

**ANALISIS PERBANDINGAN JETPACK COMPOSE DAN
FLUTTER DALAM PENGEMBANGAN APLIKASI BERBASIS
ANDROID MENGGUNAKAN *TECHNICAL DOMAIN***

SKRIPSI

Untuk memenuhi salah satu syarat memperoleh gelar Sarjana S-1
Program Studi Informatika



Disusun Oleh:

AHMAD HARIS RIFANI

19106050012

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2023



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI

Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-1841/Un.02/DST/PP.00.9/07/2023

Tugas Akhir dengan judul : ANALISIS PERBANDINGAN JETPACK COMPOSE DAN FLUTTER DALAM PENGEMBANGAN APLIKASI BERBASIS ANDROID MENGGUNAKAN TECHNICAL DOMAIN

yang dipersiapkan dan disusun oleh:

Nama : AHMAD HARIS RIFANI
Nomor Induk Mahasiswa : 19106050012
Telah diujikan pada : Senin, 17 Juli 2023
Nilai ujian Tugas Akhir : A

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR



Ketua Sidang

Mandahadi Kusuma, M.Eng.
SIGNED

Valid ID: 64b88c36f32ee



Penguji I

Muhammad Mustakim, S.T. M.T.
SIGNED

Valid ID: 64c21983bacc5e



Penguji II

Dwi Otik Kurniawati, M.Eng.
SIGNED

Valid ID: 64c21127b8e83



Yogyakarta, 17 Juli 2023
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi
Dr. Dra. Hj. Khurul Wandati, M.Si.
SIGNED

Valid ID: 64c388e6d2482

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Ahmad Haris Rifani

NIM : 19106050012

Program Studi : Informatika

Fakultas : Sains dan Teknologi

menyatakan bahwa skripsi saya yang berjudul “**Analisis Perbandingan Jetpack Compose dan Flutter dalam Pengembangan Aplikasi Berbasis Android Menggunakan *Technical Domain***” merupakan hasil penelitian saya sendiri, tidak terdapat pada karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi, dan bukan plagiasi karya orang lain kecuali bagian-bagian tertentu yang penulis ambil sebagai acuan dan disebutkan dalam daftar pustaka.

Yogyakarta, 19 Juni 2023

Yang menyatakan,

STATE ISLAMIC UNIVERSITY
SUNAN KAHMAD
YOGYAKARTA



Ahmad Haris Rifani

NIM. 19106050012

PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi/Tugas Akhir

Lamp : -

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka saya selaku dosen pembimbing berpendapat bahwa skripsi Saudara:

Nama : Ahmad Haris Rifani

NIM : 19106050012

Judul Skripsi : Analisis Perbandingan Jetpack Compose dan Flutter dalam Pengembangan Aplikasi Berbasis Android Menggunakan *Technical Domain*

sudah dapat diajukan kepada Program Studi Informatika dan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Informatika.

Dengan ini saya berharap agar skripsi/tugas akhir Saudara tersebut dapat segera di-munaqosyah-kan. Atas perhatiannya saya ucapkan terimakasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 19 Juni 2023

Pembimbing,



Mandahadi Kusuma, M.Eng.

NIP. 19841115 201903 1 003

MOTTO

”Life goes on. Let’s Live on”

سافر تجد عوضا عن تفارقه . وانصب فإن لذيق العيش في النصب

”Merantaulah, maka kau akan menemukan pengganti dari apa yang kau tinggalkan. Berlelah-lelahlah. Karena nikmatnya hidup terasa setelah lelah berjuang” – Imam Syafi’i



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

HALAMAN PERSEMBAHAN

Saya persembahkan tugas akhir ini kepada kedua orang tua saya yang telah sabar dan tak henti-hentinya mendoakan saya, serta senantiasa mencari jalan untuk mendidik dan merawat saya hingga detik ini.



KATA PENGANTAR



Puji syukur kehadiran Allah SWT, karena atas rahmat-Nya, penulis dapat menyelesaikan penyusunan skripsi ini dengan judul "Analisis Perbandingan Jetpack Compose dan Flutter dalam Pengembangan Aplikasi Berbasis Android Menggunakan *Technical Domain*" sebagai salah satu syarat untuk memperoleh gelar Sarjana S-1 pada program studi Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga Yogyakarta.

Penulis menyadari bahwa penyusunan skripsi ini tidak mungkin terwujud tanpa dukungan dan bantuan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis ingin mengucapkan terima kasih yang setulus-tulusnya kepada:

1. Keluarga, terkhusus kedua orangtua; ayah, Arif Wardani, S.Hut., M.Si., dan ibu, Nahdiah, S.P., S.Pd., M.Pd., yang telah sabar dan terus mendoakan, memberikan kasih, perhatian serta dukungan yang tak terhingga dan tak terbilang;
2. Bapak Mandahadi Kusuma, M.Eng. selaku Dosen Pembimbing Skripsi yang telah memberikan bimbingan, arahan, serta masukan yang berharga sepanjang penulisan skripsi ini;
3. Bapak Ir. Sumarsono, S.T., M.Kom., selaku Dosen Penasehat Akademik yang telah membimbing dan mengarahkan penulis selama masa perkuliahan;
4. Ibu Ir. Maria Ulfah Siregar, S. Kom., MIT., Ph. D., selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta, serta seluruh dosen dan karyawan Program Studi Informatika UIN Sunan Kalijaga Yogyakarta yang telah memberi ilmu dan bantuan selama masa perkuliahan;
5. Ibu Dra. Hj. Khurul Wardati, M.Si., selaku Dekan Fakultas Sains dan Teknologi;
6. Bapak Prof. Dr. Phil. Al Makin, S.Ag., MA., selaku Rektor Universitas Islam Negeri Sunan Kalijaga Yogyakarta.

7. Seluruh kawan Informatika angkatan 2019 yang telah menjadi teman diskusi, saling membantu serta memberikan dukungan hingga saat ini;
8. Semua pihak yang tidak bisa penulis sebutkan satu persatu dalam membantu menyelesaikan skripsi ini baik secara langsung maupun tidak langsung.

Akhir kata, penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik, saran, dan masukan yang membangun dari pembaca. Semoga skripsi ini dapat memberikan manfaat dan dapat mendorong untuk penelitian-penelitian selanjutnya.

Yogyakarta, 20 Juni 2023

Penulis,



Ahmad Haris Rifani

NIM. 19106050012

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

INTISARI

Pada saat ini, dalam pengembangan aplikasi android pada umumnya menggunakan bahasa pemrograman berbasis *native*, seperti Java ataupun Kotlin. Namun terdapat bahasa pemrograman lain yang memiliki *framework* yang dapat diaplikasikan pada lintas *platform*, seperti Dart. Berbeda dengan Dart yang memiliki *framework* Flutter agar dapat mengembangkan aplikasi android, Kotlin memerlukan integrasi dengan XML, sehingga cukup kompleks hanya untuk membuat satu tampilan. Maka atas dasar itulah Jetpack Compose dibuat. Meskipun tergolong teknologi yang baru dirilis, Jetpack Compose telah banyak digunakan oleh banyak perusahaan besar seperti Twitter hingga Pinterest, yang berarti secara kualitas sudah cukup mumpuni. Oleh karena itu, perlunya penelitian untuk membandingkan kedua *framework* tersebut, yang mana menggunakan *technical domain* (domain perbandingan) sebagai instrumen yang membantu dalam perbandingan.. Penelitian ini bertujuan untuk mendapatkan hasil berdasarkan pengujian dengan melakukan perbandingan berdasarkan *performance* dan *testability* dengan studi kasus aplikasi kalkulator dan aplikasi daftar film. Pada domain *performance*, yang dibandingkan adalah berdasarkan parameter ukuran file aplikasi, konsumsi CPU dan konsumsi memori. Sedangkan pada domain *testability*, yang dibandingkan adalah hasil implementasi *unit testing*.

Setelah dilakukan penelitian, didapatkan hasil bahwa terdapat perbedaan hasil pada kedua jenis aplikasi. Dalam domain *performance*, Flutter lebih unggul dalam parameter konsumsi CPU, baik pada aplikasi kalkulator maupun aplikasi daftar film. Sedangkan Jetpack Compose lebih unggul dalam parameter konsumsi memori, baik pada aplikasi kalkulator maupun aplikasi daftar film. Adapun dalam parameter ukuran aplikasi, Jetpack Compose lebih unggul pada aplikasi daftar film, sedangkan Flutter lebih unggul pada aplikasi kalkulator. Dalam domain *testability*, baik Jetpack Compose dan Flutter tidak terdapat hasil yang berbeda pada pengujian *unit testing*. Yang membedakan hanyalah pada durasi pengujian. Setelah dilakukan pengujian, didapatkan hasil bahwa Jetpack Compose lebih unggul dalam durasi *unit testing* pada aplikasi kalkulator walaupun berdasarkan standar deviasi lebih tinggi namun durasi tertingginya tidak lebih dari durasi *unit testing* yang dimiliki oleh Flutter serta selisih durasi antar tiap pengujian tidak terlalu lebar. Adapun Flutter lebih unggul dalam durasi *unit testing* pada aplikasi daftar film walaupun berdasarkan standar deviasi lebih tinggi karena selisih durasi antar tiap pengujian cukup lebar. Berdasarkan analisis, perbedaan hasil pada kedua jenis aplikasi dapat disebabkan beberapa faktor, seperti mesin render yang dipakai maupun teknologi kompilasi.

Kata kunci: Android, Framework, Domain, Jetpack Compose, Flutter, Performa, Testabilitas

ABSTRACT

Currently, android application development generally uses native-based programming languages, such as Java or Kotlin. However, there are other programming languages that have frameworks that can be applied across platforms, such as Dart. Unlike Dart which has the Flutter framework in order to develop android applications, Kotlin requires integration with XML, making it quite complex just to create one view. So on that basis Jetpack Compose was created. Although it is a newly released technology, Jetpack Compose has been widely used by many large companies such as Twitter to Pinterest, which means that the quality is quite capable. Therefore, there is a need for research to compare the two frameworks, which uses the technical domain (comparison domain) as an instrument that helps in comparison. This research aims to get results based on testing by comparing based on performance and testability with case studies of calculator applications and movie list applications. In the performance domain, what is compared is based on the parameters of application file size, CPU consumption and memory consumption. While in the testability domain, what is compared is the result of unit testing implementation.

After the research, it was found that there are differences in the results of both types of applications. In the performance domain, Flutter is superior in CPU consumption parameters, both in calculator applications and movie list applications. While Jetpack Compose is superior in memory consumption parameters, both in calculator applications and movie list applications. As for the application size parameter, Jetpack Compose is superior to the movie list application, while Flutter is superior to the calculator application. In the testability domain, both Jetpack Compose and Flutter have no different results in unit testing. The only difference is the duration of the test. After testing, the results show that Jetpack Compose is superior in the duration of unit testing on calculator applications even though based on a higher standard deviation but the highest duration is not more than the duration of unit testing owned by Flutter and the difference in duration between each test is not too wide. Meanwhile, Flutter is superior in the duration of unit testing on movie list applications even though based on standard deviation it is higher because the difference in duration between each test is quite wide. Based on the analysis, the difference in results between the two types of applications can be caused by several factors, such as the rendering engine used or the compilation technology.

Keywords: Android, Framework, Domain, Jetpack Compose, Flutter, Performance, Testability

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
PERNYATAAN KEASLIAN SKRIPSI.....	iii
PERSETUJUAN SKRIPSI/TUGAS AKHIR	iv
MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
INTISARI.....	ix
<i>ABSTRACT</i>	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	4
2.1 Tinjauan Pustaka	4
2.2 Landasan Teori	11
2.2.1 Android	11
2.2.2 Kotlin	11
2.2.3 Jetpack Compose.....	12

2.2.4	Dart.....	12
2.2.5	Flutter	13
2.2.6	Domain.....	13
BAB III METODE PENELITIAN.....		15
3.1	Alur Penelitian.....	15
3.2	Metode Penelitian.....	16
3.3	Alat dan Bahan Penelitian	16
3.4	Analisis Kebutuhan	17
3.5	Studi Kasus.....	17
3.6	Teknis Perbandingan	18
BAB IV HASIL DAN PEMBAHASAN		19
4.1	Tahap Implementasi	19
4.1.1	Tampilan Aplikasi Kalkulator.....	19
4.1.2	Tampilan Aplikasi Daftar Film	19
4.2	Hasil Perbandingan.....	20
4.2.1	<i>Performance</i>	20
4.2.2	<i>Testability</i>	24
4.3	Analisis Hasil Perbandingan.....	30
BAB V PENUTUP.....		32
5.1	Kesimpulan.....	32
5.2	Saran	33
DAFTAR PUSTAKA		34
LAMPIRAN.....		36
<i>CURRICULUM VITAE</i>		49

DAFTAR TABEL

Tabel 2.1 Ringkasan Tinjauan Pustaka	7
Tabel 2.2 <i>Technical Domain</i>	13
Tabel 4.1 Perbandingan Rata-rata Hasil Pengujian <i>Performance</i>	24
Tabel 4.2 Perbandingan Durasi Pengujian <i>Testability</i>	29



DAFTAR GAMBAR

Gambar 3.1 Bagan Alur Penelitian	15
Gambar 3.2 Bagan Alur Teknis Perbandingan	18
Gambar 4.1 Tampilan Aplikasi Kalkulator.....	19
Gambar 4.2 Tampilan Aplikasi Daftar Film	20
Gambar 4.3 Tampilan Awal Apptim.....	21
Gambar 4.4 Tampilan Jenis Ponsel yang Terhubung	21
Gambar 4.5 Daftar Aplikasi yang Terunduh pada Ponsel.....	21
Gambar 4.6 Pilihan Waktu Pengujian.....	22
Gambar 4.7 Keterangan Pengujian	22
Gambar 4.8 Tampilan Ketika Pengujian.....	22
Gambar 4.9 Tampilan Pasca Pengujian	23
Gambar 4.10 Tampilan Hasil Pengujian.....	23
Gambar 4.11 Tampilan Riwayat Pengujian	23
Gambar 4.12 Deklarasi JUnit pada Jetpack Compose	25
Gambar 4.13 Deklarasi Test pada Flutter	25
Gambar 4.14 Kode <i>Unit Testing</i> Aplikasi Daftar Film pada Jetpack Compose ...	27
Gambar 4.15 Kode <i>Unit Testing</i> Aplikasi Daftar Film pada Flutter.....	28
Gambar 4.16 Perbandingan Hasil Unit Testing Aplikasi Kalkulator.....	29
Gambar 4.17 Perbandingan Hasil Unit Testing Aplikasi Daftar Film	29

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di dalam kehidupan sehari-hari, ponsel (*mobile phone*) telah menjadi kebutuhan banyak orang. Pada Januari 2023 saja terdapat lebih dari dua pertiga dari total populasi dunia yang menggunakan ponsel dengan jumlah mencapai 5.44 miliar, yang mana 4 dari 5 ponsel merupakan ponsel pintar (*smartphone*)¹.

Dengan maraknya penggunaan ponsel pintar tentu berdampak dengan beragamnya *operating system* yang ada, salah satunya adalah android. Android itu sendiri adalah *operating system* terpopuler pada pasar *mobile* dengan persentase 71.74% pada Januari 2023².

Dalam suatu sistem operasi, tentu tak lengkap tanpa adanya aplikasi, yang notabene menggunakan bahasa pemrograman untuk mengembangkannya. Pada saat ini, dalam pengembangan aplikasi android pada umumnya menggunakan bahasa pemrograman berbasis *native*, seperti Java ataupun Kotlin. Namun terdapat bahasa pemrograman lain yang memiliki *framework* yang dapat diaplikasikan pada lintas *platform*, seperti Dart.

Dilihat dari segi pemakaian, memang bahasa pemrograman *native* android seperti Java maupun Kotlin lebih banyak digunakan daripada bahasa pemrograman lintas *platform* seperti Dart. Namun jika dilihat dari persentase pada hasil survei Stackoverflow pada tahun 2022, selisih antara Kotlin dan Dart tidak terlalu jauh, hanya terpaut 2,62%³, yang menandakan secara kualitas Dart dapat bersaing dengan bahasa *native* seperti Kotlin.

¹ DataReportal. (2023). *Digital Around The World*. Retrieved February 7, 2023, from <https://datareportal.com/global-digital-overview>

² StatCounter. (2023). *Mobile Operating System Market Share Worldwide - January 2023*. Retrieved February 7, 2023, from <https://gs.statcounter.com/os-market-share/mobile/worldwide>

³ StackOverflow. (2022). *2022 Developer Survey*. Retrieved March 14, 2023, from <https://survey.stackoverflow.co/2022/>

Berbeda dengan Dart yang memiliki *framework* Flutter agar dapat mengembangkan aplikasi android, Kotlin memerlukan integrasi dengan XML, sehingga cukup kompleks hanya untuk membuat satu tampilan. Maka atas dasar itulah Jetpack Compose dibuat.

Jetpack Compose sendiri merupakan *framework* atau *toolkit* yang dibuat untuk memudahkan dalam pengembangan aplikasi android dengan hanya mengandalkan satu basis kode yaitu Kotlin. Jetpack Compose memiliki sifat seperti Flutter, yaitu *UI Declarative*, sehingga secara penulisan kode terdapat kemiripan.

Meskipun tergolong teknologi yang baru dirilis, Jetpack Compose telah banyak digunakan oleh banyak perusahaan besar seperti Twitter hingga Pinterest, yang berarti secara kualitas sudah cukup mumpuni. Walaupun begitu, Jetpack Compose terbilang belum terlalu stabil, karena ada beberapa kekurangan dalam kondisi tertentu, seperti ketergantungan pada pembaruan dependensi yang sering terjadi, dan beberapa fitur yang sering dihapus dan diganti. Berbeda dengan Flutter yang sudah cukup stabil mengingat telah rilis beberapa tahun lebih awal daripada Jetpack Compose.

Oleh karena itu, perlunya penelitian untuk membandingkan kedua *framework* tersebut, mengingat walaupun Jetpack Compose terbilang masih baru namun sudah dapat menyaingi kualitas dari Flutter. Adapun penelitian ini difokuskan pada analisis perbandingan terhadap pengembangan aplikasi berbasis android dan menggunakan *technical domain* sebagai instrumen yang membantu dalam perbandingan.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dalam penelitian ini yang menjadi rumusan masalah adalah bagaimana mendapatkan hasil perbandingan antara *framework* Jetpack Compose dengan *framework* Flutter dengan melakukan pengujian menggunakan *technical domain* sebagai instrumen yang membantu dalam perbandingan.

1.3 Batasan Masalah

Pada penelitian ini agar pembahasan tidak terlalu luas, maka penulis membatasi penulisan yang dibuat sebagai berikut:

- a. Membangun dua aplikasi Android masing-masing menggunakan *framework* Jetpack Compose dan *framework* Flutter, dengan kriteria memiliki tampilan yang serupa.
- b. Aplikasi yang dikembangkan berbasis android, dengan studi kasus aplikasi kalkulator dan aplikasi daftar film.
- c. Perbandingan difokuskan pada dua domain, yaitu:
 - 1) *Performance*, dengan parameter yang dibandingkan adalah ukuran file aplikasi, konsumsi CPU dan konsumsi memori.
 - 2) *Testability*, dengan melakukan perbandingan atas hasil implementasi *unit testing* pada Flutter dan Jetpack Compose

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mendapatkan hasil berdasarkan pengujian dengan melakukan perbandingan menggunakan *technical domain* sebagai instrumen yang membantu dalam perbandingan.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut:

- a. Mengetahui hasil perbandingan antara Jetpack Compose dan Flutter dalam pengembangan aplikasi android berdasarkan *performance* dan *testability*.
- b. Hasil penelitian dapat dijadikan referensi bagi para pengembang aplikasi android dalam menentukan teknologi yang unggul dalam aspek tertentu sesuai dengan kebutuhan.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, diperoleh kesimpulan sebagai berikut:

1. Dalam domain *performance*, Flutter lebih unggul dalam parameter konsumsi CPU, baik pada aplikasi kalkulator maupun aplikasi daftar film. Sedangkan Jetpack Compose lebih unggul dalam parameter konsumsi memori, baik pada aplikasi kalkulator maupun aplikasi daftar film. Adapun dalam parameter ukuran aplikasi, Jetpack Compose lebih unggul pada aplikasi daftar film, sedangkan Flutter lebih unggul pada aplikasi kalkulator.
2. Dalam domain *testability*, baik Jetpack Compose dan Flutter tidak terdapat hasil yang berbeda pada pengujian *unit testing*. Yang membedakan hanyalah pada durasi pengujian. Setelah dilakukan pengujian, didapatkan hasil bahwa Jetpack Compose lebih unggul dalam durasi *unit testing* pada aplikasi kalkulator walaupun berdasarkan standar deviasi lebih tinggi namun durasi tertingginya tidak lebih dari durasi *unit testing* yang dimiliki oleh Flutter serta selisih durasi antar tiap pengujian tidak terlalu lebar. Adapun Flutter lebih unggul dalam durasi *unit testing* pada aplikasi daftar film walaupun berdasarkan standar deviasi lebih tinggi karena selisih durasi antar tiap pengujian cukup lebar.
3. Terdapat beberapa faktor yang menyebabkan perbedaan hasil pada kedua aplikasi, seperti:
 - a) Mesin render: Flutter menggunakan mesin rendering khusus yang disebut Skia, sedangkan Jetpack Compose menggunakan mesin rendering yang terintegrasi dengan Android. Skia dikenal dengan performa tinggi dan kemampuannya untuk merender grafis yang kompleks dengan cepat. Hal ini dapat memberikan keuntungan performa untuk aplikasi Flutter, terutama ketika menampilkan grafik yang kompleks.

- b) Kompilasi: Aplikasi Flutter dikompilasi menjadi kode mesin, sedangkan Jetpack Compose yang notabene merupakan bagian dari Kotlin, sehingga aplikasinya dikompilasi menjadi Java *bytecode* yang berjalan di atas Java *Virtual Machine* (JVM).
4. Jetpack Compose merupakan *framework* yang masih baru, sehingga tidak heran masih berbagai aspek yang dapat dikembangkan. Kelebihan dari Jetpack Compose adalah merupakan bagian dari Android Jetpack yang terintegrasi langsung dengan ekosistem android. Sehingga jika ingin membangun aplikasi khusus android, pada umumnya disarankan menggunakan Jetpack Compose, walaupun secara kualitas Flutter juga dapat menyaingi teknologi yang berbasis untuk *native*. Namun karena keduanya memiliki kelebihan dan kekurangan masing-masing, maka tidak dapat dikatakan bahwa salah satu lebih unggul daripada yang lain. Jika pengembang lebih terbiasa dengan bahasa Kotlin dan ingin menggunakan teknologi yang baru, maka Jetpack Compose mungkin menjadi pilihan yang lebih baik. Namun, jika pengembang ingin fokus pada kinerja dan kecepatan pengembangan serta efektifitas biaya, maka Flutter dapat menjadi pilihan yang lebih baik.

5.2 Saran

Penelitian ini masih jauh dari kata sempurna. Oleh karena itu, disarankan untuk penelitian selanjutnya memperhatikan hal-hal sebagai berikut:

1. Penelitian pada domain *performance* dapat menambahkan parameter lain, seperti *FPS* dan konsumsi data.
2. Penelitian pada domain *testability* dapat menambahkan metode lain, seperti *integration testing*.
3. Implementasi *testability* dapat menggunakan *library* yang lain.
4. Aplikasi yang dikembangkan dapat lebih kompleks dengan menggunakan berbagai macam *library*.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, diperoleh kesimpulan sebagai berikut:

1. Dalam domain *performance*, Flutter lebih unggul dalam parameter konsumsi CPU, baik pada aplikasi kalkulator maupun aplikasi daftar film. Sedangkan Jetpack Compose lebih unggul dalam parameter konsumsi memori, baik pada aplikasi kalkulator maupun aplikasi daftar film. Adapun dalam parameter ukuran aplikasi, Jetpack Compose lebih unggul pada aplikasi daftar film, sedangkan Flutter lebih unggul pada aplikasi kalkulator.
2. Dalam domain *testability*, baik Jetpack Compose dan Flutter tidak terdapat hasil yang berbeda pada pengujian *unit testing*. Yang membedakan hanyalah pada durasi pengujian. Setelah dilakukan pengujian, didapatkan hasil bahwa Jetpack Compose lebih unggul dalam durasi *unit testing* pada aplikasi kalkulator walaupun berdasarkan standar deviasi lebih tinggi namun durasi tertingginya tidak lebih dari durasi *unit testing* yang dimiliki oleh Flutter serta selisih durasi antar tiap pengujian tidak terlalu lebar. Adapun Flutter lebih unggul dalam durasi *unit testing* pada aplikasi daftar film walaupun berdasarkan standar deviasi lebih tinggi karena selisih durasi antar tiap pengujian cukup lebar.
3. Terdapat beberapa faktor yang menyebabkan perbedaan hasil pada kedua aplikasi, seperti:
 - a) Mesin render: Flutter menggunakan mesin rendering khusus yang disebut Skia, sedangkan Jetpack Compose menggunakan mesin rendering yang terintegrasi dengan Android. Skia dikenal dengan performa tinggi dan kemampuannya untuk merender grafis yang kompleks dengan cepat. Hal ini dapat memberikan keuntungan performa untuk aplikasi Flutter, terutama ketika menampilkan grafik yang kompleks.

- b) Kompilasi: Aplikasi Flutter dikompilasi menjadi kode mesin, sedangkan Jetpack Compose yang notabene merupakan bagian dari Kotlin, sehingga aplikasinya dikompilasi menjadi Java *bytecode* yang berjalan di atas Java *Virtual Machine* (JVM).
4. Jetpack Compose merupakan *framework* yang masih baru, sehingga tidak heran masih berbagai aspek yang dapat dikembangkan. Kelebihan dari Jetpack Compose adalah merupakan bagian dari Android Jetpack yang terintegrasi langsung dengan ekosistem android. Sehingga jika ingin membangun aplikasi khusus android, pada umumnya disarankan menggunakan Jetpack Compose, walaupun secara kualitas Flutter juga dapat menyaingi teknologi yang berbasis untuk *native*. Namun karena keduanya memiliki kelebihan dan kekurangan masing-masing, maka tidak dapat dikatakan bahwa salah satu lebih unggul daripada yang lain. Jika pengembang lebih terbiasa dengan bahasa Kotlin dan ingin menggunakan teknologi yang baru, maka Jetpack Compose mungkin menjadi pilihan yang lebih baik. Namun, jika pengembang ingin fokus pada kinerja dan kecepatan pengembangan serta efektifitas biaya, maka Flutter dapat menjadi pilihan yang lebih baik.

5.2 Saran

Penelitian ini masih jauh dari kata sempurna. Oleh karena itu, disarankan untuk penelitian selanjutnya memperhatikan hal-hal sebagai berikut:

1. Penelitian pada domain *performance* dapat menambahkan parameter lain, seperti *FPS* dan konsumsi data.
2. Penelitian pada domain *testability* dapat menambahkan metode lain, seperti *integration testing*.
3. Implementasi *testability* dapat menggunakan *library* yang lain.
4. Aplikasi yang dikembangkan dapat lebih kompleks dengan menggunakan berbagai macam *library*.

DAFTAR PUSTAKA

- Barbacci, M. (2004). *Software quality attributes: Modifiability and usability*. In Software Engineering Institute, Carnegie Mellon [ieee.org.ar. http://www.ieee.org.ar/downloads/Barbacci-05-notas1.pdf](http://www.ieee.org.ar/downloads/Barbacci-05-notas1.pdf)
- Bourezeg, M. (2023, March 16). *Jetpack Compose vs Flutter (With Examples)*. Netguru. Retrieved June 9, 2023, from <https://www.netguru.com/blog/jetpack-compose-versus-flutter>
- DataReportal. (2023). *Digital Around The World*. Retrieved February 7, 2023, from <https://datareportal.com/global-digital-overview>
- Dicoding. (2020). *Memulai Pemrograman Dengan Dart*. Retrieved February 7, 2023, from <https://www.dicoding.com/academies/191>
- Gerdessen, A. (2007). *Framework comparison method Comparing two frameworks based on technical domains, focussing on customisability and modifiability*. <https://homepages.cwi.nl/~paulk/thesesMasterSoftwareEngineering/2007/AntonGerdessen.pdf>
- Hussain, H., Khan, K., Farooqui, F., Ali Arain, Q., & Farah Siddiqui, I. (2021). *Comparative Study of Android Native and Flutter App Development*. <https://www.researchgate.net/publication/361208165>
- Khanna, I. (2022, November 15). *Flutter vs Jetpack Compose: The Battle of the Decade*. Retrieved June 9, 2023, from <https://blog.droidchef.dev/flutter-vs-jetpack-compose-the-battle-of-the-century/>
- Olsson, M. (2020). *A Comparison of Performance and Looks Between Flutter and Native Applications: When to prefer Flutter over native in mobile application development*. <http://urn.kb.se/resolve?urn=urn:nbn:se:bth-19712>
- Prasetyo, M. A. (2022). *Perbandingan Jetpack Compose Dan Flutter Untuk Pengembangan Layout Android Thukul Menggunakan Pengukuran Qualitative Weight and Sum (QWS)*. [etd.umm.ac.id. https://etd.umm.ac.id/id/eprint/449/](https://etd.umm.ac.id/id/eprint/449/)
- Pinedamg. (2022, February 9). *How Flutter & Dart Code Gets Compiled To Native Apps*. Medium. Retrieved June 9, 2023, from

<https://pinedamg.medium.com/how-flutter-dart-code-gets-compiled-to-native-apps-c4612ea0ef0e>

Pressman, R. S. (2000). *Software engineering: A practitioner's approach* (5th ed). McGraw Hill.

Raharjo, B. (2019). *Pemrograman Android Dengan Flutter*. Bandung: Penerbit Informatika.

Riyadi Hermawan. (2022). *Pengertian Android Beserta Sejarah, Kelebihan dan Kekurangannya*, Nesabamedia. Retrieved May 26, 2022, from <https://www.nesabamedia.com/pengertian-android-beserta-kelebihan-dan-kekurangannya/>

Rohman, N., & Toro, R. (2018). *Kotlin Android Developer Expert*. Bandung: PT. Presentologics.

StackOverflow. (2022). *2022 Developer Survey*. Retrieved March 14, 2023, from <https://survey.stackoverflow.co/2022/>

StatCounter. (2023). *Mobile Operating System Market Share Worldwide - January 2023*. Retrieved February 7, 2023, from <https://gs.statcounter.com/os-market-share/mobile/worldwide>

Yumusak, Can. (2022, July 9). *Explained: Compose Compiler and Runtime*. Speaker Deck. Retrieved June 9, 2023, from <https://speakerdeck.com/canyudev/explained-compose-compiler-and-runt>

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA