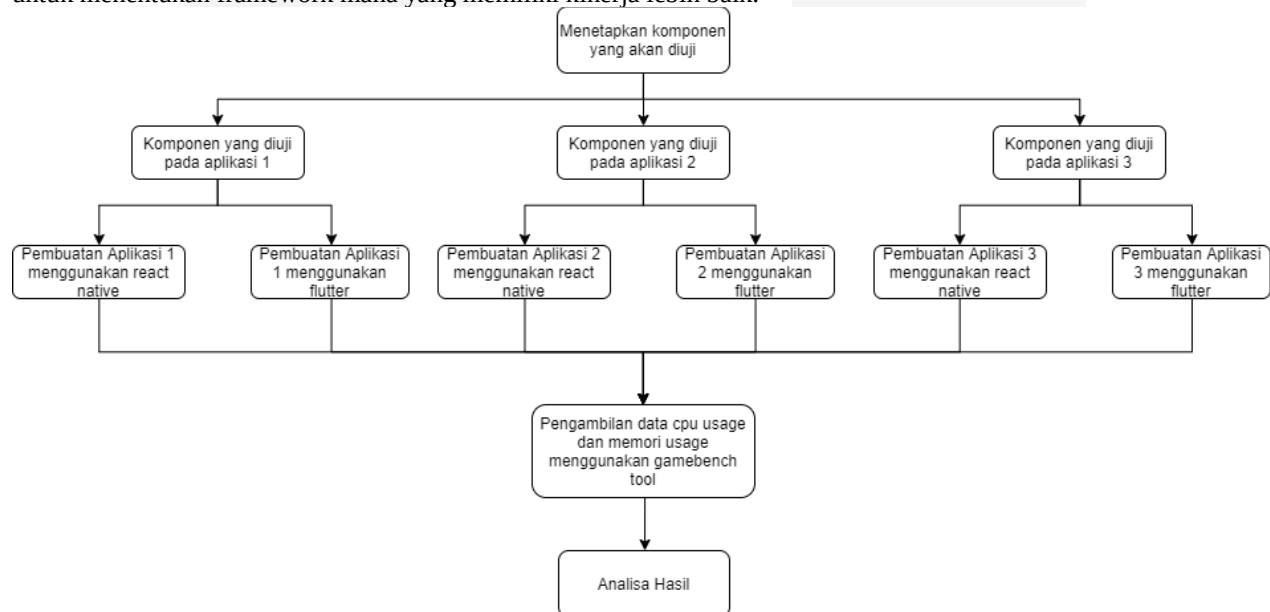


Gambar 7 Gamebench

3. Sistem yang Dibangun

3.1 Gambaran Umum Sistem

Pertama, Sistem dibangun dengan menetapkan komponen yang akan diuji untuk ketiga aplikasi terlebih dahulu, berdasarkan komponen pengujian yang telah ditetapkan, aplikasi mulai dibuat menggunakan framework react native dan framework flutter, setelah aplikasi selesai dibuat, selanjutnya dilakukan pengambilan data *cpu usage* dan *memory usage* menggunakan gamebench tool, *cpu usage* dan *memory usage* yang rendah digunakan sebagai kriteria untuk menentukan framework mana yang memiliki kinerja lebih baik.

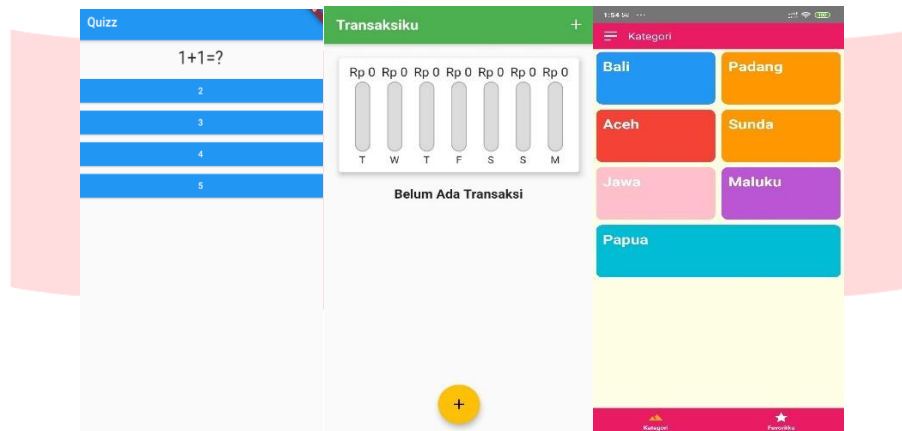


Gambar 8 Flowchart gambaran umum sistem penelitian

3.2 Aplikasi Yang Diuji

Pada Penelitian kali ini, terdapat 3 aplikasi yang akan diuji pertama aplikasi sederhana kuis pilihan ganda, terdapat 5 soal dengan 4 pilihan jawaban, tujuan menguji aplikasi ini adalah untuk mengetahui bagaimana kinerja framework pada implementasi aplikasi yang sederhana. Kedua, aplikasi transaksi, aplikasi ini berfungsi untuk menyimpan transaksi pengeluaran per hari, berdasarkan inputan dari pengguna kemudian menampilkannya dalam bentuk grafik bar per minggunya dengan skala 100. Tujuan menguji aplikasi ini adalah untuk mengetahui bagaimana

kinerja aplikasi terhadap input, edit, dan menampilkan output ke pengguna. Ketiga, aplikasi Resep Nusantara, aplikasi ini berfungsi untuk melihat gambar dan menampilkan resep makanan, gambar didownload dari website yang sudah di berikan, dan teks disimpan pada source code Tujuan menguji aplikasi ini adalah untuk mengetahui bagaimana kinerja aplikasi dalam menangani aktivitas pengunduhan gambar, menampilkan teks ke pengguna, dan melakukan scrolling (lihat gambar 8). Tampilan lengkap aplikasi dapat dilihat pada halaman lampiran.



Gambar 9 Aplikasi yang diuji

4. Evaluasi

4.1 Hasil Pengujian

Tabel 1 Hasil pengujian aplikasi 1

	Skenario Pengujian	Flutter Cpu Usage	React Native Cpu Usage	Flutter Memory Usage (MB)	React Native Memory Usage(MB)
1	Membuka Aplikasi	25%	37%	170	110
2	Memilih jawaban pada soal 1	11%	16%	175	115
3	Memilih jawaban pada soal 2	9%	12%	174	116
4	Memilih jawaban pada soal 3	8%	14%	176	120
5	Memilih jawaban pada soal 4	9%	13%	174	119
6	Restart Quiz	6%	11%	172	118
7	Idle	0%	0%	160	106
		9.71%	14.71%	171.57	114.86

Tabel 2 Hasil pengujian aplikasi 2

	Skenario Pengujian	Flutter Cpu Usage	React Native Cpu Usage	Flutter Memory Usage	React Native Memory Usage
1	Meuka Aplikasi	20%	33%	152	105
2	Menekan tool add	10%	15%	161	110
3	Menekan form nama transaksi	10.3%	17%	173	112
4	Mengisi nama transaksi	5.3%	10%	170	113
5	Menekan form jumlah	10%	16%	174	116
6	Mengisi angka jumlah	2%	5%	176	111
7	Menekan button tanggal	13%	16%	179	125

8	Memilih tanggal	15%	20%	208	134
9	Menyimpan Transaksi	11%	15%	211	138
10	Menghapus Transaksi	5%	7%	171	114
11	Idle	0%	0%	197	120
		9.24%	14.00%	179.27	118

Tabel 3 Hasil pengujian aplikasi 3

	Skenario Pengujian	Flutter Cpu Usage	React Native Cpu Usage	Flutter Memory Usage	React Native Memory Usage
1	Meuka Aplikasi	22%	36%	158	121
2	Menekan tool resep	15%	25%	214	156
3	Melakukan download gaar	10%	18%	173	124
4	Memilih resep menu	10%	14%	221	161
5	Melakukan scroll pada resep	15%	27%	194	136
6	Menekan tool back	12%	20%	194	132
7	Menampilkan gaar yang di download sebelumnya	9%	12%	220	154
8	Melakukan scroll pada menu	10%	20%	204	143
9	Menyimpan Menu Favorit	17%	25%	233	158
10	Menghapus Menu Favorit	7%	11%	211	148
11	Berpindah ke tab favorit	11%	16%	210	151
12	Idle	0%	0%	197	142
	Rata-rata	11.50%	18.67%	202.42	143.83

Hasil pengujian pada aplikasi ketiga menunjukkan *network usage* pada react native dan flutter memiliki kinerja yang mirip, dengan flutter 0.852 MB/S dan react native 0.823 MB/S

4.2 Analisis Hasil Pengujian

Berdasarkan hasil pengujian pada Tabel 1 React native menggunakan rata-rata cpu 14.71% dan menggunakan memori 114.86 MB sementara Flutter menggunakan rata-rata cpu 9.71% dan menggunakan memori 171.57 MB lalu berdasarkan hasil pengujian pada Tabel 2 React native menggunakan rata-rata cpu 14% dan menggunakan memori 118 MB sementara Flutter menggunakan rata-rata cpu 9.24% dan menggunakan memori 179.27 MB, kemudian berdasarkan hasil pengujian pada Tabel 3 React native menggunakan rata-rata cpu 18.67% dan menggunakan memori 143.83 MB sementara flutter menggunakan rata-rata cpu 11.50% dan menggunakan rata-rata memori 202.42 MB.

5. Kesimpulan

Berdasarkan analisis hasil pengujian, dapat disimpulkan baik react native maupun flutter memiliki keunggulan masing-masing dimana react native lebih unggul pada penggunaan memori yang lebih rendah sementara flutter lebih unggul pada penggunaan cpu yang lebih rendah. Sehingga dapat disimpulkan, Flutter sebagai suatu framework terbaru memiliki kinerja yang cukup baik dibandingkan dengan react native, sehingga memberi opsi lain bagi para developer untuk mengembangkan aplikasi *cross platform*.

Masih terdapat banyak penelitian yang dapat dilakukan, mengingat terdapat banyak skenario pengujian yang dapat dilakukan pada suatu aplikasi, saran terkait penelitian ini dan penelitian selanjutnya adalah membandingkan kinerja aplikasi yang dibuat dengan *framework* react native ataupun flutter dengan pengembangan nativenya, sehingga menarik nantinya dapat terlihat seberapa besar *trade off* perbedaan kinerja yang terjadi.

Daftar Pustaka

- [1] "Statista," 2019. [Online]. Available: www.statista.com/statistics/336095/number-of-smartphone-users-worldwide. [Accessed 22 December 2019].
- [2] "Statista," 2019. [Online]. Available: <https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>. [Accessed 22 December 22].
- [3] "Statcounter," 2019. [Online]. Available: <http://gs.statcounter.com/os-market-share/mobile/worldwide>. [Accessed 22 December 2019].
- [4] "Stackoverflow Developer Survey Results," 2019. [Online]. Available: <https://insights.stackoverflow.com/survey/2019>. [Accessed 22 December 2019].
- [5] "React Native," 2019. [Online]. Available: <https://facebook.github.io/react-native/> [Accessed 22 December 2019]. [Accessed 22 December 2019].
- [6] "what's revolutionry about flutter," 2018. [Online]. Available: <https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514> [Accessed 22 December 2019]. [Accessed 22 December 2019].
- [7] "Gamebench," 2019. [Online]. Available: <https://www.gamebench.net/>. [Accessed 22 December 2019].
- [8] W. Wu, "React Native vs Flutter cross-platform mobile application frameworks," 2018.
- [9] N. Hansson, "Effects on Performance and Usability for Cross Platform Application Development using React native," 2016.
- [10] L. A and B. V, "Evaluating application scenarios with react native: A comparative study between native and react native development," 2016.