

## **SKRIPSI**

**ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN  
*FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS  
*MOBILE***

***ANALYSIS OF THE LEVEL OF EFFECTIVENESS BETWEEN REACT  
NATIVE AND FLUTTER IN DEVELOPING MOBILE-BASED TODO LIST  
APPLICATIONS***

**MUHAMMAD KIDFARI**

**192941021**



**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER  
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI MAKASSAR  
TAHUN 2024**



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS NEGERI MAKASSAR  
FAKULTAS TEKNIK

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

Alamat: Jl. Daeng Tata Raya Parangtambung Makassar - 90224  
Telp 0411-864935, Fax 0411 - 861507, HP 0853-1122-4040, Email jtk@unm.ac.id | Laman jtk.ft.unm.ac.id

LEMBAR PENGESAHAN TUTUP

Yang bertanda tangan di bawah ini, dengan ini menerangkan bahwa mahasiswa tersebut  
di bawah ini:

Nama : MUJI, KIDFARI  
NIM : 1929041021  
Program Studi : Pendidikan Teknik Informatika dan Komputer - (S1)  
Jurusan : Teknik Informatika dan Komputer  
Judul : Analisis Tingkat Efektifitas antara React Native dan Flutter dalam  
Pembuatan Aplikasi To Do List Berbasis Mobile

Setelah skripsi yang disusun oleh mahasiswa tersebut kami periksa, maka dinyatakan  
memenuhi syarat untuk melaksanakan ujian skripsi.

Makassar, 20 Februari 2024

Pembimbing II,

Pembimbing I

Prof. Dr. Ir. H. Muhammad Yahya, M.Kes., M.Eng.  
IPU.ASEAN.Eng  
NIP. 19630623 199103 1 002

Dr. Ir. Ridwan Daud Mahande, S.Pd., M.Pd., HPM.  
NIP. 19840804 201903 1 005

Mengetahui,



Dr. Ir. Mustari S. Lamada, S.Pd. M.T.  
NIP. 19750305 200501 1 001

Ketua Program Studi  
Pendidikan Teknik Informatika dan Komputer - (S1)

Fathahillah, S.Pd., M.Eng.  
NIP. 19860326 201504 1 001



KEMENTERIAN PENDIDIKAN DAN  
KEBUDAYAAN, RISET, DAN TEKNOLOGI  
UNIVERSITAS NEGERI MAKASSAR  
FAKULTAS TEKNIK

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

Alamat: Jl. Duri, Tata Raya Patangpanjang, Makassar - 90224  
Telp. 0411-864935, Fax. 0411 - 861507, HP. 0853-1122-4040, Email: jtk@unm.ac.id | Laman: tktk.unm.ac.id

LEMBAR PENGESAHAN HASIL

Yang bertanda tangan di bawah ini, dengan ini menerangkan bahwa mahasiswa tersebut  
di bawah ini:

Nama : Muh Kifdari  
NIM : 1929041021  
Program Studi : Pendidikan Teknik Informatika dan Komputer - (S1)  
Jurusan : Teknik Informatika dan Komputer  
Judul : Analysis Tingkat Efektifitas Antara React Native Dan Flutter  
Dalam Pembuatan Aplikasi ToDo List Berbasis Mobile

Setelah laporan hasil penelitian yang disusun oleh mahasiswa tersebut kami periksa, maka  
dinyatakan memenuhi syarat untuk melaksanakan seminar hasil penelitian.

Pembimbing I,

Prof. Dr. Ir. H. Muhammad Yahya  
M.Kes., M.Eng., JPU., ASEAN Eng.  
NIP. 19781103 201012 1 002

Makassar, 09 Oktober 2023

Pembimbing II,

Dr. Ir. Ridwan Daud Mahantie, S.Pd.,  
M.Pd., IPM  
NIP. 19750720 201012 2 001

Mengetahui,



Ketua Program Studi  
Pendidikan Teknik Informatika dan Komputer - (S1)

Fatkhulillah, S.Pd., M.Eng.  
NIP. 19860326 201504 1 001





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS NEGERI MAKASSAR  
FAKULTAS TEKNIK

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

Alamat: JL. Daeng Tata Raya Parangtambung Makassar – 90224

Telp. 0411-864935, Fax. 0411 – 861507, HP. 0853-1122-4040, Email: jtik@unm.ac.id | Laman tik.ft.unm.ac.id

LEMBAR PENGESAHAN  
REVISI HASIL PENELITIAN

Yang bertanda tangan di bawah ini, dengan ini menerangkan bahwa mahasiswa tersebut di bawah ini:

Nama : Muh Kidfari  
NIM : 1929041021  
Program Studi : Pendidikan Teknik Informatika dan Komputer  
Jurusan : Teknik Informatika dan Komputer – S1  
Judul : Analysis Tingkat Efektifitas Antara React Native Dan Flutter Dalam Pembuatan Aplikasi ToDo List Berbasis Mobile

Setelah laporan proposal yang disusun oleh mahasiswa tersebut kami periksa, maka dinyatakan memenuhi syarat untuk melaksanakan **Ujian Tutup**.

Makassar, 07 Februari 2024

Pembimbing 1,

Prof. Dr. Ir. H. Muhammad Yahya  
M.Kes., M.Eng., JPU., ASEAN Eng.  
NIP.19630623 199103 1 002

Pembimbing 2,

Dr. Ir. Ridwan Daud Mahande. S.Pd.,  
M.pd., IPM  
NIP.19840804 201903 1 005

Penanggap 1

Fathahillah, S.Pd., M.Eng.  
NIP.19860326 201504 1 001

Penanggap 2

Ninik Rahyu Ashadi , S.Pd., M.Pd.  
NIP.19910728 201903 2 014

Mengetahui,



Ketua Program Studi  
Pendidikan Teknik Informatika dan Komputer

Dr. Ir. Mustaqi S. Lamada, S.Pd. M.T.  
NIP. 19750505 200501 1 001

## **MOTTO DAN PERSEMBAHAN**

**“Berdiri sama tinggi duduk sama rendah ketawa sama sama”**

*Kepada Ayah, Kak, almarhum ibu, keluarga tercinta, kekasih, dan teman-temanku yang senantiasa memberikan dukungan tiada henti-hentinya, dengan penuh rasa syukur, penulis mempersembahkan hasil jerih payah dalam penulisan skripsi ini. Setiap langkah yang penulis ambil tidak akan berarti tanpa kehadiran, dorongan, dan cinta kalian. Terima kasih atas doa, semangat, dorongan serta bimbingan yang tak ternilai. Semoga dedikasi ini menjadi bukti kebersamaan kita dalam perjalanan hidup. Sukses penulis adalah berkat kalian semua*

## ABSTRAK

**Muh.kidfari, 2024.** *Analisis Tingkat Efektivitas Antara React Native dan Flutter Dalam pembuatan Aplikasi To Do List Berbasis Mobile.* Universitas Negeri Makassar (dibimbing oleh Muhammad Yahya dan Ridwan Daud Mahande).

Penelitian ini bertujuan untuk mengetahui tingkat efektivitas *React Native* dan *Flutter* dalam pengembangan aplikasi *To-Do List* berbasis *Mobile*. Metode penelitian yang digunakan adalah Studi Kasus dengan enam tahapan, meliputi identifikasi masalah, Implementasi, Pengujian, Pengumpulan, Analisis, dan presentasi Hasil Penelitian. Subjek penelitian terdiri dari 10 orang pengembang yang bergabung dalam organisasi KeDai Computerworks. Penelitian ini secara menyeluruh Memanfaatkan Standar ISO 9126 untuk mengevaluasi *fungsionalitas*, *portabilitas*, *efisiensi*, dan *usability* yang terintegrasi dalam kedua framework tersebut. Hasil evaluasi menunjukkan bahwa baik *React Native* maupun *Flutter* berhasil memenuhi persyaratan aplikasi. Pengujian fungsionalitas menghasilkan skor sempurna 100%, menunjukkan kinerja optimal dalam menjalankan seluruh fitur aplikasi. Uji portabilitas menegaskan kemampuan di mana aplikasi dapat berjalan tanpa kesalahan pada berbagai sistem operasi dan resolusi layar. Evaluasi efisiensi mengungkap keunggulan *Flutter* dalam responsivitas dibandingkan dengan *React Native* dengan nilai perbandingan antara *Flutter* (1.44) dan *React Native* (1.43). Penilaian *usability* menunjukkan bahwa *Flutter* lebih mudah dipelajari (Learnability: 87.5%) dibandingkan *React Native* (52.5%). *Flutter* unggul dalam Efisiensi menu (*Flutter* 75%, *React Native* 62.5%). Memorability lebih baik di *Flutter* (87.5%) daripada *React Native* (52.5%). Keseluruhan, kepuasan pengguna lebih tinggi di *Flutter* (87.5%) dibandingkan *React Native* (52.5%).

**Kata Kunci:** *Framework, React Native, Flutter, Mobile, Pengembang.*

## ABSTRACT

**Muh.kidfari, 2024.** Analysis of Effectiveness Levels Between React Native and Flutter in Developing Mobile-Based To-Do List Applications. State University of Makassar (supervised by Muhammad Yahya and Ridwan Daud Mahande).

This research aims to determine the level of effectiveness of React Native and Flutter in the development of Mobile-based To-Do List applications. The research method used is a case study with six stages, including problem identification, implementation, testing, collection, analysis, and presentation of research results. The research subjects consisted of 10 developers who joined the KeDai Computerworks organization. This research thoroughly utilized the ISO 9126 standard to evaluate the functionality, portability, efficiency, and usability integrated in both frameworks. The evaluation results show that both React Native and Flutter successfully fulfill the application requirements. Functionality testing resulted in a perfect score of 100%, indicating optimal performance in executing all of the app's features. The portability test confirmed the ability for the app to run without errors on various operating systems and screen resolutions. The efficiency evaluation revealed Flutter's superiority in responsiveness compared to React Native with a comparison score between Flutter (1.44) and React Native (1.43). The usability assessment shows that Flutter is easier to learn (Learnability: 87.5%) than React Native (52.5%). Flutter excels in menu Efficiency (Flutter 75%, React Native 62.5%). Memorability is better in Flutter (87.5%) than React Native (52.5%). Overall, user satisfaction is higher in Flutter (87.5%) than React Native (52.5%).

**Key words:** Framework, React Native, Flutter, Mobile, Developer.

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, yang telah memberikan berkat dan rahmat-Nya sehingga skripsi yang berjudul “Analisis tingkat efektivitas antara *react native* dan *Flutter* dalam pembuatan aplikasi *ToDo list* berbasis *mobile*” dapat diselesaikan dengan baik sebagai syarat mendapatkan gelar sarjana di program studi Pendidikan Teknik Informatika dan Komputer, Universitas Negeri Makassar.

Pertama-tama, penulis ingin mengucapkan terima kasih yang tak terhingga kepada kedua orang tua tercinta Alm Dra. Jahidah dan Arsono S. Sos., M. Si., karena telah melahirkan dan senantiasa memberikan dukungan moril, kasih sayang, serta doa restu dalam setiap langkah perjalanan akademis ini. Semua jerih payah ini adalah berkat kasih sayang dan dukungan tanpa batas dari keduanya. Semoga segala bakti dan kebahagiaan senantiasa menyertai keduanya. Terima kasih, Ibu dan Ayah, atas segala pengorbanan dan cinta yang telah diberikan.

Ucapan Terimakasih yang juga tak terhingga atas bimbingan dan dedikasi dari Bapak Prof. Ir, H. Muhammad Yahya, M.Kes., M.Eng., IPU., ASEAN Eng. dan Bapak Dr. Ir. Ridwan Daud Mahende. S.Pd., M.Pd., IPM, sebagai dosen pembimbing. Pengetahuan dan pengalaman yang Ibu dan Bapak berikan telah memberi penulis, landasan yang kokoh dalam perjalanan akademik penulis. Peneliti merasa beruntung dan berterima kasih atas waktu, energi, dan inspirasi yang telah diberikan.

Secara khusus, penulis juga menyampaikan ucapan terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Bapak Prof. Dr. Ir. Husain Syam, M.TP., IPU ASEAN Eng., sebagai Rektor Universitas Negeri Makassar yang telah menerima penulis sebagai mahasiswa di Fakultas Teknik Universitas Negeri Makassar.
2. Bapak Prof. Dr. Ir. H. Muhammad Yahya, M.Kes., IPU ASEAN Eng., sebagai Dekan Fakultas Teknik UNM; Prof. Dr. Ir. Jamaluddin.M.P. IPM., sebagai Wakil Dekan I; Dr. Ir. Ahmad Rifqi Asrib, M.T., sebagai Wakil Dekan II; Dr. Ir. Amiruddin, S.T., M.T.IPM., selaku Wakil Dekan III dan Drs. Ir. Faisal Syafar, M.Si., M.Inf.Tech., Ph.D. IPU, selaku Wakil Dekan IV yang telah memberikan dukungan akademik, administratif, dan kemahasiswaan selama proses pendidikan dan penyelesaian studi.
3. Bapak Dr. Ir. Mustari S. Lamada, M.T., yang merupakan Ketua Jurusan Teknik Informatika dan Komputer.
4. Ibu Dr. Sanatang, M.T., yang merupakan Sekretaris Jurusan Teknik Informatika dan Komputer, yang telah memberikan nasehat akademik, bimbingan dan arahan kepada penulis dari awal hingga akhir masa perkuliahan.
5. Bapak Fatahillah, S.Pd., M.Eng., yang merupakan Ketua Program Studi Pendidikan Teknik Informatika dan Komputer, juga sebagai penanggap yang telah memberikan saran perbaikan dan masukan yang membangun.
6. Bapak Dr. H. Aminuddin Bakry, M.S., selaku mantan pembimbing akademik yang telah memberikan nasehat akademik, bimbingan dan arahan kepada penulis selama masa perkuliahan.

7. Bapak dan Ibu Dosen Jurusan Teknik Informatika dan Komputer FT UNM yang telah memberikan bekal ilmu pengetahuan dan pengalaman selama dibangku perkuliahan.
  8. Keluarga besar KeDai Computerworks, yang yang telah mewadahi penulis untuk belajar, berekspresi, serta mengembangkan diri. Atas kerja sama dan dukungan dari berbagai pihak.
- penulis berdoa kepada Tuhan Yang Maha Esa, semoga semua bantuan yang diberikan mendapatkan balasan yang baik. Penulis berharap agar semua pihak dapat memberikan saran dan kritik yang konstruktif demi meningkatkan kualitas skripsi ini.

Makassar 19 januari 2024  
Penulis

**Muhammad Kidfari**

## DAFTAR ISI

<b>LEMBAR PENGESAHAN TUTUP .....</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN HASIL .....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN REVISI HASIL PENELITIAN .....</b>	<b>iv</b>
<b>MOTTO DAN PERSEMBAHAN.....</b>	<b>v</b>
<b>ABSTRAK.....</b>	<b>vi</b>
<b>ABSTRACT.....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiv</b>
<b>DAFTAR TABEL.....</b>	<b>xv</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
A. Latar Belakang.....	1
B. Rumusan Masalah.....	4
C. Tujuan Penelitian.....	4
D. Manfaat Penelitian.....	5
<b>BAB II KAJIAN TEORI.....</b>	<b>6</b>
A. Kajian Teori.....	6
1. Efektivitas.....	6
2. React Native.....	12
3. Flutter.....	14
4. Aplikasi ToDo List.....	16
5. Aplikasi Mobile.....	17

6. Aplikasi Native.....	17
7. Aplikasi Website.....	18
8. Hybrid Aplikasi.....	19
B. Penelitian Relevan.....	19
C. Kerangka Pikir.....	21
<b>BAB III METODE PENELITIAN.....</b>	<b>22</b>
A. Jenis Penelitian.....	22
B. Waktu dan Lokasi Penelitian.....	23
C. Populasi dan Sampel.....	23
D. Desain Penelitian.....	26
E. Prosedur Penelitian.....	27
F. Teknik Pengumpulan Data.....	28
G. Instrument Plot Usability.....	29
H. Teknik Analisis Data.....	31
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>	<b>36</b>
A. Hasil Penelitian.....	36
1. Fitur Aplikasi yang Dirancang.....	36
2. Pengkodean.....	40
3. Pengujian Sistem.....	41
4. Validasi Instrumen.....	43
5. Functionality Testing.....	46
6. Portability Testing.....	51
7. Efficiency Testing.....	53

8. Usability Testing.....	57
B. Pembahasan.....	62
<b>BAB V KESIMPULAN DAN SARAN.....</b>	<b>66</b>
A. Kesimpulan.....	66
B. Saran.....	66

**DAFTAR PUSTAKA****LAMPIRAN****RIWAYAT HIDUP**

## DAFTAR GAMBAR

Gambar 2. 1. <i>Framework React Native</i> .....	12
Gambar 2. 2. <i>Framework Flutter</i> .....	14
Gambar 2. 3. <i>Dart</i> .....	15
Gambar 2. 4. Kerangka Pikir .....	21
Gambar 3. 1. Desain Penelitian .....	27
Gambar 4. 1. <i>Login Screen</i> .....	36
Gambar 4. 2. Opsi <i>Login Google</i> .....	37
Gambar 4. 3. <i>To-Do Screen</i> .....	38
Gambar 4. 4. <i>Done Screen</i> .....	38
Gambar 4. 5. <i>Add To-do Screen</i> .....	39
Gambar 4. 6. <i>Update To-do Screen</i> .....	40
Gambar 4. 7. Proses pengembangan <i>Flutter</i> dari <i>Android Studio</i> .....	41
Gambar 4. 8. Proses pengembangan <i>React Native</i> dari <i>Android Studio</i> .....	41
Gambar 4. 9. Hasil Uji Sistem <i>Flutter</i> dari <i>Android Studio</i> .....	42
Gambar 4. 10. Hasil Uji Sistem <i>React Native</i> dari <i>Android Studio</i> .....	42
Gambar 4. 11. Diagram Hasil Validasi Instrumen .....	45
Gambar 4. 12. Diagram Hasil Analisa <i>Functionality</i> .....	48
Gambar 4. 13. Diagram Hasil Analisa <i>Potability</i> .....	52
Gambar 4. 14. Diagram Hasil Analisa <i>Efficiency</i> .....	55
Gambar 4. 15. Diagram Hasil Analisa <i>Usability</i> .....	61

## **DAFTAR TABEL**

Tabel 3. 1 Skala <i>Likert</i> .....	28
Tabel 3. 2 <i>Plot Aspek Usability</i> .....	34
Tabel 3. 3 Skor <i>SUS</i> .....	34
Tabel 4.1. Rekapitulasi Hasil Validasi Instrumen Kelayakan Materi .....	43
Tabel 4.2. Hasil Analisa Functionality .....	47
Tabel 4.3. Hasil Analisa Portability .....	51
Tabel 4.4. Hasil Analisa Efficiency .....	54
Tabel 4.5. Task-Task Usability Testing .....	58
Tabel 4.7. Rekap Nilai Usability .....	60

## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Pasca pandemi covid-19 yang panjang, semua instansi pendidikan menyatakan semua siswa dapat kembali belajar secara *offline* atau kembali ke sekolah. Banyak siswa atau mahasiswa yang kembali ke sekolah menengah atau perguruan tinggi untuk semester lain atau baru memulai studi secara *offline*, setelah sekian lama belajar di rumah. Siswa diharuskan untuk merencanakan hari, menjadwalkan sesi belajar, dan bahkan mengingat tugas-tugas penting dalam kehidupan pribadi.

Berdasarkan fenomena di atas, pembuatan aplikasi *ToDo list* berbasis *mobile* menggunakan *framework cross-platform* telah menjadi bagian penting karena dapat membantu meningkatkan efisiensi dalam mengelola dan mengembangkan kode untuk *platform* yang berbeda, sehingga menjadi urgensi dalam penelitian ini. Terdapat banyak pilihan dalam mengembangkan aplikasi *ToDo list* berbasis *mobile*, meskipun pada dasarnya sistem operasi *mobile* yang paling disukai adalah sistem operasi *Android* dan *iOS* (Packer, Wade, Diaz, Habjan, McIntyre, & Ancona, 2019). Alat pengembangan *platform* ini adalah *Android Studio* dan *Xcode* (Vilček & Jakopec 2017). Alat pengembangan *Xcode* dan *Android Studio* berbeda satu sama lain karena mereka mendukung bahasa perangkat lunak yang berbeda dan memiliki antarmuka yang berbeda. Bahasa *Java* atau *Kotlin* digunakan untuk pengembangan *Android*, sedangkan bahasa perangkat lunak *Objective-C* atau *Swift* digunakan untuk *iOS*. Perlu banyak waktu untuk menggunakan bahasa

perangkat lunak ini dan beradaptasi dengan lingkungan pengembangan ini. Mengembangkan dan menguji aplikasi yang sama dengan menggunakan alat pengembangan ini secara terpisah untuk setiap *platform* dapat mengakibatkan banyak kehilangan waktu dan biaya pemeliharaan yang tinggi (Kaur & Mishra 2019). Sehingga metode lintas *platform* yang muncul untuk mendukung semua *platform* dengan satu kode untuk aplikasi yang dibuat menggunakan alat yang berbeda menjadi sangat populer karena alasan ini (Yatsenko, Obodiak, Yatsenko., 2019).

Setiap *platform* memiliki penerapan *use case* yang berbeda. Masalah beberapa implementasi fitur juga menjadi masalah karena setiap *platform* memiliki perusahaan yang berbeda yang mengelolanya seperti *Android* yang dikelola oleh *Google* sedangkan *iOS* oleh *Apple*, dan tidak akan ada keseragaman dalam kode dan gaya di setiap *platform* mungkin berbeda satu sama lain (Gerasimov, Bilovol, Ivanova., 2015). Membangun sebuah aplikasi *Planner* seperti aplikasi *To-Do list* berbasis *mobile* membutuhkan strategi peluncuran yang baik dan membutuhkan ide yang cepat, validasi, dan riset pasar. Berdasarkan penelitian Keskin & Kılınç (2015), pengembang aplikasi cenderung menghadapi masalah seperti keterbatasan waktu dan anggaran untuk mengembangkan aplikasi mereka yang sangat krusial. Banyak waktu yang terbuang saat membuat aplikasi sehingga membutuhkan siklus pengembangan yang cepat dan harus memenuhi standar pengembangan aplikasi saat ini. Ada kebutuhan pengguna untuk memiliki beberapa *platform* (*Android*, *iOS*, dan *Website*) agar aplikasi *To Do list* berbasis *mobile* dapat bersaing di pasar. Selain itu, waktu yang dibutuhkan untuk membuat

setiap *platform* memakan banyak waktu dan jika setiap *platform* dikembangkan menggunakan bahasa asli mereka seperti *Android* menggunakan *java* dan *iOS* menggunakan *swift* akan rumit dalam pembaruan dan peningkatan lebih lanjut.

*Xamarin, Ionic, Adobe Phonegap, React Native, Flutter* adalah contoh metode lintas *platform* (Yatsenko dkk, 2019). Dalam penelitian ini ingin menggunakan *react native* dan *Flutter* sebagai fokus dalam penelitian, dimana dua metode tersebut didukung oleh *Google* dan *Facebook*. *Flutter* adalah *toolkit UI open-source* yang menggunakan bahasa *Dart* yang dapat membuat aplikasi seluler dengan basis kode tunggal dan mengkompilasi kode ke dalam *Android* dan *iOS*. *Flutter* diciptakan oleh *Google* tahun 2018 dan dapat membuat aplikasi yang mewarisi jenis tampilan, nuansa, dan kinerja yang sama seolah-olah akan dikembangkan sebagai aplikasi seluler asli. Sedangkan *React Native* adalah struktur berbasis *JavaScript* yang memungkinkan pengembangan aplikasi lintas *platform* untuk perangkat seluler dan dapat dijalankan mendekati *Native* seperti *Flutter*. Meskipun bersifat *open source* dan gratis, aplikasi dapat dikembangkan untuk *iOS* dan *Android* dengan *single coding* dengan *state management* dan dengan *plugin* (Eisenman, 2015).

Berdasarkan observasi awal peneliti terhadap mahasiswa jurusan teknik informatika dan komputer yaitu mahasiswa membuat sebuah aplikasi atau *website*, berupa tugas matakuliah ataupun tugas akhir. Pada proses pembuatan aplikasi dan *website*, tentunya menggunakan *framework* tertentu yang berfungsi untuk memudahkan proses pengembangan aplikasi atau *website* dengan menyediakan struktur dasar yang memudahkan proses pemrograman sistem,

dalam hal waktu dan keamanan. Perkembangan teknologi memudahkan para *programmer* dalam membuat sebuah aplikasi atau *website*, yang dulunya sebuah *platform* hanya dapat dijalankan pada satu manufaktur. Seiring perkembangan dunia teknologi, sebuah *platform* dapat dijalankan lebih dari satu manufaktur atau biasa kita kenal dengan *platform* lintas manufaktur dengan menggunakan *framework* tertentu. Namun, masih banyak mahasiswa yang kebingungan dalam penggunaan *framework* pada proses pembuatan aplikasi atau *website* dikarenakan alasan penggunaan *framework* tersebut mengikuti pada apa yang telah dipelajari selama proses perkuliahan, sehingga mahasiswa kurang mengetahui perbedaan dari setiap *framework* yang ada.

Oleh karena itu, peneliti ingin menyusun sebuah penelitian yang berjudul **“Analisis tingkat efektivitas antara *react native* dan *Flutter* dalam pembuatan aplikasi *To Do list* berbasis mobile”** yang bertujuan untuk menganalisis efektivitas dari penggunaan *framework* dalam hal ini *React Native* dan *Flutter* yang dimana sering digunakan dalam membuat sebuah *platform* lintas manufaktur.

## **B. Rumusan Masalah**

Berdasarkan penjelasan latar belakang, maka rumusan masalah ini yaitu bagaimana analisis tingkat efektivitas antara *react native* dan *Flutter* dalam pembuatan aplikasi *To Do List* berbasis mobile.

## **C. Tujuan Penelitian**

Berdasarkan rumusan masalah yang telah ditentukan, maka tujuan penelitian ini yaitu untuk mengetahui analisis tingkat efektivitas antara *react native* dan

*Flutter* dalam pembuatan aplikasi *To Do List* berbasis *mobile*.

#### **D. Manfaat Penelitian**

Adapun manfaat penelitian ini terbagi menjadi dua, yaitu:

1. Manfaat Teoritis

Hasil penelitian ini diharapkan dapat menjadi referensi atau bahan kajian dalam melakukan penelitian selanjutnya

2. Manfaat Praktis

- a. Bagi Jurusan Teknik Informatika dan Komputer UNM

Penelitian ini diharapkan dapat menjadi saran dalam materi perkuliahan khususnya pada penggunaan *framework*.

- b. Bagi Mahasiswa

Mahasiswa akan mendapatkan wawasan mendalam tentang perbandingan antara *React Native* dan *Flutter*, memungkinkan mereka membuat keputusan yang lebih informasional dalam memilih framework yang sesuai untuk proyek pengembangan aplikasi *mobile* mereka.

- c. Bagi Peneliti

Penelitian ini diharapkan dapat menjadi sumber pengetahuan dan wawasan untuk mengembangkan penelitian selanjutnya.

## **BAB II**

### **KAJIAN TEORI**

#### **A. Kajian Teori**

##### **1. Efektivitas**

Efektif mengandung arti sebagai pengaruh, efek, akibat, atau hasil. Sedangkan efektivitas diartikan sebagai keaktifan, daya guna dan adanya kesesuaian dalam suatu kegiatan antara seseorang dengan tujuan yang ingin dicapai (KBBI). Kata efektif berasal dari bahasa Inggris *effective* yang artinya berhasil atau segala sesuatu yang dilakukan dengan baik. Efektivitas memiliki arti berhasil ataupun tepat guna. Efektif merupakan kata dasar, sementara kata sifat dari efektif adalah efektivitas. Mardiasmo (2017) mengemukakan bahwa efektivitas adalah ukuran berhasil tidaknya pencapaian tujuan suatu organisasi mencapai tujuannya. Apabila suatu organisasi mencapai tujuannya maka organisasi tersebut telah berjalan dalam efektif. Indikator efektivitas menggambarkan jangkauan akibat dan dampak (*outcome*) dari keluaran (*output*) program dalam mencapai tujuan program.

Menurut Beni (2016), efektivitas adalah hubungan antara *output* dan tujuan atau dapat juga dikatakan merupakan ukuran seberapa jauh tingkat *output*, kebijakan dan prosedur dari organisasi. Efektivitas juga berhubungan dengan derajat keberhasilan suatu operasi pada sektor publik sehingga suatu kegiatan dikatakan efektif jika kegiatan tersebut mempunyai pengaruh besar terhadap kemampuan menyediakan pelayanan masyarakat yang merupakan sasaran yang telah ditentukan. Sedangkan Bungkaes dalam (Gibson, JL and Ivancevich, 2010)

Dalam (Musyarrayah, 2021) berpendapat bahwa efektivitas merupakan hubungan antara *output* maupun tujuan. Dalam artian efektivitas merupakan ukuran bagaimana serta seberapa jauh tingkat *output*, kebijakan, maupun prosedur dari organisasi mencapai tujuan yang ditetapkan sebelumnya.

Mengukur efektivitas tidak hanya dilakukan untuk mengukur efektivitas kinerja suatu organisasi saja namun pengukuran efektivitas dapat juga dilakukan untuk mengukur suatu efektivitas sistem. Efektivitas Sistem adalah melihat sejauh mana tingkat keberhasilan sistem aplikasi dalam kegiatan organisasi yang mencakup upaya peningkatan kinerja dengan menggunakan sistem aplikasi tersebut. Sehingga apakah penggunaan sistem aplikasi tersebut dapat efektif dalam menunjang kinerja penggunanya yang mana akan bermanfaat bagi penggunanya (Ventica, 2021).

Adapun indikator efektivitas yang digunakan oleh peneliti dengan melihat kualitas sebuah sistem tersebut yang merujuk Organisasi Standarisasi Internasional atau ISO yang bertujuan untuk mengetahui persepsi dari sebuah proyek perangkat lunak. ISO yang digunakan adalah ISO 9126 yang meliputi *functionality, efficiency, reliability, usability, maintainability, dan probability*.

a) *Functionally*

*Functionality* merupakan kemampuan perangkat lunak (*software*) untuk menyediakan fungsi sesuai kebutuhan pengguna pada kondisi tertentu (ISO 9126). Fungsionalitas merupakan atribut dari keberadaan fungsi dan sifat spesifik dari perangkat lunak dimana fungsi tersebut memenuhi atau mengandung kebutuhan. Fungsionalitas dinilai dengan mengevaluasi fitur dan

kemampuan dari program, sifat umum dari fungsi yang dikirimkan dan keamanan sistem secara keseluruhan. Aspek dari *functionality* dibagi menjadi 4 sub-karakteristik (ISO 9126) yaitu :

- 1) *Suitability*, kemampuan *software* untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna.
- 2) *Accuracy*, kemampuan *software* dalam memberikan hasil yang sesuai dengan kebutuhan sistem.
- 3) *Interoperability*, kemampuan *software* untuk berinteraksi dengan satu atau lebih sistem tertentu.
- 4) *Security*, kemampuan *software* untuk mencegah akses yang tidak diinginkan pada sistem.
- 5) *Standars* atau *compliance*, kemampuan *software* dalam mematuhi standar sesuai dengan peraturan yang berlaku.

b) *Efficiency*

Menurut ISO 9126 (2017), “*efficiency is concerned with the system resources used when providing the required functionality*”. *Efficiency* merupakan salah satu karakteristik perangkat lunak yang berhubungan dengan sumber daya yang digunakan ketika perangkat lunak menjalankan fungsi – fungsinya. Terdapat dua sub karakteristik pada faktor kualitas *efficiency* antara lain:

- 1) *Time Behavior*, kemampuan yang dimiliki perangkat lunak dalam memberikan respon waktu pengolahan yang sesuai saat melakukan fungsinya.

2) *Resource Utilization*, kemampuan perangkat lunak dalam menggunakan sumber daya yang dimiliki ketika menjalankan fungsi – fungsi yang telah ditentukan.

Analisis faktor kualitas *efficiency* dilakukan dengan menghitung rata – rata *response time* yang digunakan untuk melakukan beberapa tugas pada aplikasi yang telah dikembangkan. Lutfi Pratama dalam Nielsen (2020) membagi *response time* menjadi tiga interval. Dimana setiap tingkatan interval tersebut mempengaruhi sikap *user* terhadap suatu aplikasi berbasis *website*. Apabila *response time* lebih dari 10 detik, maka *user* akan meninggalkan situs dengan segera.

c) *Reliability*

Menurut ISO 9126 aspek *reliability* berkaitan dengan usaha untuk mempertahankan tingkat kinerja tertentu ketika digunakan dalam kondisi tertentu. *Reliability* (keandalan) adalah kemampuan yang diharapkan dari program untuk melakukan fungsinya yang membutuhkan ketelitian. Keandalan adalah sejauh mana produk beroperasi tanpa kegagalan dalam kondisi tertentu selama periode waktu tertentu. Pengembangan aplikasi terutama aplikasi yang berbasis *online*, diperlukan pengujian performa dengan cara menjalankan aplikasi pada waktu yang bersamaan dan dilakukan oleh banyak pengguna. Pengujian aspek *reliability* pada penelitian dilakukan dengan menggunakan suatu *automated software* untuk menguji performa aplikasi.

d) *Usability*

*Usability* merupakan kemampuan produk perangkat lunak untuk dipahami, dipelajari dan digunakan oleh pengguna tertentu untuk mencapai atau memperoleh kepuasan dalam hal penggunaannya. Pada model FRUPS yang dikembangkan oleh Robert Grady dan Hewlett-Packard Co, karakteristik *usability* digolongkan ke dalam kebutuhan non-fungsional. Kebutuhan non-fungsional perangkat lunak tidak berhubungan secara langsung dengan fitur – fitur perangkat lunak yang dikembangkan. Pada ISO 9126, karakteristik *usability* dibagi menjadi 4 sub-karakteristik yaitu:

- 1) *Understandability*, kemampuan *software* dalam kemudahan untuk dipahami.
  - 2) *Learnability*, kemampuan *software* dalam kemudahan untuk dipelajari.
  - 3) *Operability*, kemampuan *software* dalam memberikan kemudahan dalam pengoperasiannya.
  - 4) *Attractiveness*, kemampuan *software* dalam menarik pengguna.
- e) *Maintainability*

Perangkat lunak yang telah selesai dikembangkan kemudian diimplementasikan dan disebarluaskan, namun pada kenyataannya masih terdapat hal – hal yang perlu untuk diperbaiki. Upaya memperbaiki kesalahan – kesalahan yang terdapat pada *software* ini disebut dengan *maintainability* atau dapat dikatakan sebagai proses pemeliharaan *software*. Menurut ISO 9126, *maintainability* merupakan kemampuan *software* untuk dapat dengan mudah dipahami, dikembangkan dan dimodifikasi. Proses modifikasi meliputi koreksi dan perbaikan. Pemeliharaan *software* dapat digolongkan menjadi tiga jenis,

yaitu pemeliharaan terhadap *error coding*, penyesuaian perangkat lunak pada lingkungan dan modifikasi dari segi fungsionalitas. Karakteristik *maintainability* memiliki empat sub karakteristik yaitu (ISO 9126):

- 1) *Analysability*, kemampuan *software* dalam mendiagnosis penyebab kegagalan – kegagalan pada sistem.
- 2) *Changeability*, kemampuan *software* untuk dapat dimodifikasi.
- 3) *Stability*, kemampuan *software* untuk dapat meminimalkan dampak yang tak terduga dari proses modifikasi.
- 4) *Testability*, kemampuan *software* untuk dimodifikasi dan divalidasi dengan perangkat lunak lain.

f) *Portability*

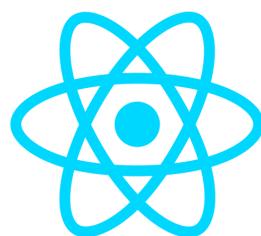
*Portability* berhubungan dengan kemampuan perangkat lunak yang dipindahkan ke lingkungan berbeda. Pada aspek *portability* menjawab pertanyaan bagaimana perangkat lunak dapat dipindahkan dari *hardware* maupun *software* tertentu ke yang lainnya. Pada ISO 9126 karakteristik *portability* memiliki empat sub-karakteristik yaitu *adaptability*, *installability*, *co-existence*, dan *replaceability*.

- 1) *Adaptability*, kemampuan perangkat lunak diadaptasi pada lingkungan yang berbeda – beda.
- 2) *Installability*, berkaitan dengan kemampuan perangkat lunak untuk dapat di-install pada lingkungan yang berbeda.
- 3) *Co-existence*, kemampuan perangkat lunak dalam berbagi sumber daya dengan perangkat lunak lain.

- 4) *Replaceability*, merupakan kemampuan perangkat lunak untuk dapat digunakan pada lingkungan tertentu.

## 2. *React Native*

*React Native* adalah kerangka kerja *JavaScript* yang memungkinkan Anda membuat aplikasi seluler *real-time* yang di render secara *native* untuk *iOS* dan *Android*. *React Native* dibangun di atas *React*, toolkit *JavaScript* Facebook untuk membuat antarmuka pengguna, meskipun dirancang untuk *platform* seluler daripada *Website*. Dengan kata lain, pengembang *Website* kini dapat membuat aplikasi seluler yang terlihat dan terasa sepenuhnya "asli", semuanya sambil menggunakan pustaka *JavaScript* yang sudah dikenal. Solusi untuk ini mungkin adalah kerangka kerja yang baru disajikan *React Native*, dikembangkan oleh *Facebook*, pertama kali dirilis pada tahun 2015, dan sekarang merupakan proyek *open source* di *GitHub*. Dengan *React Native* kita dapat menggunakan yang terbaik dari *React* (seperti *fast re-builds* dan DOM deklaratif) dan dapat menjalankan *instance JavaScript Core* di dalam aplikasi dan dengan itu membuat komponen khusus *platform*.



Gambar 2. 1 *Framework React Native*

*React Native* didefinisikan sebagai alat/kerangka skrip asli untuk pengembangan aplikasi seluler lintas *platform* secara umum yang diperkenalkan

oleh *Facebook* dan dirilis pada tahun 2015. Aplikasi *React Native* dirancang dan dikembangkan menggunakan *JavaScript*. *ECMAScript* adalah bahasa skrip berdasarkan *JavaScript* dan memengaruhi tampilan atau perilaku halaman *Website*. *React Native* mendukung sistem operasi Android dan iOS.

Arsitektur *React Native* pada dasarnya terdiri dari tiga bagian: modul asli, mesin virtual *JavaScript* dan jembatan *React Native*. Kode asli untuk *iOS* dan *Android* sebagian besar ditulis menggunakan *Objective-C* dan *Java*. Namun, di *React Native*, pengembang tidak perlu menulis kode asli untuk *iOS* atau *Android*. Juga, *React Native* menggunakan *JavaScript Core* untuk menjalankan kode. *JavaScript Core* adalah mesin *JavaScript open-source* yang dibuat untuk *WebKit*. *React Native bridge* adalah jembatan komunikasi antara *thread native* dan *JavaScript*. *React Native bridge* membantu merutekan panggilan dari modul asli ke API asli.

*React Native* dibangun di atas prinsip-prinsip yang digunakan dan diimplementasikan di *React*, yang dikenal sebagai *React*. *React* adalah *framework* berbasis komponen untuk membuat antarmuka untuk aplikasi *Website* yang elemen dasarnya adalah sebuah komponen. Komponen *React* mengimplementasikan metode render yang mengambil *input* dan mengembalikan *output* yang sesuai. Sesuai dengan kebutuhan proyek, berbagai komponen disusun untuk membangun *UI* yang lebih kompleks dan transfer data antar komponen dilakukan menggunakan alat peraga, dari induk ke anak.

### 3. *Flutter*

#### a) Definisi

*Flutter* adalah kerangka kerja lintas *platform* untuk mengembangkan aplikasi yang dapat berjalan di Android, iOS, dan sistem operasi generasi berikutnya *Google Fuschia*. *Flutter* dibuat oleh tim *browser Google Chrome* dan dirilis secara publik pada tahun 2016. Alasan pembuatan *Flutter* adalah untuk mengevaluasi apakah rendering bisa lebih efisien jika model tata letak tradisional diabaikan. Ini menghasilkan cara berpikir yang berbeda tentang tata letak. Alih-alih membiarkan tata letak ditentukan oleh seperangkat aturan yang besar, setiap *widget* dalam tata letak akan menentukan seperangkat aturannya sendiri.



Gambar 2. 2 *Framework Flutter*

*Flutter* sebagai *framework* sangat menjanjikan dan saat ini memiliki komunitas dev yang besar. Bahkan saat ini kita dapat menemukan aplikasi kompleks di pasar yang didasarkan pada *Flutter*, seperti *Alibaba*, *Google Ads*, *Reflectly*, *Birch Finance*, *Hamilton Musical*, *Hookle* (Skuza, 2019). Menurut pendapat Penulis, teknologi ini adalah pilihan yang baik untuk aplikasi berukuran kecil dan menengah atau ketika konten dan fitur dasar memerlukan iterasi yang konstan.

b) Dart

Aplikasi *Flutter* menggunakan bahasa pemrograman *Dart*. *Dart* dikembangkan dan dikelola oleh *Google*. Ini dikembangkan sebagai penerus *JavaScript* dan menggabungkan banyak fitur dalam standar *JavaScript* ES7. Seperti disebutkan sebelumnya, *Dart* dikompilasi sebelumnya. Ini memungkinkan eksekusi lebih cepat dan menghindari jembatan *JavaScript*. Kode masih membutuhkan antarmuka untuk berkomunikasi dengan kode *platform*. Namun ini jauh lebih cepat daripada jembatan *JavaScript*. *Flutter* juga tidak menggunakan *widget platform OEM*, mengurangi berapa kali komunikasi dengan *platform* terjadi.



Gambar 2. 3 *Dart*

*Dart* juga dapat dikompilasi menggunakan *compiler just in time*. Hal ini dilakukan selama pengembangan dan memungkinkan siklus pengembangan yang cepat. *Flutter* memanfaatkan *compiler just in time* menggunakan fitur yang disebut *hot reload*. Muat ulang panas memungkinkan perubahan kode dimuat ulang dan disuntikkan pada perangkat dengan sangat cepat. Ini memungkinkan pengembangan aplikasi *Flutter* yang sangat efisien. Kelemahannya adalah aplikasi akan kurang efisien selama pengembangan.

c) Widget

*Flutter* tidak menggunakan *widget OEM* asli, melainkan menyediakan widgetnya sendiri. *Widget* tidak dirender oleh sistem. Sebagai gantinya, aplikasi itu sendiri menyertakan perender. *Flutter* hanya membutuhkan kanvas untuk merender *widget* di layar perangkat. Itu juga membutuhkan akses ke layanan (kamera, lokasi) dan acara (pengatur waktu, acara sentuh).

Ini memungkinkan *Flutter* untuk memutuskan kapan dan bagaimana *widget* akan dirender. Ini juga memungkinkan *widget* menjadi lebih dapat disesuaikan dan diperluas. Kelemahan dari pendekatan ini adalah ukuran aplikasi meningkat, karena perlu menghosting *widget* dan perender di dalam aplikasi.

#### 4. Aplikasi *ToDo List*

*ToDo List App* adalah sejenis aplikasi yang umumnya digunakan untuk mencatat tugas sehari-hari kita atau membuat daftar semua yang harus kita lakukan, dengan tugas yang paling penting di bagian atas daftar, dan tugas yang paling tidak penting di bagian bawah. *ToDo List* sangat membantu dalam merencanakan jadwal harian kita. Kami dapat menambahkan lebih banyak tugas kapan saja dan menghapus tugas yang sudah selesai. Dalam daftar *ToDo* versi ini, pengguna akan mendapatkan empat opsi:

- a. Buat (tambah) tugas baru atau tambahkan *ToDo* baru di Aplikasi Daftar *ToDo*.
- b. Lihat semua tugas atau Lihat semua *ToDos* yang ditambahkan ke aplikasi.
- c. Hapus semua *ToDo* dari daftar *ToDos*
- d. Keluar dari aplikasi

## 5. Aplikasi *Mobile*

Pada dasarnya ada tiga jenis arsitektur yang digunakan dalam pengembangan aplikasi *mobile*: aplikasi *Website*, *native*, dan *hybrid*. Dan dalam penelitian ini penulis menggunakan arsitektur *native* sebagai acuan untuk pembuatan aplikasi yang akan digunakan sebagai media perbandingan.

## 6. Aplikasi *Native*

Pengembangan aplikasi seluler asli adalah salah satu cara tradisional untuk mengembangkan aplikasi, di mana aplikasi biasanya dirancang dan dikembangkan untuk *platform* atau perangkat tertentu. Karena aplikasi asli berorientasi *platform*, pengembangan aplikasi menggunakan bahasa pemrograman asli *platform* dan arsitektur perangkat lunak, yang kemudian dikompilasi menjadi aplikasi asli yang sebenarnya. Untuk *Android*, bahasa pemrograman *Java* umumnya digunakan untuk mengembangkan aplikasi, sedangkan aplikasi di *iOS* dikodekan menggunakan *Objective-C/Swift*. Demikian pula, *Model-View Controller (MVC)* adalah arsitektur yang direkomendasikan untuk *iOS*, dan untuk *Android*, arsitektur umumnya adalah *Model-View-View Model (MVVM)*. Selain itu, aplikasi dapat diunduh dan diinstal dengan bantuan toko aplikasi, per se *Google Play store* atau *Apple App Store*. Aplikasi harus langsung diinstal pada perangkat yang membantu aplikasi untuk memanfaatkan perangkat keras & perangkat lunak yang khusus untuk perangkat, seperti *Global Positioning System (GPS)*, kamera, dll.

Salah satu kelebihan dari aplikasi *mobile native* dapat berupa pengalaman pengguna (*UX*). Pengembang menulis kode asli untuk pengembangan aplikasi tersebut, jadi, ini memberi mereka kesempatan untuk menggunakan *API* asli

(*Application Programming Interfaces*) dan perpustakaan yang didukung oleh *platform* tertentu. Pengembang juga dapat membuat antarmuka pengguna (*UI*) dengan bantuan komponen *UI* asli. Selain itu, karena aplikasi diinstal pada perangkat, aplikasi biasanya tidak perlu terhubung ke Internet untuk dijalankan, namun mungkin ada beberapa tugas dalam aplikasi yang mungkin memerlukan fasilitas Internet untuk berfungsi.

Sementara aplikasi asli menyediakan *UX* yang lebih baik, itu masih membutuhkan basis kode terpisah untuk *platform* yang berbeda. Ketika aplikasi yang sama perlu dikembangkan di berbagai *platform*, pengembang perlu memiliki pengetahuan yang diperlukan tentang metode pengembangan, bahasa pemrograman, panduan desain, dan teknologi khusus untuk *platform* tersebut. Karena itu, pengembangan dan pemeliharaan aplikasi asli membutuhkan lebih banyak orang dengan keahlian yang diperlukan. Hal ini meningkatkan biaya pembangunan.

## 7. Aplikasi *Website*

Aplikasi *Website* dikembangkan dengan bantuan teknologi seperti *CSS*, *HTML5* dan *JavaScript*. Oleh karena itu, aplikasi ini dapat diproses di *browser Website*, yang membuatnya mudah diakses oleh sebagian besar perangkat. Tampilan dan nuansa aplikasi ini konsisten di beberapa perangkat. Aplikasi dapat tampil di *platform* yang berbeda dengan beberapa perubahan kecil karena satu basis kode digunakan untuk beberapa *platform* dan tampilan serta nuansa aplikasi serupa di *platform* tersebut. Fitur menggunakan basis kode tunggal adalah salah satu keunggulan aplikasi *Website*.

Kelemahan utama dari aplikasi *Website* adalah grafik yang *intens* dan masalah yang terjadi ketika aplikasi perlu disesuaikan dengan grafik tersebut, dan juga aplikasi ini tidak dapat digunakan tanpa koneksi internet. Selain itu, fitur-fitur seperti *GPS*, kamera, notifikasi *push*, *database*, dll. tidak dapat diakses saat menggunakan aplikasi *Website*. Namun, dengan standar *API HTML5*, beberapa fitur ini saat ini dapat diakses. Untuk membuat fitur ini tersedia, jenis aplikasi *Website* baru dirancang, yang disebut aplikasi *Website progresif (PWA)*.

*PWA* menggunakan satu aplikasi di berbagai *platform* seperti *Android*, *iOS*, dan *windows*. Hal ini dapat diandalkan dan efisien. Ini memungkinkan pemuatan instan dan pemberitahuan *push* bahkan ketika tidak ada koneksi internet. Ini berisi aspek positif dari aplikasi asli; itu akan dapat mengakses sumber daya di ponsel dan tidak memerlukan fungsi toko aplikasi. Namun, teknologi ini baru dan belum matang.

## 8. *Hybrid* Aplikasi

Aplikasi *hybrid* terdiri dari fitur aplikasi asli dan aplikasi *Website*. Mirip dengan aplikasi asli, aplikasi *hybrid* dapat diunduh dan diinstal di perangkat secara langsung. Ini memberikan akses aplikasi ke kemampuan perangkat keras perangkat. Pada saat yang sama, aplikasi *hybrid* dikembangkan menggunakan teknologi aplikasi *Website* seperti *HTML5*, *CSS* dan *JavaScript* .

## B. Penelitian Relevan

1. Penelitian yang dilakukan oleh Shumaya Resty Ramadhani, Juni Nurma Sari, Indah Lestari, dan Susiyanti (2021) yang berjudul “**Pengembangan Aplikasi Monitoring Penyebaran Virus Covid-19 Berbasis Mobile Area**

**Pekanbaru dengan *Prototyping***. Aplikasi ini dikembangkan dengan pendekatan *hybrid* menggunakan *Flutter* dan pendekatan model proses *prototyping*. Metode ini digunakan untuk menyesuaikan karakteristik proyek sesuai dengan kebutuhan *stakeholder* (dalam hal ini pemerintah) yang tidak spesifik dan dibutuhkan secara cepat.

2. Penelitian yang dilakukan oleh Adlan Husein Malahella, Issa Arwani, Tibyani (2020) yang berjudul “**Pemanfaatan *Framework React Native* dalam Pengembangan Aplikasi Pemesanan Minuman Kopi pada Kedai *Bycoffee***”. Oleh karena itu pada tugas akhir ini dibangun sebuah aplikasi pemesanan minuman kopi di kedai kopi *ByCoffee* dengan menggunakan *framework React Native* yang diharapkan dapat mempermudah dan mempercepat proses pelayanan di kedai *ByCoffee*.
3. Penelitian yang dilakukan oleh Zakki Abdurrahman Haris (2021) yang berjudul “**Komparasi *Flutter* Dan *React Native* Dalam Pengembangan Perangkat Bergerak**”. Hasil yang diidentifikasi dalam pengujian dengan beberapa domain menunjukkan bahwa *React Native* baik untuk kustomisasi (setup dan konfigurasi), *modifiability* (level *coupling* data), dan performa (mode *debug*). Sedangkan *Flutter* baik untuk *modifiability* (jumlah fungsi, level *cohesion procedural*), *performance* (mode *profile*), dan *testability*.
4. Penelitian yang dilakukan oleh Fauzi Sholichin, dkk (2019) yang berjudul “**Review of IOS Architectural Pattern for Testability, Modifiability, and Performance Quality**”. Dari hasil pengujian *testability* dengan membandingkan banyak kode program dan waktu eksekusi dalam

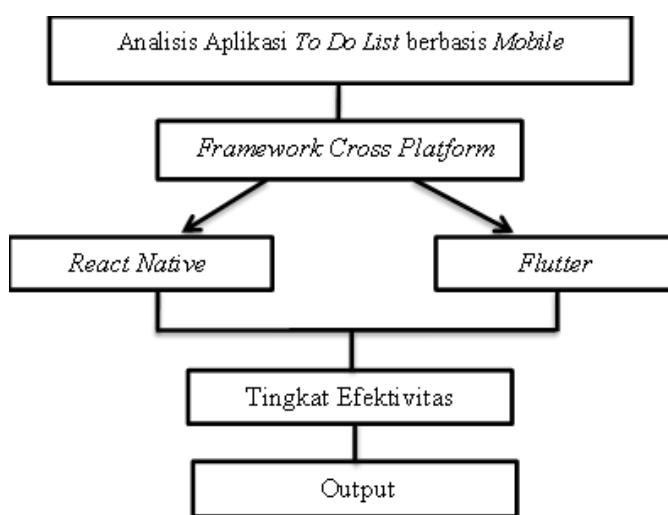
menjalankan aplikasi, *MVVM* lebih unggul dibandingkan arsitektur lainnya.

Pengujian modifikasi dilakukan dengan menganalisis dependensi suatu komponen dan menghitung jumlah fungsi, dengan hasil *VIPER* lebih unggul dalam pengujian dependensi komponen dan *MVVM* lebih unggul dengan sedikit jumlah fungsi.

Berdasarkan penelitian relevan diatas penulis dapat mengambil kesimpulan bahwa masih minimnya artikel maupun penelitian yang memberikan informasi terkait *Framework React Native* dan *Flutter* sehingga minim pengetahuan yang dapat membantu *Develover* dalam pemilihan *Framework* yang tepat untuk pembuatan ataupun pengembangan aplikasi kedepannya.

### C. Kerangka Pikir

Pada kerangka pikir ini akan dijelaskan Langkah-langkah yang dilakukan dalam penelitian yang kemudian dijadikan acuan dalam menyelesaikan masalah dalam penelitian dan mencari hasil penelitian. Alur kerangka pikir dalam penelitian ini memiliki beberapa Langkah, antara lain:



Gambar 2. 4 Kerangka Pikir

## **BAB III**

### **METODE PENELITIAN**

#### **A. Jenis Penelitian**

Jenis penelitian yang digunakan yakni studi kasus *framework Flutter* dan *React Native* pada pembuatan aplikasi *to do list* berbasis *mobile*. Studi kasus merupakan penelitian yang memusatkan perhatian pada suatu kasus tertentu dengan menggunakan individu atau kelompok sebagai bahan studinya. Penggunaan penelitian studi kasus ini biasanya difokuskan untuk menggali dan mengumpulkan data yang lebih dalam terhadap obyek yang diteliti untuk dapat menjawab permasalahan yang sedang terjadi. Sehingga bisa dikatakan bahwa penelitian bersifat deskriptif dan eksploratif. Penelitian *case study* (studi kasus) berbeda dengan penelitian survei. Pada penelitian survei jumlah sampelnya cukup luas sedangkan pada *case study* jumlah sampel yang diambil sangat sedikit atau hanya beberapa orang saja. (Hasibuan, 2007).

Pada penelitian ini, studi kasus digunakan untuk membuktikan kualitas perangkat lunak menggunakan kedua *framework*. Penggunaan studi kasus ini difokuskan untuk pengujian pada *performance* dan implementasi *testability*. Dalam pengujian *performance* dibutuhkan fitur yang dapat menekan atau memberatkan memori dan CPU seperti pengguliran daftar data, animasi, dan perpindahan data. Dalam implementasi *testability* studi kasus dibutuhkan dalam melakukan pengujian unit *testing* dan *integration testing* seperti pemanggilan *API* dan terdapat beberapa halaman yang terintegrasi.

## **B. Waktu dan Lokasi Penelitian**

### **1. Waktu Penelitian**

Waktu yang digunakan untuk penelitian ini dilaksanakan selama 6 bulan yang terhitung sejak Juli sampai Januari 2024.

### **2. Lokasi Penelitian**

Lokasi penelitian ini dipilih berdasarkan kriteria tertentu yang merupakan tempat dimana peneliti melakukan penelitian terutama dalam menangkap fenomena atau peristiwa yang sebenarnya terjadi dari objek yang diteliti dalam rangka mendapatkan data-data peneliti yang akurat. Adapun yang menjadi tempat atau lokasi penelitian ini yaitu Organisasi KeDai computerworks.

## **C. Populasi dan Sampel**

Populasi merupakan wilayah generalisasi yang terdiri atas objek atau subjek yang mempunyai kualitas dan karakteristik tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya (Arikunto, 2010). Sampel merupakan bagian dari jumlah karakteristik yang dimiliki oleh populasi. Untuk itu sampel yang diambil dari populasi harus betul-betul mewakili (Sugiyono, 2012).

Teknik sampling yang digunakan dalam penelitian ini adalah non-probability sampling. Teknik non-probability sampling yaitu teknik pengambilan sampel yang tidak memberi peluang/ kesempatan sama bagi setiap unsur atau anggota populasi untuk dipilih menjadi sampel (Sugiyono, 2012). Jenis non-probability sampling yang digunakan adalah purposive sampling. Purposive sampling yaitu teknik sampling yang digunakan peneliti jika peneliti mempunyai

pertimbangan-pertimbangan tertentu di dalam pengambilan samplenya atau penentuan sampel untuk tujuan tertentu.

Adapun teknik sampling untuk non-probabilitas yang penulis gunakan yaitu sampel sengaja (*purposive sampling*) yaitu penulis hanya meneliti terhadap informan yang betul-betul berkompeten atau ahli dalam bidangnya sehingga peneliti mendapatkan informasi yang dibutuhkan (Indrawan dan Yaniawati, 2014). Dalam penelitian ini kemudian penulis menggunakan metode khususnya judgement sampling yang merupakan turunan dari *purposive sampling*, karena penulis membutuhkan jawaban – jawaban yang paham tentang penelitian yang penulis lakukan. Dalam hal ini merupakan posisi yang paling baik untuk menjawab pertanyaan – pertanyaan atau memberikan informasi terkait. Persyaratan responden untuk penelitian "Analisis Tingkat Efektivitas Antara *React Native* dan *Flutter* dalam Pembuatan Aplikasi *To-Do List* Berbasis *Mobile*" dapat mencakup beberapa kriteria yang akan membantu memastikan bahwa responden memiliki pemahaman yang memadai tentang pengembangan aplikasi *mobile* dan dapat memberikan wawasan yang berharga dalam perbandingan kedua kerangka kerja. Berikut adalah beberapa persyaratan yang mungkin diterapkan pada responden:

1. Pengalaman sebagai Pengembang Aplikasi *Mobile*:

Responden harus memiliki pengalaman dalam pengembangan aplikasi *mobile*, baik dengan menggunakan *React Native* maupun *Flutter*. Pengalaman minimal yang diharapkan adalah setidaknya 1 tahun sebagai pengembang aplikasi *mobile*.

2. Familiaritas dengan React Native dan Flutter:

Responden harus memiliki pemahaman yang baik tentang baik *React Native* maupun *Flutter*, termasuk pengetahuan tentang bahasa pemrograman yang digunakan (*JavaScript* untuk *React Native* dan *Dart* untuk *Flutter*).

3. Pengalaman Membangun Aplikasi To-Do List:

Idealnya, responden harus memiliki pengalaman dalam pengembangan atau kontribusi pada aplikasi *To-Do List* berbasis *mobile*, baik dengan menggunakan *React Native* atau *Flutter*.

4. Ketersediaan untuk Berpartisipasi:

Responden harus bersedia dan dapat berpartisipasi dalam pengisian kuesioner atau wawancara terkait penelitian ini.

5. Kemampuan untuk Memberikan Masukan Mendalam:

Responden harus dapat memberikan masukan dan pemahaman mendalam tentang penggunaan *React Native* dan *Flutter* dalam pengembangan aplikasi *To-Do List*, termasuk masalah, kelebihan, dan kelemahan yang mereka temui.

6. Bekerja atau Pernah Bekerja pada Proyek Serupa:

Diutamakan jika responden telah bekerja pada proyek aplikasi *To-Do List* berbasis *mobile* dengan menggunakan *React Native* atau *Flutter*.

7. Kemampuan Menyampaikan Pendapat dengan Jujur:

Responden harus bersedia memberikan pendapat dengan jujur dan tanpa bias terhadap salah satu kerangka kerja.

8. Kemampuan Berkommunikasi dengan Baik:

Kemampuan berkomunikasi yang baik penting untuk menjawab pertanyaan dengan jelas dan memberikan pandangan yang tepat.

9. Ketersediaan waktu untuk Berpartisipasi:

Responden harus memiliki ketersediaan waktu yang cukup untuk berpartisipasi dalam penelitian ini, baik dalam pengisian kuesioner atau berpartisipasi dalam wawancara jika diperlukan.

10. Tidak Terikat Kontrak Keamanan atau Rahasia:

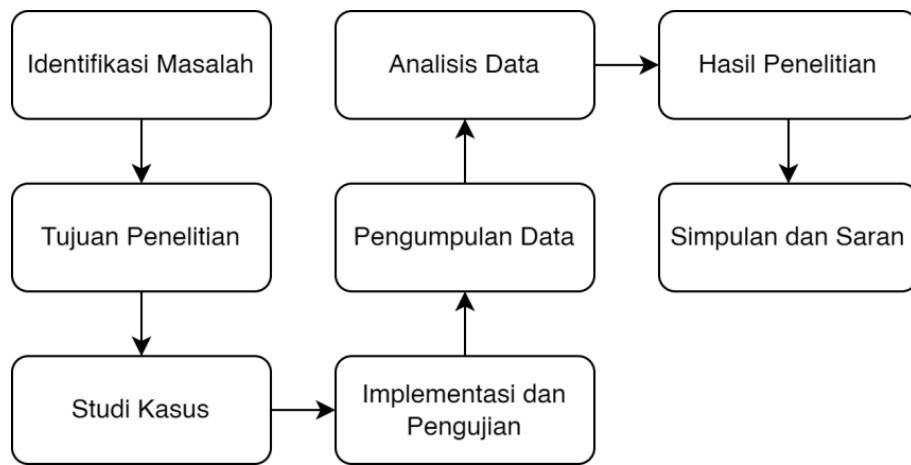
Responden harus memastikan bahwa mereka tidak terikat oleh kontrak keamanan atau perjanjian kerahasiaan yang melarang mereka berbicara tentang pengalaman atau pemahaman mereka terkait dengan pengembangan aplikasi menggunakan *React Native* atau *Flutter*.

Memastikan bahwa responden memenuhi persyaratan ini akan membantu dalam memastikan bahwa data yang diperoleh dalam penelitian ini akan bermanfaat dan relevan untuk tujuan analisis tingkat efektivitas antara *React Native* dan *Flutter* dalam pembuatan aplikasi *To-Do List* berbasis *mobile*.

Oleh karena itu maka di tentukan sampel diambil dari 10 orang yang berafiliasi dalam organisasi KeDai *Computerworks* karena telah memenuhi kriteria sampel yang digunakan dalam penelitian ini.

#### **D. Desain Penelitian**

Pada penelitian ini membandingkan *framework Flutter* dan *react native* dengan empat implementasi dan pengujian komparasi domain yang terdiri dari *customizability*, *modifiability*, *testability*, dan *performance*. Adapun alur metode penelitian yaitu:



Gambar 3. 1 Desain Penelitian

### E. Prosedur Penelitian

Prosedur penelitian yang digunakan terbagi kedalam beberapa tahap yaitu:

1. Tahap Persiapan
  - a. Merumuskan dan mengidentifikasi masalah, meninjau kepustakaan yang relevan, mengidentifikasi kerangka teoritis.
  - b. Mengidentifikasi populasi dan sampel yang diteliti, mengkhususkan metode untuk mengukur variabel penelitian, merancang rencana sampling, mengakhiri dan meninjau rencana penelitian, melaksanakan penelitian dan melakukan revisi.
  - c. Membuat instrumen dan pengumpulan data penelitian.
  - d. Melakukan uji coba instrumen penelitian meliputi uji validitas.
2. Tahap kegiatan inti
  - a. Membagikan angket kepada 10 responden ahli yang menjadi objek dalam penelitian ini.

- b. Menganalisis data yang telah diperoleh menggunakan analisis statistik deskriptif.
3. Tahap akhir

Pada tahap akhir, agar hasil penelitian dapat dibaca, dimengerti dan diketahui oleh pembaca maka hasil penelitian tersebut disusun dalam bentuk kesimpulan dari hasil penelitian.

## F. Teknik Pengumpulan Data

Untuk mendapatkan data-data dan informasi yang akurat, maka teknik pengumpulan data yang digunakan dalam penelitian ini yaitu:

### 1. Kuesioner

Kuesioner pada penelitian ini diperuntukkan untuk responden yang telah memiliki kriteria tertentu. Pada kuesioner ini, peneliti menggunakan skala likert dengan empat pilihan jawaban yaitu (1) sangat tidak setuju skor 1, (2) tidak setuju skor 2, (3) setuju skor 3, (4) sangat setuju skor 4.

**Tabel 3.1 Skala Likert**

No	Pilihan Jawaban	Skor
1	Sangat Tidak Setuju	1
2	Tidak Setuju	2
3	Setuju	3
4	Sangat Setuju	4

## 2. Dokumentasi

Dokumentasi dalam penelitian ini adalah data jumlah responden yang menggunakan *framework* antara *react native* dan *Flutter* dalam pembuatan aplikasi *To Do List* berbasis *mobile*.

### G. Instrument *Plot Usability*

*Plot Usability* adalah proses memvisualisasikan dan menganalisis data terkait *Usability* (kegunaan atau kemudahan penggunaan) suatu produk atau sistem. Dalam konteks penilaian *usability*, *Plot Usability* dapat mencakup berbagai aspek yang diukur selama pengujian atau penilaian *Usability*.

Pengguna mengisi kuisioner yang sudah dibagikan berdasarkan pengalamannya (apa yang dilihat dan dirasakan) pada saat melakukan task atau tugas-tugas yang di kerjakan. Tiap-tiap pertanyaan dari kuesioner tersebut bertujuan untuk menunjukkan tingkat *usability* menurut penerimaan user, yang akan dinilai dalam 5 aspek (Dedi Rianto Rahadi, 2014).

Menurut Dedi Rianto Rahadi, aspek-aspek dalam usability testing ini mencakup lima hal, yaitu:

- a. *Learnability*, menjelaskan tingkat kemudahan pengguna atau user untuk menyelesaikan task-task dasar ketika pertama kali mereka melihat atau berhadapan dengan sistem yang ada.
- b. *Efficiency*, menjelaskan seberapa cepat pengguna dapat menyelesaikan tugas-tugas yang ada saat mereka pertama kali mempelajari sistem tersebut.

- c. *Memorability*, menjelaskan tentang tingkat kemudahan pengguna atau user dalam menggunakan sistem dengan baik, setelah beberapa lama tidak menggunakannya.
  - d. *Errors*, menjelaskan kemungkinan terjadinya error atau kesalahan yang dilakukan oleh pengguna dan seberapa mudah mereka dapat mengatasinya.
  - e. *Satisfaction*, menjelaskan tentang tingkat kepuasan pengguna dalam menggunakan sistem yang telah dibuat.

Hasil plot kelima aspek *usability* di atas terhadap 11 pertanyaan kuesioner yang dapat dilihat pada Tabel 3.1

Tabel 3.2. Plot Aspek Usability

No	Pertanyaan	Flutter				React Native			
		LR	EF	MR	SF	LR	EF	MR	SF
	aplikasi mudah dikenali								
6	Menurut saya, aplikasi ini mudah dibaca								
7	Saya merasa aplikasi mudah diinstal								
8	Menurut saya, simbol-simbol pada aplikasi mudah dipahami								
<b>Aspek Interaksi (INTERACTION) :</b>									
9	Saya merasa menu dan tampilan aplikasi mudah diingat								
10	Menurut saya, aplikasi ini mudah dipelajari								
11	Saya menganggap aplikasi ini praktis digunakan								

Keterangan :

**LR** = Learnability

**EF** = Efficiency

**MR** = Memorability

**SF** = Satisfaction

## H. Teknik Analisis Data

Adapun teknik analisis data yang digunakan pada penelitian ini yaitu:

### 1. Analisis *Functionality*

Analisis aspek kualitas *functionality* dilakukan dengan observasi terhadap fungsi – fungsi perangkat lunak menggunakan instrumen *test case*. Setiap *test*

*case* menggambarkan apakah suatu fungsi dapat berjalan sebagaimana mestinya atau tidak. Dari fungsi – fungsi yang telah diujikan kemudian dilakukan perhitungan dengan menggunakan *metric* ISO 9126 mengenai sub karakteristik *suitability*. Apabila nilai yang dihasilkan mendekati angka 1 maka dapat disimpulkan bahwa aplikasi yang dikembangkan memenuhi dari aspek *functionality*.

## **2. Analisis *Portability***

Untuk menguji aspek portabilitas, dilakukan pengamatan terhadap aplikasi. Ini dilakukan dengan menjalankan aplikasi pada berbagai jenis perangkat android yang menggunakan versi os dan resolusi layar yang variatif. Pada pengujian aspek portability, yang digunakan adalah installability yaitu mengamati apakah aplikasi dapat di-install dan berjalan pada perangkat yang berbeda, kemudian melihat apakah aplikasi dapat berjalan dengan semestinya tanpa mengalami error. Selanjutnya, dilakukan perhitungan dan pencatatan skor persentase hasil pengujian untuk kemudian dilakukan penarikan kesimpulan pada aplikasi yang dibuat. (Dwi Gusna. 2023).

## **3. Analisis *Efficiency***

Menurut ISO 9126 (2017), “*efficiency* is concerned with the system resources used when providing the required functionality”. *Efficiency* merupakan salah satu karakteristik perangkat lunak yang berhubungan dengan sumber daya yang digunakan ketika perangkat lunak menjalankan fungsi – fungsinya. Terdapat dua sub karakteristik pada faktor kualitas *efficiency* antara lain :

1) *Time Behavior*, kemampuan yang dimiliki perangkat lunak dalam memberikan respon waktu pengolahan yang sesuai saat melakukan fungsinya.

2) *Resource Utilization*, kemampuan perangkat lunak dalam menggunakan sumber daya yang dimiliki ketika menjalankan fungsi – fungsi yang telah ditentukan.

Dalam melakukan analisis sub karakteristik *time behavior* dapat menggunakan *metric response time*. *Response time* merupakan ukuran perkiraan waktu untuk melakukan tugas yang diberikan pengguna ke sistem. Secara umum, *response time* diukur mulai dari pengguna menjalankan suatu kontrol tertentu misalnya melakukan klik sampai aplikasi merespon dengan menampilkan output atau action (Neutrina Nilamsar, 2014).

Analisis faktor kualitas *efficiency* dilakukan dengan menghitung rata – rata *response time* yang digunakan untuk melakukan beberapa tugas pada aplikasi yang telah dikembangkan. (Nielsen,2014), membagi *response time* menjadi tiga interval. Dimana setiap tingkatan interval tersebut mempengaruhi sikap user terhadap suatu aplikasi.

#### **4. Analisis *Usability***

Analisis kualitas aspek *usability* dilakukan dengan menggunakan metode kuesioner. Kuesioner dibagikan kepada 10 responden ahli sebagai aspek *usability*. Kuesioner yang digunakan mengacu pada kuesioner *System Usability Scale (SUS)* oleh John Brooke. Pada kuesioner tersebut jawaban setiap item pertanyaan menggunakan skala *likert* yang mempunyai gradasi sangat positif dan gradasi sangat negatif. Untuk menghitung skor pada kuesioner, *SUS*

mempunyai aturan tersendiri. Untuk pernyataan nomor ganjil, nilai skor yang dijawab pada kuesioner dikurangi dengan satu. Sedangkan untuk pernyataan dengan nomor genap, angka lima dikurangi dengan nilai skor yang dijawab. Kemudian semua skor dijumlahkan selanjutnya dikalikan dengan angka 2,5. Skor *SUS* memiliki range nilai 0 – 100. Skor *SUS* yang diperoleh dari seluruh responden kemudian dihitung nilai rata – ratanya dengan menggunakan rumus sebagai berikut:

$$\bar{x} = \frac{\sum x}{N}$$

Keterangan:

$\bar{x}$  : Rata - Rata

$\sum x$  : Jumlah Skor

N : Jumlah Responden

Suatu *range* nilai yang dapat digunakan untuk membantu dalam menentukan apakah skor *SUS* yang diperoleh menunjukkan suatu aplikasi dapat diterima baik atau tidak dari segi *usability*. Sehingga pada penelitian ini, setelah diketahui skor rata – rata *SUS* maka skor tersebut dibandingkan dengan *range* nilai.

**Tabel 3.3 Skor *SUS***

Skor <i>SUS</i>	Interpretasi
< 50	<i>Not Acceptable</i>
50 – 70	<i>Marginal</i>
> 70	<i>Acceptable</i>

Selain itu, dilakukan pengujian reliabilitas instrumen *usability* dengan *internal consistency*, karena instrumen hanya dicobakan sekali saja kepada responden.

## **BAB IV**

### **HASIL PENELITIAN DAN PEMBAHASAN**

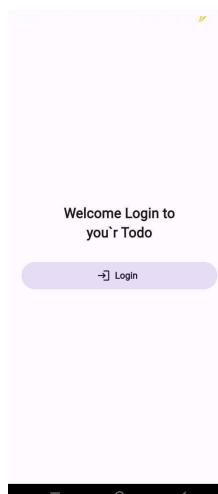
#### **A. Hasil Penelitian**

Berdasarkan hasil penelitian yang dilakukan dari berbagai aspek seperti *Functionality*, *Portability*, *Efficiency*, dan *Usability*. Maka diperoleh hasil sebagai berikut.

##### **1. Fitur Aplikasi yang Dirancang**

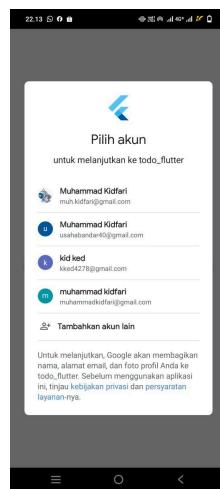
Peneliti merancang aplikasi mobile *To Do List*, dengan menggunakan 2 kerangka kerja (*framework*) yaitu *React Native* dan *Flutter*, kedua *framework* digunakan untuk membuat aplikasi yang serupa yang nanti akan diujikan stress testing, tujuan dari pengujian tersebut adalah untuk mengetahui tingkat efektifitas antara dua *framework* tersebut. Berikut fitur-fitur dari aplikasi *To do list* yang dirancang adalah :

###### **a. Login Screen**



**Gambar 4.1. Login Screen**

Layar *login* pada aplikasi adalah tempat di mana kita memasukkan email dan kata sandi untuk masuk. Di sana, biasanya ada tombol masuk, opsi lupa kata sandi, dan tautan kebijakan privasi. Kadang-kadang, kita juga bisa *login* dengan akun media sosial. Kalau ada kesalahan, seperti memasukkan informasi yang salah, biasanya ada pesan untuk memberi tahu kita. Ada juga tautan untuk membuat akun baru jika belum punya. Tujuannya adalah memudahkan kita masuk ke aplikasi dengan cepat dan tanpa ribet. Terdapat pilihan login dengan *Google* untuk lebih memberikan kesan yang fleksibel bagi pengguna. Berikut tampilan jika *login* dengan *Google* :



**Gambar 4.2. Opsi *Login Google***

Penambahan opsi *login* dengan *Google* dalam suatu aplikasi sering dianggap sebagai pilihan yang baik. Dalam konteks ini, "bagus" dapat diartikan sebagai memberikan keuntungan praktis bagi pengguna. Dengan menggunakan *login Google*, pengguna dapat mengakses aplikasi tanpa harus mengingat kata sandi tambahan, meningkatkan kenyamanan. Keamanan juga bisa ditingkatkan karena metode otentikasi *Google* yang kuat. Selain itu, penggunaan *login* dengan *Google* bisa meningkatkan basis pengguna karena kemudahan akses.

### b. To-Do Screen



**Gambar 4.3. To-Do Screen**

*To-Do Screen* pada aplikasi *To-Do List* ini merupakan *page* yang menampilkan keseluruhan *To-Do* yang masih dalam proses penggerjaan atau belum dikerjakan, tampilan *page* ini berbentuk *list* untuk mempermudah user melakukan pemeriksaan dan melihat *To-Do* apa saja yang belum di kerjakan. dan jika sudah selesai maka user dapat mengklik dalam page ini *user* juga dapat menhapus *To-Do* yang sudah di hapus sebelumnya, dan juga dapat melakukan *log-out*.

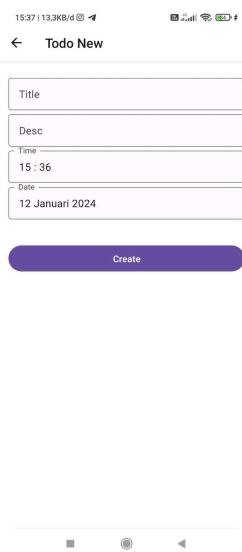
### c. Done Screen



**Gambar 4.4. Done Screen**

*Done Screen* pada aplikasi *To-Do List* ini merupakan *page* yang menampilkan keseluruhan *To-Do* yang sudah di kerjakan, tampilan *page* ini berbentuk *list* untuk mempermudah user melakukan pemeriksaan dan melihat *To-Do* apa saja yang sudah di kerjakan. dalam *page* ini. *User* juga dapat membatalkan ceklis, dan akan di kembalikan ke daftar *To-Do List* yang belum selesai. dalam *page* ini user juga dapat menhapus *To-Do* yang sudah di hapus sebelumnya, dan juga dapat melakukan *log-out*.

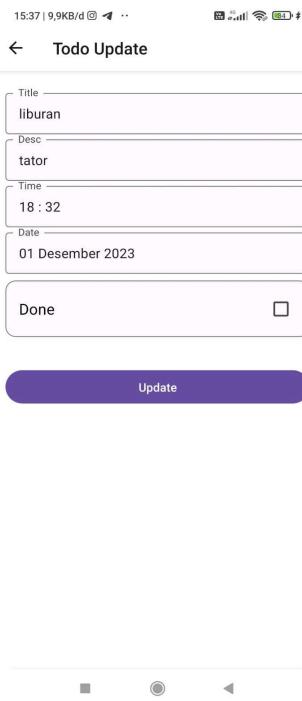
#### d. *Add To-do Screen*



**Gambar 4.5 Add To-do Screen**

*Done Screen* pada aplikasi *To-Do List* ini merupakan *page* yang menampilkan keseluruhan *To-Do* yang sudah di kerjakan, tampilan *page* ini berbentuk *list* untuk mempermudah *user* melakukan pemeriksaan dan melihat *To-Do* apa saja yang sudah di kerjakan. dalam *page* ini. *User* juga dapat membatalkan ceklis, dan akan di kembalikan ke daftar *To-Do List* yang belum selesai.

### e. Update To-do Screen

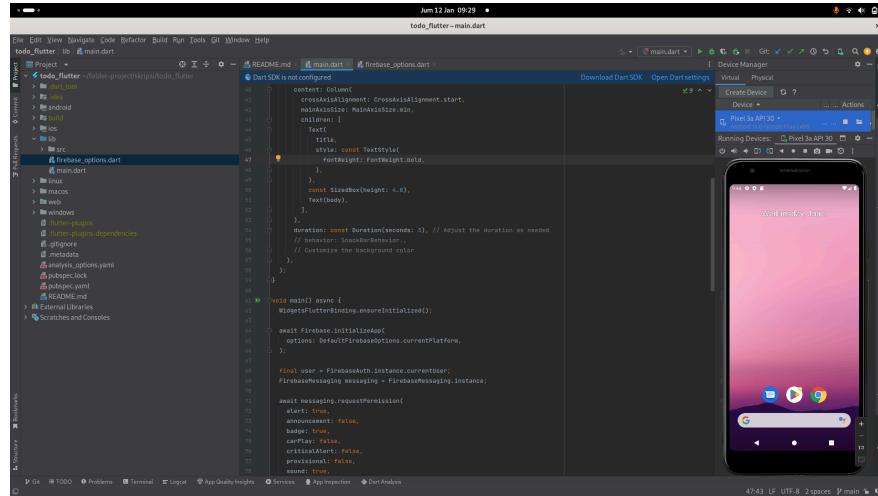


**Gambar 4.6. Update To-do Screen**

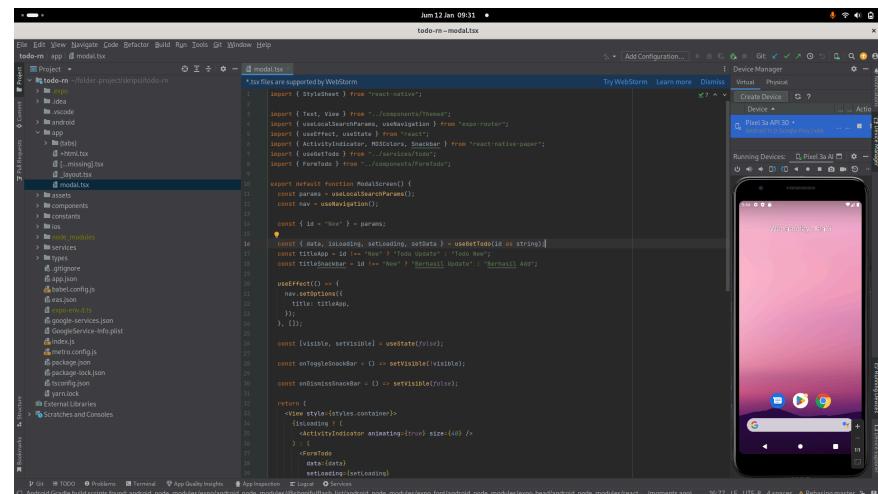
*Done Screen* pada aplikasi *To-Do List* ini merupakan *page* yang menampilkan keseluruhan *To-Do* yang sudah di kerjakan, tampilan *page* ini berbentuk *list* untuk mempermudah user melakukan pemeriksaan dan melihat *To-Do* apa saja yang sudah di kerjakan. dalam *page* ini. *User* juga dapat membatalkan ceklis, dan akan di kembalikan ke daftar *To-Do List* yang belum selesai.

## 2. Pengkodean

Setelah *prototipe* disetujui oleh pihak pengevaluasi, selanjutnya prototipe dikembangkan dalam bentuk aplikasi berbasis mobile di kedua *framework* *Flutter* dan *React Native* menggunakan *Android Studio*, dan *Firebase*. berikut hasil uji coba materi dalam implementasi *To-do List*.



**Gambar 4.7 Proses pengembangan *Flutter* dari *Android Studio***



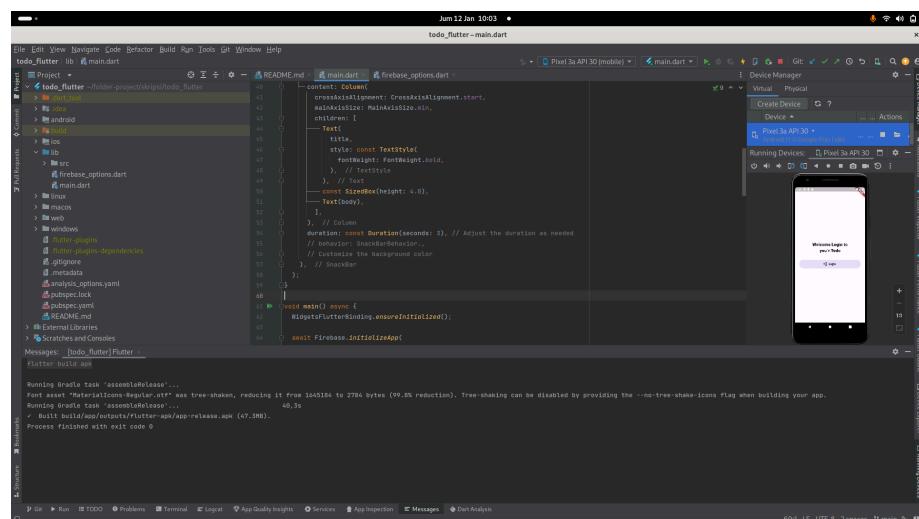
**Gambar 4.8 Proses pengembangan *React Native* dari *Android Studio***

Setelah melakukan pengembangan *To-do List*, pengembang selanjutnya membuat desain *layout* dalam pada android studio serta membuat *logic* aplikasi dengan menggunakan bahasa *Dart* untuk *Flutter* dan juga bahasa *TypeScript* untuk *React Native* pada *android studio*.

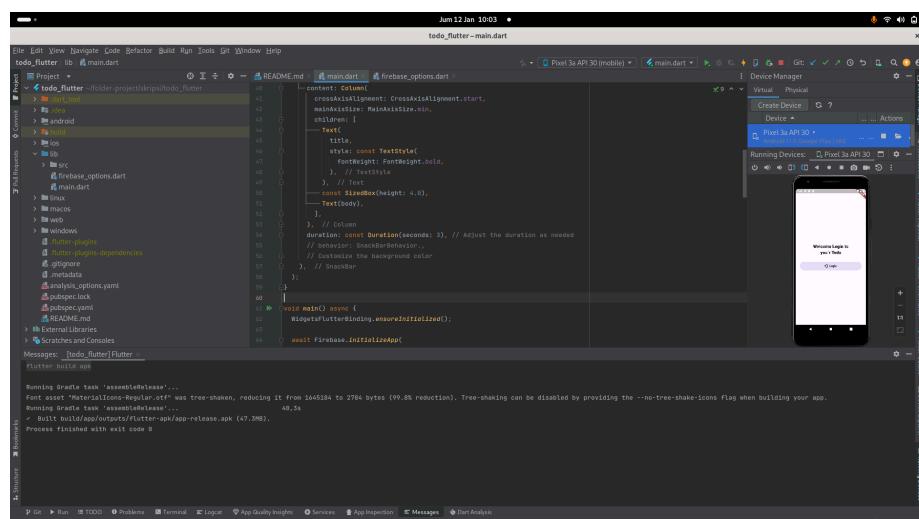
### 3. Pengujian Sistem

Hasil pengujian sistem digunakan untuk mengetahui apakah terdapat *bug* atau error sebelum melanjutkan ke tahap evaluasi secara menyeluruh. Berikut

hasil uji sistem saat membangun aplikasi *Flutter* dan *React Native* dalam *Android Studio*.



**Gambar 4.9 Hasil Uji Sistem *Flutter* dari *Android Studio***



Gambar 4.10. Hasil Uji Sistem *React Native* dari *Android Studio*

Dari gambar di atas dapat dilihat bahwa hasil *build* atau pengembangan aplikasi saat melewati beberapa tahap pengecekan *bug* atau *error* oleh Android Studio berhasil atau menampilkan pesan "*Process finished with exit code 0*". Hal ini menandakan bahwa aplikasi siap untuk dievaluasi secara menyeluruh.

#### 4. Validasi Instrumen

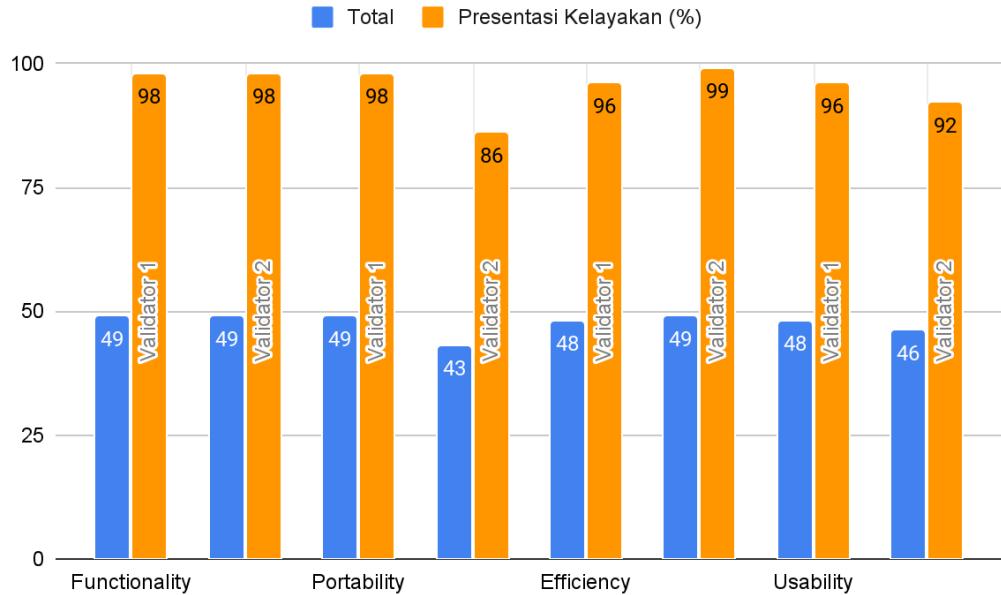
Untuk uji kelayakan uji *Functionality*, *Portability*, *Efficiency*, dan *Usability*. Peneliti terlebih dahulu melakukan validasi instrumen dengan rekap hasil seperti tabel 4.1 berikut :

**Tabel 4.1. Rekapitulasi Hasil Validasi Instrumen Kelayakan Materi**

No	Validator	Aspek	Skor	Total	Presentasi Kelayakan	Keterangan	
<b>Instrumen Uji Kelayakan <i>Functionality</i></b>							
1	Validator 1	Petunjuk	14	49	98%	Sangat Layak	
		Isi	15				
		Bahasa	20				
2	Validator 2	Petunjuk	14	49	98%	Sangat Layak	
		Isi	15				
		Bahasa	20				
<b>Rata-Rata</b>			<b>98</b>	<b>98%</b>	<b>Sangat Layak</b>		
<b>Instrumen Uji Kelayakan <i>Portability</i></b>							
1	Validator 1	Petunjuk	14	49	98%	Sangat Layak	
		Isi	15				
		Bahasa	20				
2	Validator 2	Petunjuk	14	43	86%	Sangat Layak	
		Isi	15				
		Bahasa	20				
<b>Rata-Rata</b>			<b>92</b>	<b>92%</b>	<b>Sangat Layak</b>		
<b>Instrumen Uji Kelayakan <i>Efficiency</i></b>							
1	Validator 1	Petunjuk	14	48	96%	Sangat Layak	
		Isi	15				

No	Validator	Aspek	Skor	Total	Presentasi Kelayakan	Keterangan	
		Bahasa	20				
2	Validator 2	Petunjuk	14	49	99%	Sangat Layak	
		Isi	15				
		Bahasa	20				
<b>Rata-Rata</b>				<b>97</b>	<b>97%</b>	<b>Sangat Layak</b>	
<b>Instrumen Uji Kelayakan Usability</b>							
1	Validator 1	Petunjuk	14	48	96%	Sangat Layak	
		Isi	15				
		Bahasa	20				
2	Validator 2	Petunjuk	14	46	92%	Sangat Layak	
		Isi	15				
		Bahasa	20				
<b>Rata-Rata</b>				<b>94</b>	<b>94%</b>	<b>Sangat Layak</b>	

Berdasarkan hasil uji validitas instrumen di atas, dapat disimpulkan bahwa uji *Functionality*, *Portability*, *Efficiency*, dan *Usability* telah terbukti valid dan dapat diandalkan untuk melanjutkan penelitian. berikut diagram yang menggambarkan hasil dari tabel diatas.



**Gambar 4.11. Diagram Hasil Validasi Instrumen**

Diagram kolom tersebut menyajikan hasil uji kelayakan aplikasi dalam empat kategori, yaitu Functionality, Portability, Efficiency, dan Usability, yang dinilai oleh dua validator (Validator 1 dan Validator 2).

a. *Functionality*:

1. Validator 1 memberikan skor sebesar 49, mencapai presentasi kelayakan sebesar 98%.
2. Validator 2 memberikan skor sebesar 49, dengan presentasi kelayakan 98%.

b. *Portability*:

1. Validator 1 memberikan skor sebesar 49, mencapai presentasi kelayakan sebesar 98%.
2. Validator 2 memberikan skor sebesar 43, dengan presentasi kelayakan 86%.

c. *Efficiency*:

1. Validator 1 memberikan skor sebesar 48, mencapai presentasi kelayakan sebesar 96%.
2. Validator 2 memberikan skor sebesar 49, dengan presentasi kelayakan 99%.

d. *Usability*:

1. Validator 1 memberikan skor sebesar 48, mencapai presentasi kelayakan sebesar 96%.
2. Validator 2 memberikan skor sebesar 46, dengan presentasi kelayakan 92%.

Hasil uji kelayakan aplikasi menunjukkan konsistensi antara Validator 1 dan Validator 2, dengan presentasi kelayakan yang tinggi secara umum yakni diatas 90%, dengan keterangan sangat layak.

Proses selanjutnya akan melibatkan pengujian sesuai dengan metodologi penelitian yang telah dirancang, termasuk tahapan identifikasi masalah, implementasi, pengujian, pengumpulan data, analisis, dan presentasi hasil penelitian. Dengan demikian, pengujian ini diharapkan dapat memberikan wawasan yang lebih mendalam mengenai efektivitas *React Native* dan *Flutter* dalam pengembangan aplikasi *To-Do List* berbasis *Mobile*, serta memberikan dasar data yang solid untuk evaluasi lebih lanjut terhadap keduanya.

## **5. *Functionality Testing***

Data uji coba dalam pengujian ini yaitu 10 orang untuk menguji aplikasi *ToDo List* pada aspek *Functionality* dengan mengisi lembar observasi aspek

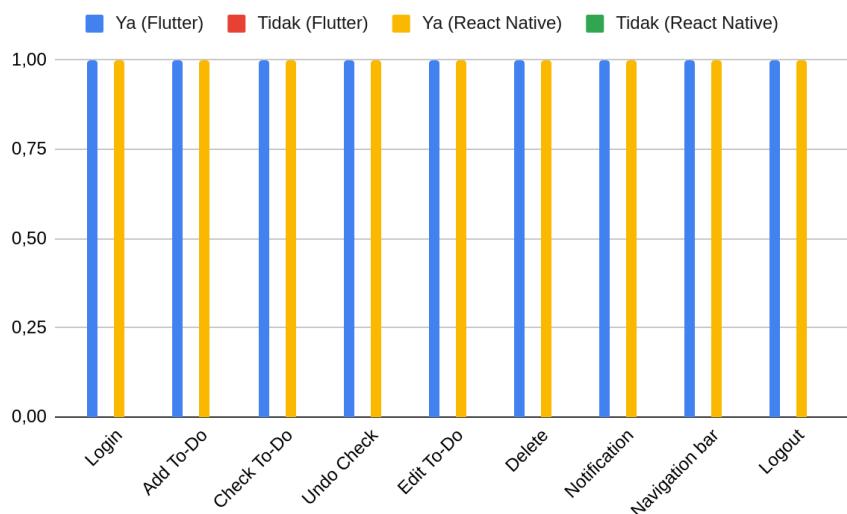
*Functionality* yang telah disediakan. Data diambil dengan melakukan penambahan *script*. Pengujian dilakukan dengan memberikan lembar observasi aspek *Functionality* kepada anggota di organisasi Kedai *Computerworks*. Hasil pengujian aspek *Functionality* dapat dilihat pada Tabel 4.1.

**Tabel 4.2. Hasil Analisa *Functionality***

No	Fungsi	Hasil Yang diharapkan	Hasil			
			Flutter		React Native	
			Ya	Tidak	Ya	Tidak
1	<i>Login</i>	Halaman menu utama terbuka sesuai dengan Akun yang dibuka oleh <i>User</i>	1	0	1	0
2	<i>Add To-Do</i>	<i>To-Do</i> yang telah di <i>Update</i> dapat disimpan pada <i>Database</i> dan menampilkan halaman <i>view To-Do</i>	1	0	1	0
3	<i>Check To-Do</i>	<i>To-Do</i> yang Telah dibuat dapat ditandai Jika telah selesai belum selesai	1	0	1	0
4	<i>Undo Check</i>	<i>To-Do</i> yang telah di tandai selesai dapat di <i>Undo Check</i> bila kegiatan belum selesai sepenuhnya	1	0	1	0
5	<i>Edit To-Do</i>	<i>User</i> dapat Melakukan <i>Update</i> Pada <i>To-Do</i> yang Sebelumnya telah dibuat	1	0	1	0
6	<i>Delete</i>	<i>User</i> dapat Melakukan <i>Delete</i> Pada <i>To-Do</i> yang tidak lagi di perlukan atau telah lewat masa nya	1	0	1	0
7	<i>Notification</i>	<i>User</i> akan mendapatkan Notifikasi Pada <i>To-Do</i> yang telah mencapai waktu yang telah ditentukan	1	0	1	0
8	<i>Navigation bar</i>	Halaman berpindah ketika menekan <i>Button</i> Di bagian	1	0	1	0

No	Fungsi	Hasil Yang diharapkan	Hasil			
			Flutter		React Native	
			Ya	Tidak	Ya	Tidak
		bawah aplikasi sesuai halaman yang di tekan				
9	<i>Logout</i>	kembali ke halaman <i>Login</i>	1	0	1	0

Berdasarkan hasil pengujian, persentase analisa *functionality* aplikasi *To-Do List* dengan hasil 100% *Flutter* dan *React Native*. Berikut adalah diagram yang menggambarkan hasil dari tabel diatas.



**Gambar 4.12. Diagram Hasil Analisa *Functionality***

Dari diagram diatas, kualitas perangkat lunak dari sisi *functionality* telah sesuai dengan atribut *functionality* dan mempunyai skala yang sangat tinggi karena semua fungsi telah berjalan dengan baik. Berikut adalah deskripsi dan diagram berdasarkan data yang diberikan:

a. *Login*:

1. Fungsi: Membuka halaman menu utama sesuai dengan akun pengguna.

2. Hasil: *Flutter* - Ya, *React Native* - Ya
  3. Deskripsi: Fitur *login* pada *Flutter* berhasil diimplementasikan, sedangkan pada *React Native* berhasil diimplementasikan.
- b. *Add To-Do*:
1. Fungsi: Menyimpan *To-Do* yang diperbarui pada database dan menampilkan halaman view *To-Do*.
  2. Hasil: *Flutter* - Ya, *React Native* - Ya
  3. Deskripsi: Fitur menambahkan *To-Do* pada *Flutter* berhasil diimplementasikan, sedangkan pada *React Native* berhasil diimplementasikan.
- c. *Check To-Do*:
1. Fungsi: Menandai *To-Do* sebagai selesai atau belum selesai.
  2. Hasil: *Flutter* - Ya, *React Native* - Ya
  3. Deskripsi: Fitur menandai *To-Do* pada *Flutter* berhasil diimplementasikan, sedangkan pada *React Native* berhasil diimplementasikan.
- d. *Undo Check*:
1. Fungsi: Membatalkan tanda selesai pada *To-Do*.
  2. Hasil: *Flutter* - Ya, *React Native* - Ya
  3. Deskripsi: Fitur membatalkan tanda selesai pada *Flutter* berhasil diimplementasikan, sedangkan pada *React Native* berhasil diimplementasikan.
- e. *Edit To-Do*:

1. Fungsi: Pengguna dapat memperbarui detail *To-Do* yang telah dibuat sebelumnya.
  2. Hasil: *Flutter* - Ya, *React Native* - Ya
  3. Deskripsi: Fitur mengedit *To-Do* pada *Flutter* berhasil diimplementasikan, sedangkan pada *React Native* berhasil diimplementasikan.
- f. *Delete*:
1. Fungsi: Pengguna dapat menghapus *To-Do* yang tidak lagi diperlukan atau telah lewat waktu.
  2. Hasil: *Flutter* - Ya, *React Native* - Ya
  3. Deskripsi: Fitur menghapus *To-Do* pada *Flutter* berhasil diimplementasikan, sedangkan pada *React Native* berhasil diimplementasikan.
- g. *Notification*:
1. Fungsi: Pengguna mendapatkan notifikasi untuk *To-Do* yang telah mencapai waktu yang ditentukan.
  2. Hasil: *Flutter* - Ya, *React Native* - Ya
  3. Deskripsi: Fitur notifikasi pada *Flutter* berhasil diimplementasikan, sedangkan pada *React Native* berhasil diimplementasikan.
- h. *Navigation Bar*:
1. Fungsi: Halaman berpindah ketika menekan tombol di bagian bawah aplikasi sesuai halaman yang ditekan.
  2. Hasil: *Flutter* - Ya, *React Native* - Ya

3. Deskripsi: Fitur navigasi pada *Flutter* berhasil diimplementasikan, sedangkan pada *React Native* berhasil diimplementasikan.
  - i. *Logout*:
    1. Fungsi: Kembali ke halaman *login*.
    2. Hasil: *Flutter* - Ya, *React Native* - Ya
    3. Deskripsi: Fitur *logout* pada *Flutter* berhasil diimplementasikan, sedangkan pada *React Native* berhasil diimplementasikan.

## 6. *Portability Testing*

Pengujian *portability* dimaksudkan mengetahui kemampuan aplikasi *To-Do List* untuk berjalan pada lingkungan yang berbeda. Pengujian dilakukan pada lingkungan sistem operasi dan ukuran layar yang berbeda. Indikator suatu perangkat lunak memenuhi dari aspek *portability* yaitu apabila perangkat lunak tersebut masih dapat dioperasikan dengan baik tanpa kehilangan fungsionalitas aslinya. Berikut merupakan beberapa tampilan hasil pengujian dari segi *portability*.

**Tabel 4.3. Hasil Analisa *Portability***

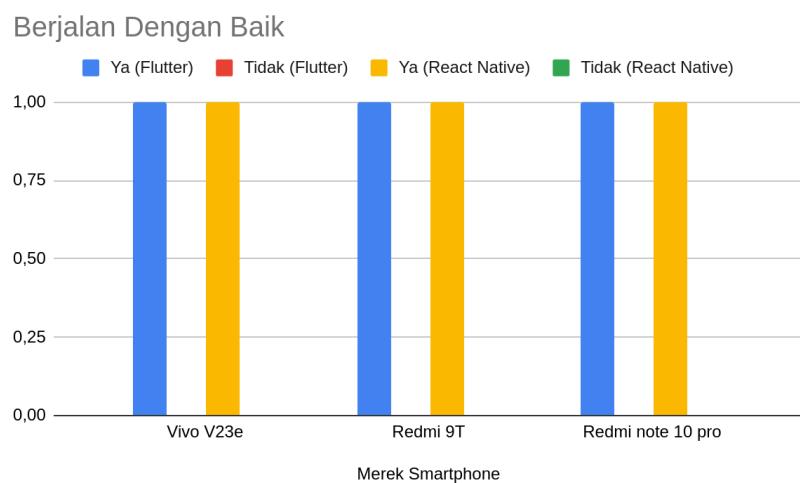
No	Merek <i>Smartphone</i>	Versi	Resolusi Layar	Hasil	
				<i>Flutter</i>	<i>React Native</i>
1	Vivo V23e	12	2400×1080	Berjalan dengan baik	Berjalan dengan baik
2	Redmi 9T	10	2340 x 1080	Berjalan dengan baik	Berjalan dengan baik
3	Redmi note 10 pro	12	2400 x 1080	Berjalan dengan baik	Berjalan dengan baik

$$\text{Persentase Portability} = \frac{\text{Skor yang diobservasi}}{\text{Skor maksimum}} \times 100\%$$

*Skor yang diharapkan*

$$\text{Hasil} = 3/3 \times 100\% = 100\%$$

Dari hasil pengamatan, kedua *framework* didapatkan hasil pengujian dengan nilai hasil 100%. Berikut adalah diagram yang menggambarkan hasil dari tabel diatas.



**Gambar 4.13. Diagram Hasil Analisa Portability**

Dari diagram diatas, dapat di deskripsikan sebagai berikut.

1. Vivo V23e

Aplikasi *To-Do List* yang dikembangkan dengan menggunakan *Flutter* dan *React Native* telah diuji pada Vivo V23e dengan versi *Android* 12 dan resolusi layar  $2400 \times 1080$ . Hasil pengujian menunjukkan bahwa aplikasi berjalan dengan baik pada perangkat ini baik menggunakan *Flutter* maupun *React Native*. Kinerja yang konsisten ini menandakan tingkat kompatibilitas yang tinggi pada smartphone ini, memberikan pengalaman pengguna yang memuaskan.

## 2. Redmi 9T

Uji coba dilakukan pada Redmi 9T dengan versi *Android* 10 dan resolusi layar 2340 x 1080. Baik *Flutter* maupun *React Native* memberikan hasil yang positif, dengan aplikasi *To-Do List* berjalan lancar pada perangkat ini. Hal ini mengindikasikan bahwa framework yang digunakan dapat diandalkan untuk memberikan pengalaman pengguna yang serupa pada berbagai model dan versi perangkat *Android*.

## 3. Redmi Note 10 Pro

Redmi Note 10 Pro, dengan versi *Android* 12 dan resolusi layar 2400 x 1080, juga menghasilkan performa yang baik pada pengujian aplikasi *To-Do List*. Baik *Flutter* maupun *React Native* menunjukkan kemampuan adaptasi yang kuat terhadap perangkat ini. Dengan demikian, dapat disimpulkan bahwa aplikasi ini dapat diandalkan dan konsisten dalam berbagai kontek perangkat *Android*.

Sehingga dapat disimpulkan bahwa aplikasi memenuhi aspek portability. artinya aplikasi *To-Do List* dapat berjalan dengan baik dan tidak ditemukan error pada Sistem Operasi dan Resolusi Layar yang berbeda-beda.

## 7. *Efficiency Testing*

Pengujian *performance efficiency* dilakukan dengan menghitung rata-rata waktu respon aplikasi untuk launching dan rendering. Pada pengujian dilakukan 3 kali dengan tiga smartphone berbeda dan spesifikasi berbeda sebagai berikut.

1. Tes pertama *smartphone* Android Vivo V23e yang memiliki spesifikasi : ukuran layar 6.44 inch 2400×1080 (FHD+); Kamera Depan: 50MP AF, Kamera Belakang: 64MP AF + 8MP *Super Wide-Angle* + 2MP *Super Macro*;

RAM 8 GB/128 GB; CPU *MediaTek Helio G96*; dan OS Android 12 (*Funtouch*).

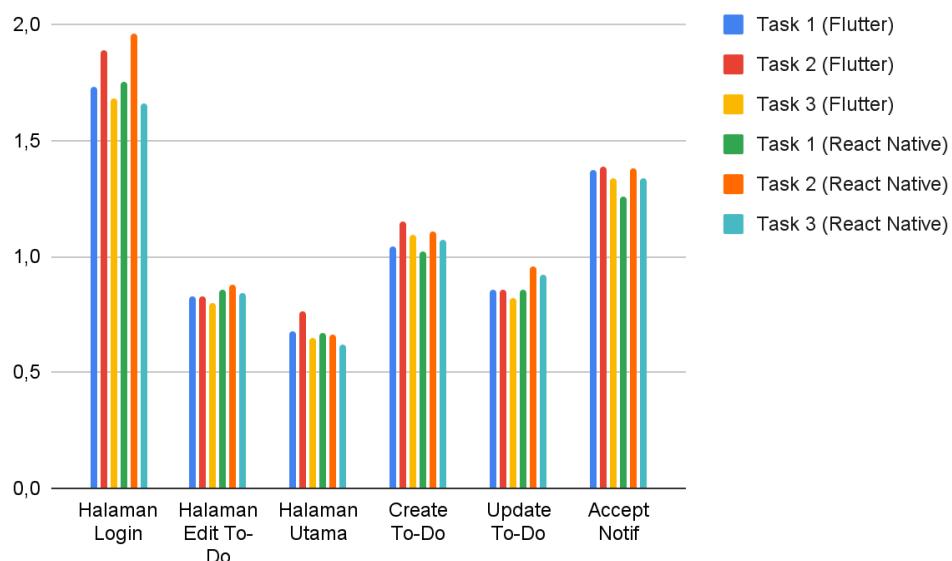
2. Tes kedua *smartphone* Android Redmi 9T yang memiliki spesifikasi: ukuran layar 6.54 inch 2340 x 1080 *Full HD Plus*; kamera selfie 8 MP (f/2.0), kamera utama 48 MP (f/1.8), kamera *makro* 2 MP (f/2.4), kamera *depth sensor* 2 MP (f/2.4), dan kamera *ultra wide* 8 MP (f/2.2); RAM 6 GB/128 GB; *chip Qualcomm Snapdragon 662*; dan OS Android 10.
  3. tes ketiga *smartphone* Android Redmi note 10 pro yang memiliki spesifikasi: ukuran layar 6.67 inch (2400 x 1080 FHD+); kamera depan 16MP, kamera *wide-angle* 108 MP, kamera *ultra-wide angle* 8 MP, kamera *telemakro* 5 MP, sensor *depth* 2 MP; RAM 2 GB; CPU *Snapdragon 732G*; OS Android 12 (R).

Pengujian dilakukan 3 kali kemudian dihitung rata-ratanya. Berikut adalah hasil dari pengujian yang dilakukan terlampir pada Tabel 4.3.

**Tabel 4.4. Hasil Analisa *Efficiency***

No	Kategori Tugas	Response Time (s)					
		Flutter			React Native		
		Task 1	Task 2	Task 3	Task 1	Task 2	Task 3
1	Create To-Do	1,04	1,15	1,09	1,02	1,11	1,07
2	Update To-Do	0,86	0,86	0,82	0,86	0,96	0,92
3	Accept Notif	1,37	1,39	1,34	1,26	1,38	1,34
<b>Jumlah</b>		9,96	10,36	9,81	9,9	10,45	9,89
<b>Rata – Rata</b>		1,42	1,48	1,40	1,41	1,49	1,41
<b>Total Rata – Rata Response Time (ms)</b>		<b>1.44</b>			<b>1.43</b>		

Dari hasil pengujian didapatkan bahwa jenis prosesor dan RAM mempengaruhi performa aplikasi. hasil pengujian yang dilakukan dapat di lihat pada diagram berikut yang menggambarkan hasil dari tabel diatas.



**Gambar 4.14. Diagram Hasil Analisa Efficiency**

Dari diagram diatas, dapat di deskripsikan sebagai berikut.

a. Menampilkan Data:

1. Halaman *Login*: Pengujian respon waktu pada halaman *login* menunjukkan hasil yang seragam antara *Flutter* dan *React Native*. *Flutter* memiliki rata-rata respon waktu 1,42 detik, sedangkan *React Native* mencatat rata-rata 1,43 detik. Meskipun terdapat sedikit perbedaan, namun keduanya masih berada dalam rentang yang dapat diterima, menjamin pengguna mendapatkan pengalaman yang cepat dan responsif saat membuka halaman *login*.
2. Halaman Edit *To-Do*: Respon waktu untuk mengedit *To-Do* pada halaman ini juga seragam antara *Flutter* dan *React Native*. Keduanya menunjukkan respon waktu yang baik, dengan rata-rata masing-masing 1,48 detik untuk *Flutter* dan 1,49 detik untuk *React Native*. Perbedaan ini kecil dan tidak signifikan, menunjukkan performa yang setara pada tugas ini.
3. Halaman Utama: Pada halaman utama, baik *Flutter* maupun *React Native* memberikan hasil respon waktu yang konsisten. *Flutter* mencapai rata-rata respon waktu sebesar 1,40 detik, sementara *React Native* memiliki rata-rata 1,41 detik. Keduanya memberikan kinerja yang memadai untuk menampilkan data pada halaman utama.

b. Mengambil Data:

1. *Create To-Do*: Respon waktu untuk membuat *To-Do* menunjukkan hasil yang konsisten antara *Flutter* dan *React Native*. Rata-rata respon waktu adalah 1,42 detik untuk *Flutter* dan 1,41 detik untuk *React Native*. Keduanya memberikan performa yang sebanding dalam tugas ini.

2. *Update To-Do:* Halaman update *To-Do* memberikan hasil respon waktu yang hampir identik antara *Flutter* (rata-rata 1,48 detik) dan *React Native* (rata-rata 1,49 detik). Perbedaan yang sangat kecil ini menandakan bahwa keduanya memiliki performa yang setara dalam mengambil data untuk melakukan pembaruan.
3. *Accept Notification:* Respon waktu untuk menerima notifikasi juga menunjukkan hasil yang seragam. *Flutter* mencapai rata-rata 1,41 detik, sedangkan *React Native* mencapai 1,41 detik. Kedua *framework* memberikan kinerja yang baik dalam mengambil data notifikasi.

Total Rata-Rata Respon Time:

- a. *Flutter*: 1,44 detik
- b. *React Native*: 1,43 detik

Kedua framework menunjukkan kinerja yang serupa dalam tugas-tugas menampilkan dan mengambil data Pengujian pada tes 1 dan 3 mendapatkan hasil lebih cepat dibandingkan dengan tes 2. dan itu terjadi di kedua *framework* *Flutter* dan *React Native*. Hal ini dikarenakan untuk tes 1 dan 3 menggunakan prosesor dan RAM tertinggi. Untuk hasil perbandingan di antara keduanya dihasilkan *Flutter* : 1.44 dan *React Native* : 1.43.

## **8. Usability Testing**

Langkah awal *usability* testing ini adalah memberikan sejumlah task atau tugas yang sudah dipersiapkan sebelumnya kepada pengguna saat berinteraksi dengan sistem yang diuji. Task-task ini diberikan kepada 10 responden yang berasal anggota KeDai computerworks yang sudah mengetahui aplikasi yang

dibuat dari *framework Flutter* dan *React Native* sehingga mereka tidak lagi mengalami kesulitan pada saat melakukan task-task tersebut. Task-task ini digunakan sebagai ‘sarana interaksi’ dalam pengukuran usability.

**Tabel 4.5. Task-Task Usability Testing**

No	Nama Task	Penjelasan Taks
1	<i>Task Login</i>	Melibatkan pembuatan sistem otentifikasi pengguna dimana pengguna dapat masuk ke aplikasi menggunakan Email.
2	<i>Task Add To-Do</i>	Pengguna dapat menambahkan tugas atau item To-Do baru ke daftar mereka. Umumnya melibatkan formulir atau antarmuka untuk memasukkan detail tugas.
3	<i>Task Check To-Do</i>	Melibatkan tanda centang atau penandaan bahwa suatu item To-Do telah selesai. Pengguna seharusnya dapat menandai bahwa suatu tugas tertentu telah terselesaikan.
4	<i>Task Undo Check</i>	Pengguna dapat membatalkan tanda centang atau menandai kembali suatu item To-Do sebagai belum selesai setelah sebelumnya ditandai sebagai selesai.
5	<i>Task Edit To-Do</i>	Pengguna perlu kemampuan untuk mengedit detail suatu item To-Do tertentu. Ini bisa mencakup mengubah nama tugas, deskripsi, tanggal jatuh tempo, atau informasi relevan lainnya.

No	Nama Task	Penjelasan Taks
6	<i>Task Delete</i>	Melibatkan memungkinkan pengguna untuk menghapus suatu item To-Do yang tidak lagi diperlukan atau relevan. Penghapusan dapat bersifat permanen atau melibatkan langkah konfirmasi.
7	<i>Task Notification</i>	Pengguna dapat menerima notifikasi terkait dengan item To-Do mereka. Ini dapat mencakup pengingat untuk tugas yang akan datang, pemberitahuan untuk tugas yang lewat waktu, atau notifikasi lainnya untuk menjaga pengguna tetap terinformasi.
8	<i>Task Navigation bar</i>	Bar navigasi menyediakan cara yang ramah pengguna bagi pengguna untuk beralih antara berbagai bagian atau fitur dalam aplikasi. Umumnya mencakup tautan atau tombol untuk Beranda, Daftar To-Do dan daftar Done
9	<i>Task Logout</i>	Pengguna dapat keluar dari akun mereka untuk memastikan keamanan informasi mereka. Ini melibatkan mengakhiri sesi saat ini dan kembali ke layar masuk.

Setelah pengguna menyelesaikan semua task yang ada, langkah selanjutnya adalah membagikan kuesioner kepada Responden yang berisi 11 pertanyaan yang sudah mewakili ketiga aspek usability.

Selanjutnya adalah melakukan rekap terhadap hasil kuesioner yang telah disebarluaskan kepada 10 responden menggunakan SUS. Hasil perhitungan rata-rata

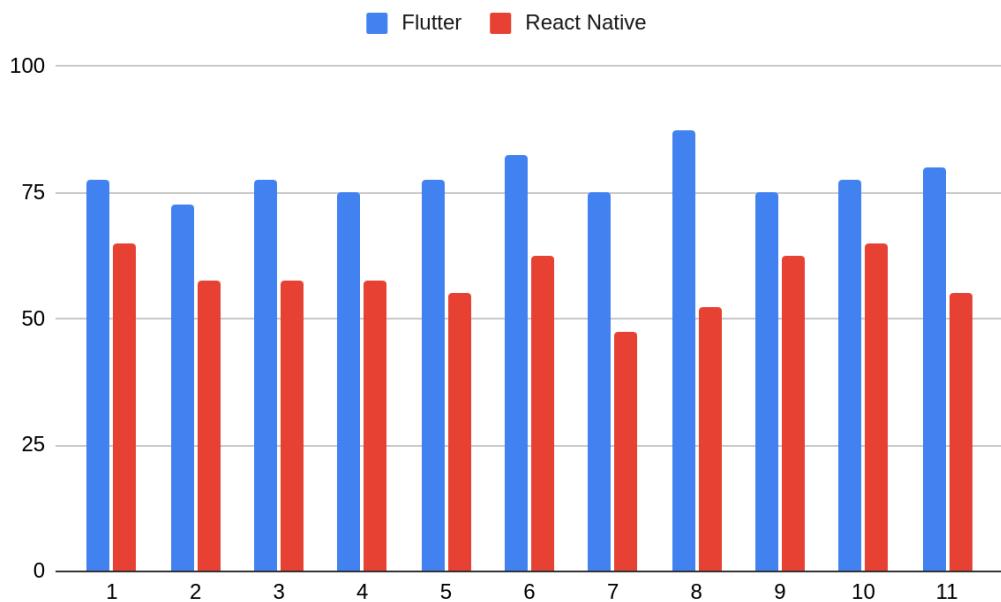
terhadap hasil *usability* testing di atas, maka diperoleh rekap nilai *usability* yang dapat dilihat pada tabel 4.6 di bawah ini.

**Tabel 4.6. Rekap Nilai Usability**

No	Pertanyaan	Nilai	
		Flutter	React Native
<b>Aspek Sistem (SYSTEM) :</b>			
1	Saya menganggap tampilan aplikasi mudah dikenali	77.5	65
2	Saya merasa aplikasi ini mudah dioperasikan	72.5	57.5
3	Menurut saya, tampilan warna pada aplikasi enak dilihat dan tidak membosankan	77.5	57.5
4	Saya merasa fungsi-fungsi dalam aplikasi terintegrasi dengan baik	75	57.7
<b>Aspek Pengguna (USER) :</b>			
5	Saya menganggap tampilan menu dalam aplikasi mudah dikenali	77.5	55
6	Menurut saya, aplikasi ini mudah dibaca	82.5	62.5
7	Saya merasa aplikasi mudah diinstal	75	47.5
8	Menurut saya, simbol-simbol pada aplikasi mudah dipahami	87.5	52.5
<b>Aspek Interaksi (INTERACTION) :</b>			
9	Saya merasa menu dan tampilan aplikasi mudah diingat	75	62.5
10	Menurut saya, aplikasi ini mudah dipelajari	77.5	65
11	Saya menganggap aplikasi ini praktis digunakan	80	55

Berdasarkan tabel 4.6 di atas, dapat dilihat dari atribut “Menurut saya, simbol-simbol pada aplikasi mudah dipahami” untuk *framework Flutter* dan *framework React Native* memiliki nilai diatas 5. Hal ini berarti aplikasi *To-Do List* telah dibuat mudah untuk dikenali oleh user dari segi tampilan antarmuka. Bila disesuaikan dengan plot aspek usability pada table 3.1, aplikasi *To-Do List* telah dibuat dengan memiliki nilai usability, yaitu: Learnability, Efficiency, Memorability, Errors dan Satisfaction yang sangat baik.

Berikut adalah grafik yang menggambarkan perbandingan antara *framework Flutter* dan *React Native*



**Gambar 4.15 Diagram Hasil Analisa Usability**

Grafik diatas menggambarkan perbedaan antara *framework Flutter* dan *React Native* dimana *Flutter* adalah warna biru, dan *React Native* adalah warna merah, yang mana *Usability Testing* menunjukkan *Flutter* lebih unggul dibanding

*React Native*. Hal ini dapat ditunjukkan dengan nilai hasil usability pada kelima atribut, sebagai berikut:

- a. Nilai atribut “Menurut saya, simbol-simbol pada aplikasi mudah dipahami” *Flutter 87.5 & React Native 52.5*. menunjukkan bahwa kedua aplikasi telah memiliki nilai Learnability.
- b. Nilai atribut “Saya merasa menu dan tampilan aplikasi mudah diingat” *Flutter 75 & React Native 62.5*, “Menurut saya, aplikasi ini mudah dipelajari” *Flutter 77.5 & React Native 65*. menunjukkan bahwa aplikasi telah memiliki nilai Efficiency.
- c. Nilai atribut “Saya menganggap tampilan menu dalam aplikasi mudah dikenali” *Flutter 87.5 & React Native 52.5*. menunjukkan bahwa kedua aplikasi telah memiliki nilai Memorability.
- d. Tidak adanya bug atau kegagalan dalam mengoperasikan aplikasi membuat aplikasi *To-Do List* telah meminimalisasi aspek Error.
- e. Keseluruhan atribut yang memiliki nilai rata-rata diatas 3 menunjukkan bahwa aplikasi *To-Do List* mempunyai aspek Satisfaction yang sangat baik.

## B. Pembahasan

Penelitian ini merupakan jenis penelitian yang menerapkan studi kasus dimana memfokuskan perhatian pada penggunaan *framework Flutter* dan *React Native* dalam pembuatan aplikasi *to-do list* berbasis *mobile*. Dalam paradigma studi kasus, penelitian ini mengambil individu atau kelompok sebagai obyek studi, bertujuan untuk menggali data yang mendalam untuk menjawab permasalahan yang dihadapi. Penelitian ini bersifat deskriptif dan eksploratif, membedakannya

dari penelitian survei yang melibatkan jumlah sampel yang lebih luas. Fokus utama dari studi kasus ini adalah membuktikan kualitas perangkat lunak yang dihasilkan menggunakan kedua framework, dengan penekanan pada pengujian performa dan implementasi testabilitas. Pengujian performa melibatkan fitur-fitur yang dapat menguji kinerja memori dan CPU, seperti pengguliran daftar data, animasi, dan perpindahan data. Sementara itu, implementasi testabilitas difokuskan pada pengujian unit dan integrasi, termasuk pemanggilan API dan integrasi beberapa halaman yang saling terhubung.

Dalam penelitian ini, *Firebase* digunakan sebagai platform penyimpanan *to-do list* yang digunakan dalam aplikasi. *Firebase* merupakan platform pengembangan aplikasi yang menyediakan layanan penyimpanan data dalam *cloud*. Dengan menggunakan *Firebase*, *to-do list* yang telah dikumpulkan dalam analisis kebutuhan dapat disimpan secara aman dan dapat diakses oleh pengguna melalui aplikasi. Data pengguna seperti nama dan *to-do list* yang telah dibuat dapat diakses kembali dengan cara login menggunakan email yang sebelumnya dipakai.

Setelah proses pengembangan selesai, dilakukan pengujian aplikasi secara menyeluruh untuk memastikan bahwa aplikasi telah berhasil dibangun tanpa adanya bug atau kesalahan yang signifikan. Pengujian dilakukan dengan melakukan simulasi interaksi pengguna dengan aplikasi, menguji semua fitur yang ada, dan memastikan bahwa aplikasi berjalan dengan baik dan sesuai dengan kebutuhan.

Selanjutnya, dilakukan evaluasi terhadap aplikasi yang telah dikembangkan. Evaluasi dilakukan untuk mengukur kualitas dan kelayakan aplikasi. Evaluasi meliputi beberapa aspek, validitas instrumen aspek *functionalities*, *portability*, *efficiency* dan *usability*. Berdasarkan uji validitas instrumen, instrumen yang digunakan dalam penelitian ini dinilai sangat layak dalam mengukur variabel yang diteliti.

Setelah melakukan pengujian functional, portability, dan efficiency. Peneliti selanjutnya melakukan uji coba kepada subjek penelitian dalam hal ini 10 anggota dari UKM KeDai computerworks dengan menguji fitur-fitur aplikasi dari kedua *Framework Flutter* dan *React Native*. Dari hasil uji coba, Dalam proses instalasi, aplikasi dapat terpasang dan berjalan dengan baik pada 10 perangkat anggota. Dari hasil uji coba fitur, 10 anggota berhasil melakukan uji coba pada fitur yang ada.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa aplikasi *to-do list* dari *framework Flutter* dan *React Native* berhasil dikembangkan tanpa adanya bug. dan keseluruhan evaluasi aplikasi dari *framework Flutter* lebih baik dalam berbagai aspek, seperti validitas instrumen, aspek materi, *functional suitability*, *portability*, *performance efficiency*, dan *usability*.

Adapun hasil pengujian menunjukkan bahwa kualitas perangkat lunak aplikasi *To-Do List* yang dibangun dengan menggunakan *framework Flutter* dan *React Native* mencapai tingkat fungsionalitas maksimal, dengan persentase analisis mencapai 100%. Kedua *framework* juga menunjukkan keunggulan dalam aspek portabilitas, di mana aplikasi dapat berjalan tanpa kesalahan pada berbagai

sistem operasi dan resolusi layar. Meskipun terdapat sedikit perbedaan dalam efisiensi, di mana jenis prosesor dan RAM mempengaruhi performa aplikasi, nilai perbandingan antara *Flutter* (1.44) dan *React Native* (1.43). Aspek *usability* aplikasi *To-Do List* pada kedua framework, *Flutter* dan *React Native*, dapat diukur melalui nilai *Learnability*, *Efficiency*, *Memorability*, *Errors*, dan *Satisfaction*. Dari data yang diberikan, nilai *Learnability* pada *Flutter* (87.5%) unggul dibandingkan *React Native* (52.5%), menandakan kemudahan pengguna dalam mempelajari aplikasi. *Efisiensi* dalam pengingatan menu dan tampilan aplikasi serta kemudahan pembelajaran menunjukkan hasil yang seimbang antara *Flutter* (75%, 77.5%) dan *React Native* (62.5%, 65%). *Memorability*, aspek yang mengukur kemampuan pengguna untuk mengingat tampilan menu, juga menunjukkan perbedaan signifikan antara *Flutter* (87.5%) dan *React Native* (52.5%). Keseluruhan, nilai *Satisfaction* pada *Flutter* (87.5%) dan *React Native* (52.5%) menegaskan tingkat kepuasan pengguna yang tinggi terhadap antarmuka aplikasi. Dengan demikian, hasil pengujian pada perbandingan *usability* menunjukkan keunggulan *Flutter* dalam *Learnability* dan *Memorability*, sementara *React Native* menunjukkan performa yang lebih baik dalam *Efficiency* dan *Satisfaction*, ini sejalan dengan penelitian relevan sebelumnya oleh (Haris, 2021).

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **A. Kesimpulan**

Berdasarkan hasil penelitian dan pembahasan pada Bab IV, dapat disimpulkan bahwa pengembangan aplikasi *To-Do List* menggunakan *framework Flutter* dan *React Native* telah menghasilkan aplikasi yang memiliki kualitas perangkat lunak yang tinggi. Pengujian *fungsionalitas, portabilitas, efisiensi, dan usability* menunjukkan bahwa kedua framework mampu memenuhi kebutuhan aplikasi dengan baik. Secara khusus, aspek fungsionalitas mencapai 100%, menandakan bahwa seluruh fitur aplikasi berjalan dengan baik. Aspek portabilitas menunjukkan bahwa aplikasi dapat beroperasi tanpa masalah pada berbagai sistem operasi dan resolusi layar. Uji efisiensi memperlihatkan adanya pengaruh jenis prosesor dan RAM terhadap performa aplikasi, dengan *Flutter* dan *React Native* yang menunjukkan *Flutter* lebih unggul. Aspek *usability* pada *Flutter* dan *React Native* menunjukkan bahwa *Flutter* unggul dalam *Learnability, Memorability, Efficiency* dan *Satisfaction*.

#### **B. Saran**

Berdasarkan temuan dalam penelitian ini, terdapat beberapa saran yang dapat diberikan:

##### **1. Pengembang kedepannya**

Melakukan pengembangan lanjutan pada aplikasi dengan menambahkan fitur-fitur tambahan yang dapat meningkatkan pengalaman pengguna.

Mengeksplorasi integrasi dengan layanan pihak ketiga untuk meningkatkan fungsionalitas aplikasi.

## 2. Peneliti selanjutnya

Melakukan penelitian lebih lanjut terkait perkembangan teknologi *mobile* dan *framework* pengembangan yang baru. Memperluas penelitian ke berbagai kasus penggunaan atau industri untuk mendapatkan pemahaman yang lebih komprehensif.

Dengan implementasi saran-saran ini, diharapkan dapat meningkatkan kualitas dan keberlanjutan *Framework React Native* dan *Flutter* serta memberikan kontribusi positif terhadap pengembangan teknologi aplikasi *mobile* secara umum.

## **DAFTAR PUSTAKA**

- A. H. Malahella, I. Arwani, and Tibyani, “Pemanfaatan Framework React Native dalam Pengembangan Aplikasi Pemesanan Minuman Kopi pada Kedai Bycoffee,” *J. Pengemb. Teknol. Inf. dan Ilmu Komputer.*, vol. 4, no. 9, pp. 3178–3184, 2020.
- Architecture, F. 2020. Flutter Architecture. Retrieved. November 2020.  
<https://www.javatpoint.com/Flutter-architecture>.
- Boukhary, S., & Colmenares, E. 2019 A Clean Approach to Flutter Development through the Flutter Clean Architecture Package. *IEEE, International Conference on Computational Science and Computational Intelligence (CSCI)*.
- Cardia, Ventica. 2021. Efektivitas Penerapan Sistem Aplikasi Pelayanan Kepegawaian (SAPK) Dalam Pengurusan Dokumen Kepegawaian Pada Kantor Regional XII BKN Pekanbaru. Skripsi. Pekanbaru: UIN Sultan Syarif Kasim Riau.
- Cheng, F. 2019. *Flutter Recipes*. CA: Apress.
- Cosmina, I. 2020. *Building Reactive Applications Using Spring*. CA: Apress.
- Courses, U. 2020. Google Flutter Courses. Retrieved. November 2020.  
<https://www.udemy.com/topic/google-Flutter>.
- Dagne, L. 2019. *Flutter for cross-platform App and SDK development*. Helsinki, Finland: Metropolia University of Applied Sciences.
- Dedi Rianto Rahad, “Pengukuran Usability Sistem Menggunakan Use Questionnaire Pada Aplikasi Android”*Jurnal Sistem Informasi (JSI)*, VOL. 6, NO. 1, April 2014,
- Dev, F. 2020. Flutter Developer Library: Retrieved. November 2020.  
<https://Flutter.dev/showcase>.
- Developers, G. 2018. Google Developer Day at GDC 2018 Livestream. Retrieved. November 2020.

[https://www.youtube.com/watch?v=5wtlj\\_q3DjE&list=PLOU2XLYxmsIIxxDKHWd\\_aI\\_d\\_oV9hqPi7q](https://www.youtube.com/watch?v=5wtlj_q3DjE&list=PLOU2XLYxmsIIxxDKHWd_aI_d_oV9hqPi7q).

Development, R. N. 2020. React Native Developer Library: Retrieved. November 2020. <https://reactnative.dev/showcase>.

Devnot. 2020. Ionic'i Taniyalım. Retrieved. November 2020. <http://devnot.com/2016/hibrit-uygulama- catisi-ionic-i-taniyalim>

Dwi Gusna. 2023. Pengembangan Aplikasi Konsultasi Menggunakan Obrolan Suara Dan Terapi Psikis Berbasis Android Untuk Penyandang Tunanetra. Program Studi Pendidikan Teknik Infromatika & Komputer Jurusan Teknik Informatika & Komputer.

Flutter 2018. Technical Overview. Retrieved. November 2020. <https://Flutter.dev/docs/resources/technical- overview>.

Flutter / Flutter? Javatpoint. 2020. What is Xamarin: Retrieved. <https://www.javatpoint.com/what-is->

Frachet, M. 2020. Understanding The React Native Bridge Concept: Retrieved. November 2020. <https://hackernoon.com/understanding-react-native-bridge-concept-e9526066ddb8>.

Gibson, JL and Ivancevich, J. M. 2010. Organisasi, Struktur dan Manajemen. Jakarta : Erlangga.

GitHub. 2020. A Small Place to Discover Languages in Github. November 2020. <https://madnight.github.io/githut/#/issues/2020/2>.

Goadrich, M. H., & Rogers, M. P. 2011. Smart Smartphone Development: Ios Versus Android. Paper presented at the Proceedings of the 42nd ACM technical symposium on computer science education, New York: USA.

Hansson, N., & Vidhall, T. 2016. Effects on performance and usability for cross-platform application development using React Native. Linköping, Sweden: Linkopings Universitet.

Haris, Zakki Abdurrahman. 2021. "Komparasi Flutter Dan React Native Dalam Pengembangan Perangkat Bergerak.".

- HARUN, FIRDAUS BIN. 2019. "Review of iOS Architectural Pattern for Testability, Modifiability, and Performance Quality." *Journal of Theoretical and Applied Information Technology* 97, no. 15.
- Issues, G. F. 2020. Want To Contribute to <https://github.com/Flutter/Flutter/issues> Erişim tarihi: 28/11/2020.
- Jagiełło, J. 2019. Performance comparison between React Native and Flutter. Umeå, Sweden: Umeå University.
- Kuzmin, N., Ignatiev, K., & Grafov, D. 2020. Experience of Developing a Mobile Application Using Flutter. (pp. 571 - 575). Springer International Publishing.
- Mardiasmo. 2017. Perpajakan. Yogyakarta: Andi Offset.
- Native, R. 2020. React Native Retrieved. November 2020. <https://reactnative.dev>
- Occhino, T. 2020. React Native: Bringing Modern Website Techniques to Mobile: Retrieved. November 2020. <https://engineering.fb.com/android/react-native-bringing-modern-Website-techniques-to-mobile>.
- Pakei, Beni. 2016. Konsep dan Analisis (Efektivitas Pengelolaan Keuangan Daerah di Era Otonomi Daerah). Jayapura : Taushia.
- Payne, R. 2019. Beginning App Development with Flutter: Create Cross-Platform Mobile Apps. CA: Apress, Retrieved from [https://doi.org/10.1007/978-1-4842-5181-2\\_2](https://doi.org/10.1007/978-1-4842-5181-2_2)
- PhoneGap. 2012. PhoneGap Beliefs, Goals, And Philosophy. <https://blog.phonegap.com/phonegap-beliefs-goals-and-philosophy-dc9d1f7d7aca> Erişim tarihi: 28/11/2020.
- S. R. Ramahani, J. N. Sari, I. Lestari dan S. Susiyanti, "Pengembangan Aplikasi Monitoring Penyebaran Virus Covid-19 Berbasis Mobile Area Pekanbaru dengan Prototyping," Inovtek Polbeng - Seri Informatika, vol. 6, no. 1, pp. 37-50, 2021.
- Singh, T., & Bhardwaj, R. 2019. Fuchsia OS-A Threat to Android. *IITM Journal of Management and IT*, 10(1), 65-67.

- Skuza, B., Mroczkowska, A., and Wlodarczyk, D. (2019). Flutter vs. React native – what to choose in 2021? Available. June 22, 2022. <https://www.thedroidsonroids.com/blog/Flutter-vs-react-native-what-to-choose-in-2021>.
- Stackoverflow. 2019. Developer Survey Results. Retrieved. November 2020. <https://insights.stackoverflow.com/survey/2019>.
- Stackoverflow. 2020. Stack Overflow Trends. Retrieved. November 2020. <https://insights.stackoverflow.com/survey/2019#technology>.
- Team, W. A. 2020. React Native on the Universal Windows Platform: Retrieved. November 2020. <https://blogs.windows.com/windowsdeveloper/2016/04/13/react-native-on-the-universal-windows-platform>.
- Trends. 2020. Tag Trends. November 2020. <http://sotagtrends.com/?tags=ionic-framework+react-native+Flutter+xamarin&relative=false>.
- Trends. 2020. React Native - Flutter comparison. (d). Flutter. November 2020.. <https://trends.google.com/trends/explore?q=React%20Native>
- Vasilescu, B., Filkov, V., & Serebrenik, A. 2013. Stackoverflow and Github: Associations Between Software Development and Crowdsourced Knowledge. 2013 International Conference on Social Computing, 188-195.
- Wu, W. 2018. React Native vs Flutter, cross-platform mobile application frameworks. Helsinki, Finland: Metropolia University of Applied Sciences.
- Zammetti, F. 2019. Flutter: A Gentle Introduction. APress, CA: Apress.



# **LAMPIRAN 1**

## **PERSURATAN**

**Lampiran 1.1 Surat Permintaan Izin Penelitian**



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI  
UNIVERSITAS NEGERI MAKASSAR (UNM)  
**FAKULTAS TEKNIK**

Alamat: Jalan Daeng Tata Raya Parangtambung Makassar  
Telp (0411) 865677 – Fax. (0411) 861377  
Laman: [ft.unm.ac.id](http://ft.unm.ac.id)

---

Nomor : 4803/UN36.2/PP/OL/2023

21 Nopember 2023

Lampiran : 1 (satu) berkas

Hal : **Permintaan Izin Penelitian**

Yth,  
Gubernur Sulawesi Selatan c.q. Kepala UPT P2T BKPM Prov. Sulawesi Selatan  
di-  
Makassar

Disampaikan bahwa mahasiswa tersebut di bawah ini:

Nama : **Muh. kidfari**

NIM : 1929041021

Program Studi : Pendidikan Teknik Informatika dan Komputer - S1

Akan mengadakan penelitian dalam rangka penulisan Skripsi yang berjudul:

**"Analysis Tingkat Efektifitas antara React Native dan Flutter dalam Pembuatan Aplikasi ToDo List Berbasis Mobile"**

Penelitian yang dimaksud direncanakan berlangsung selama kurang lebih 3 (Tiga) bulan dibawah koordinasi dosen pembimbing:

Prof. Dr. Ir. H. Muhammad Yahya, M.Kes.,  
M.Eng., IPU., ASEAN Eng.  
196306231991031002

Dr. Ir. Ridwan Daud Mahande, S.Pd.,  
M.Pd., IPM.  
198408042019031005

Sehubungan dengan judul tersebut di atas, maka kami mohon kiranya mahasiswa yang bersangkutan dapat diizinkan melakukan penelitian di **Study Club KeDai Computerworks** guna penulisan Skripsi.

Demikian penyampaian kami, atas perkenaan Bapak/Ibu diucapkan terima kasih.

Makassar, 21 Nopember 2023  
a.n Dekan  
Wakil Dekan Bidang Akademik



**Prof. Dr. Drs. Ir. Jamaluddin P, MP., IPM.  
NIP. 196707231992031002**

**Lampiran 1.2 Surat Izin penelitian**



**PEMERINTAH PROVINSI SULAWESI SELATAN**  
**DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU**

Jl. Bougenville No.5 Telp. (0411) 441077 Fax. (0411) 448936  
Website : <http://simap-new.suselprov.go.id> Email : [ptsp@suselprov.go.id](mailto:ptsp@suselprov.go.id)  
Makassar 90231

Nomor : **29870/S.01/PTSP/2023** Kepada Yth.  
Lampiran : - Rektor Univ. Dipanegara Makassar  
Perihal : **Izin penelitian**

di-  
**Tempat**

Berdasarkan surat Pembantu Dekan Bid. Akademik FT UNM Makassar Nomor :  
4803/UN36.2/PP/OL/2023 tanggal 21 Nopember 2023 perihal tersebut diatas, mahasiswa/peneliti  
dibawah ini:

N a m a : MUH.KIDFARI  
Nomor Pokok : 1929041021  
Program Studi : Pend. Teknik Informatika Dan Komputer  
Pekerjaan/Lembaga : Mahasiswa (S1)  
Alamat : Jalan Daeng Tata Raya Parangtambung Makassar  
PROVINSI SULAWESI SELATAN

Bermaksud untuk melakukan penelitian di daerah/kantor saudara dalam rangka menyusun SKRIPSI,  
dengan judul :

**" ANALISIS TINGKAT EFEKTIVITAS ANTARA REACT NATIVE DAN FLUTTER DALAM  
PEMBUATAN APLIKASI TO DO LIST BERBASIS MOBILE "**

Yang akan dilaksanakan dari : Tgl. **25 November 2023 s/d 29 Februri 2023**

Sehubungan dengan hal tersebut diatas, pada prinsipnya kami **menyetujui** kegiatan dimaksud  
dengan ketentuan yang tertera di belakang surat izin penelitian.

Demikian Surat Keterangan ini diberikan agar dipergunakan sebagaimana mestinya.

Diterbitkan di Makassar  
Pada Tanggal 21 November 2023

KEPALA DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU  
SATU PINTU PROVINSI SULAWESI SELATAN



**ASRUL SANI, S.H., M.Si.**  
Pangkat : PEMBINA TINGKAT I  
Nip : 19750321 200312 1 008

Tembusan Yth

1. Pembantu Dekan Bid. Akademik FT UNM Makassar di Makassar;
2. Pertinggal.

**Lampiran 1.3 Surat Keterangan Telah Melakukan Penelitian**



# KeDai Computerworks

Sekretariat : BTP BLOK B NO. 558 Telp. : 082193922372/085242749453

Email : ketawadama@gmail.com Website : kedai.or.id

## SURAT KETERANGAN TELAH MELAKUKAN PENELITIAN 004/E.B/KeDai/UNDIPA/XII/2023

Yang bertanda tangan dibawah ini :

Nama : Ryandika Wijaya  
Jabatan : Ketua Umum

Dengan ini menyatakan bahwa mahasiswa yang beridentitas:

Nama : Muh. Kidfari  
NIM : 1929041021  
Program Studi : Pendidikan Teknik Informatika dan Komputer

Telah selesai melakukan penelitian dan pengambilan data penelitian di **KeDai Computerworks** terhitung mulai tanggal 25 November 2023 s/d 09 Desember 2023 untuk memperoleh data penelitian dalam rangka penyusunan skripsi yang berjudul "**Analisis Tingkat Efektifitas antara React Native dan Flutter dalam Pembuatan Aplikasi ToDo List Berbasis Mobile**".

Demikian surat keterangan ini dibuat dan diberikan kepada yang bersangkutan untuk dipergunakan sepenuhnya.

Makassar, 09 Desember 2023

Hormat Kami,  
Badan Pengurus Harian  
**KeDai Computerworks**

Periode 2022-2023



**RYANDIKA WIJAYA**  
Ketua Umum



**DANDI SAPUTRA**  
Sekretaris Umum

# **LAMPIRAN 2**

# **HASIL PENELITIAN**

**Lampiran 2.1 Data Hasil Validasi Instrumen**



## LEMBAR VALIDASI INSTRUMENT KUESIONER

### *EVALUATION*

Judul : Analisis Tingkat Efektivitas Antara *React Native* Dan *Flutter* Dalam Pembuatan Aplikasi *To Do List* Berbasis Mobile  
Peneliti : Muhammad Kidfari  
NIM : 1929041021  
Nama Validator : Dr. Ir. Ridwan Daud Mahandc, S.Pd., M.Pd., Ipm

#### A. Tujuan

Instrumen ini bertujuan untuk mengumpulkan penilaian atau validasi terhadap instrumen *Kuesioner Evaluation* yang akan digunakan pada Analisis Tingkat Efektivitas Antara *React Native* Dan *Flutter* Dalam Pembuatan Aplikasi *To Do List* Berbasis Mobile.

#### B. Petunjuk Pengisian

1. Lembar validasi ini diisi oleh validator yang menilai instrument *Kuesioner Evaluation* dari Analisis Tingkat Efektivitas Antara *React Native* Dan *Flutter* Dalam Pembuatan Aplikasi *To Do List* Berbasis Mobile, sebelum diuji coba ke lapangan
2. Bapak/Ibu dimohon penilaianya dengan memberikan tanda (✓) pada kolom jawaban sesuai dengan hasil penilaian yang Bapak/Ibu berikan contoh:

No.	Komponen Indikator	Skala Penilaian			
		SS	S	TS	STS
1.	Alternatif jawaban jelas	✓			

3. Jika Bapak/Ibu ingin memberikan komentar atau saran, silakan mengisi isian pada bagian E. Komentar dan Saran
4. Berilah tanda (centang) pada salah satu pilihan mengenai hasil penilaian Bapak/Ibu pada bagian F.Kesimpulan.

#### C. Penilaian

- SS = Sangat Setuju  
S = Setuju  
TS = Tidak Setuju  
STS = Sangat Tidak Setuju

**D. Penilaian**

No.	Komponen Indikator	Skala Penilaian			
		SS	S	TS	STS
<b>Aspek Petunjuk</b>					
1	Tujuan instrument dinyatakan dengan jelas		✓		
2	Petunjuk pengisian instrument dinyatakan dengan jelas		✓		
3	Kriteria penilaian dinyatakan dengan jelas		✓		
<b>Aspek Isi (Materi)</b>					
4	Butir instrument dinayatakan dengan jelas	✓			
5	Alternatif jawaban jelas	✓			
6	Mencakup materi penilaian secara representatif	✓			
<b>Aspek Bahasa</b>					
7	Menggunakan Bahasa Indonesia yang baik	✓			
8	Menggunakan Bahasa Indonesia yang sederhana dan komunikatif		✓		
9	Menggunakan Bahasa Indonesia yang mudah dipahami		✓		
10	Kalimat tidak menimbulkan penafsiran ganda	✓			

**E. Komentar atau Saran**

.....  
 .....  
 .....  
 .....

**F. Kesimpulan**

Instrumen *Kuesioner Evaluation* yang akan digunakan pada Analisis Tingkat Efektivitas Antara *React Native* Dan *Flutter* Dalam Pembuatan Aplikasi *To Do List* Berbasis *Mobile* dinyatakan :

- Layak untuk uji coba lapangan
- Layak untuk uji coba lapangan dengan revisi
- Tidak layak untuk uji coba lapangan

Makassar,

7 / 11

2023

  
Dr. Ir. Ridwan Daud Mahande, S.Pd., M.Pd., Ipm  
NIP. 198408042019031005

## Lampiran 2.2 Lembar Validasi Ahli Sistem

	<b>LEMBAR VALIDASI AHLI SISTEM</b>
---	------------------------------------

Judul : Analisis Tingkat Efektivitas Antara *React Native* dan *Flutter* Dalam Pembuatan Aplikasi *To Do List* Berbasis *Mobile*

Peneliti : Muh.kidfari

NIM : 1929041021

Validator : Dr. Ir. Ridwan Daud Mahende S.Pd.,M.Pd., IPM

### A. Tujuan

Instrumen ini bertujuan untuk mengumpulkan penilaian aspek sistem terhadap pengembangan aplikasi *To-Do List* berbasis *Mobile* yang di kembangkan menggunakan *React Native* dan *Flutter* untuk di bandingkan sesuai kebutuhan Skripsi ini

### B. Petunjuk Pengisian

1. Lembar validasi ini diisi oleh validator yang menilai dari pengembangan aplikasi *To-Do List* berbasis *Mobile* yang di kembangkan menggunakan *React Native* dan *Flutter* untuk di bandingkan sesuai kebutuhan Skripsi terkait
2. Bapak/Ibu dimohon penilaiannya dengan memberikan tanda centang (✓) pada kolom jawaban sesuai dengan hasil penilaian yang Bapak/Ibu berikan

Contoh :

No.	Fitur yang diuji	Hasil yang diharapkan	Hasil	
			Ya	Tidak
1.	Registrasi	Memunculkan menu registrasi	✓	

3. Berikan tanda (=) pada jawaban yang dianggap salah pada pengisian sebelumnya (✗) kemudian berilah tanda centang baru (✓) pada jawaban yang dianggap benar

Contoh :

No.	Fitur yang diuji	Hasil yang diharapkan	Hasil	
			Ya	Tidak
1.	Registrasi	Memunculkan menu registrasi	✗	✓

4. Jika Bapak/Ibu ingin memberikan komentar atau saran, silakan mengisi isian pada bagian E. Komentar dan Saran
5. Berilah tanda centang (✓) pada salah satu pilihan mengenai hasil penilaian Bapak/Ibu pada bagian F.Kesimpulan

#### C. Penilaian

Ya = Berfungsi

TIDAK = Tidak Berfungsi

#### D. Penilaian

No.	Fitur yang Diuji	Hasil yang diharapkan	Hasil	
			Ya	Tidak
<i>Login</i>				
1	<i>Test Login</i>	Halaman menu utama terbuka sesuai dengan Akun yang di buka oleh <i>User</i>	✓	
<i>To-Do List</i>				
2	<i>Add To-Do</i>	<i>To-Do</i> yang telah di <i>update</i> dapat disimpan pada database dan menampilkan halaman view <i>To-Do</i>	✓	

3	<i>Test Check and Undo</i>	1. <i>To-Do</i> yang Telah di Buat dapat di tandai Jika telah selesai belum selesai 2. <i>To-Do</i> yang telah di tandai selesai dapat di <i>undo check</i> bila kegiatan belum selesai sepenuhnya	<input checked="" type="checkbox"/>	
4	<i>Test Edit To-Do</i>	User dapat Melakukan <i>Update</i> Pada <i>To-Do</i> yang Sebelumnya telah di Buat	<input checked="" type="checkbox"/>	
<i>Opsi Exit &amp; Navigation Button</i>				
5	<i>Navigation Button</i>	Halaman berpindah ketika menekan Button Di bagian bawah aplikasi sesuai halaman yang di tekan	<input checked="" type="checkbox"/>	
6	<i>Exit</i>	Aplikasi dapat menutup sesuai instruksi pengguna	<input checked="" type="checkbox"/>	

#### E. Komentar atau Saran

Layak.....  
 .....  
 .....

#### F. Kesimpulan

Berilah tanda centang (✓) pada hasil penilaian Bapak/Ibu

- Layak untuk uji coba lapangan
- Layak untuk uji coba lapangan dengan revisi
- Tidak layak untuk uji coba lapangan

Makassar,

2023



Dr. Ir. Ridwan Daud Mahande, S.Pd., M.Pd., Ipm  
NIP. 198408042019031005

### Lampiran 2.3 Data Hasil Observasi Aspek Fungsionalitas



#### **LEMBAR OBSERVASI RESPONDEN**

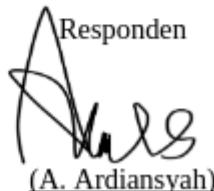
#### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : A. Ardiansyah

Pekerjaan : Digital Marketing

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar, 08/Desember/2023

Responden  
  
(A. Ardiansyah)



## **LEMBAR OBSERVASI RESPONDEN**

### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : Ade Rahmat Maulana

Pekerjaan : Web Developer

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar, 08/Desember/2023

Responden

(Ade Rahmat Maulana)



### **LEMBAR OBSERVASI RESPONDEN**

#### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : Dandi Saputra

Pekerjaan : -

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar, 08/Desember/2023

Responden  
  
(Dandi Saputra)



### **LEMBAR OBSERVASI RESPONDEN**

#### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : Dicky darmawan, S.Kom

Pekerjaan : Fullstack Developer

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar, 08/Desember/2023

Responden

  
(Dicky darmawan, S.Kom)



## **LEMBAR OBSERVASI RESPONDEN**

### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : Maulidani Mahmud

Pekerjaan : Mobile Developer

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar 08/Desember/2023

Responden

(Maulidani Mahmud)



### **LEMBAR OBSERVASI RESPONDEN**

#### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : Rahmat Hidayat

Pekerjaan : Mobile Developer

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar, 08/Desember/2023

Responden

A handwritten signature in black ink, appearing to read "Rahmat Hidayat".

(Rahmat Hidayat)



## **LEMBAR OBSERVASI RESPONDEN**

### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : Rama ismaya Sofian, S.Kom

Pekerjaan : Product Enginer

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar, 08/Desember/2023

Responden

(Rama ismaya Sofian, S.Kom)



## **LEMBAR OBSERVASI RESPONDEN**

### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : Yeheskiel P R

Pekerjaan : Web Developer

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar, 08/Desember/2023

Responden

(Yeheskiel P R)



### **LEMBAR OBSERVASI RESPONDEN**

#### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : Ricky Heri Sappa

Pekerjaan : -

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar, 08/Desember/2023

Responden

A handwritten signature in black ink, appearing to read "Ricky Heri Sappa".

(Ricky Heri Sappa)



### **LEMBAR OBSERVASI RESPONDEN**

#### **LEMBAR OBSERVASI ASPEK FUNCTIONALITY**

Nama : Muh. Muhammin Nur

Pekerjaan : -

No	Nama Task	Status	
		Berhasil	Tidak Berhasil
1	<i>Task Login</i>	✓	
2	<i>Task Add To-Do</i>	✓	
3	<i>Task Check To-Do</i>	✓	
4	<i>Task Undo Check</i>	✓	
5	<i>Task Edit To-Do</i>	✓	
6	<i>Task Delete</i>	✓	
7	<i>Task Notification</i>	✓	
8	<i>Task Navigation bar</i>	✓	
9	<i>Task Logout</i>	✓	

Makassar, 08/Desember/2023

Responden

(Muh. Muhammin Nur)

## Lampiran 2.4 Data Hasil Lembar Persetujuan Responden

	<b><u>LEMBAR PERSETUJUAN RESPONDEN</u></b>
---	--

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : A. Ardiansyah  
Umur : 23 tahun  
Pekerjaan : Digital Marketing  
Alamat : Perdos Unhas  
Nomor Hp : 0821-9319-0606

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI  
Nim : 192941021  
Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS *MOBILE* ”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

Makassar, 08 Desember 2023



A. Ardiansyah



### **LEMBAR PERSETUJUAN RESPONDEN**

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : Ade Rahmat Maulana

Umur : 23 tahun

Pekerjaan : Web Developer

Alamat : Jl.Cendrawasih

Nomor Hp : 0853-9742-5303

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI

Nim : 192941021

Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS *MOBILE* ”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

Makassar, 08 Desember 2023

Ade Rahmat Maulana



### **LEMBAR PERSETUJUAN RESPONDEN**

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : Dandi Saputra

Umur : 21 tahun

Pekerjaan : -

Alamat : Minasa Upa

Nomor Hp : 0852-4274-9453

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI

Nim : 192941021

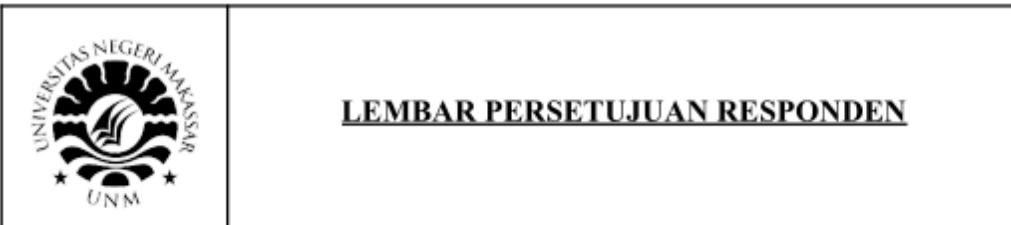
Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba

Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS *MOBILE*”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

Makassar, 08 Desember 2023

Dandi Saputra



### **LEMBAR PERSETUJUAN RESPONDEN**

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : Dicky darmawan, S.Kom

Umur : 22 tahun

Pekerjaan : Fullstack Developer

Alamat : Btn. Asal Mula

Nomor Hp : 0813-5583-4769

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI

Nim : 192941021

Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS *MOBILE* ”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

Makassar, 08 Desember 2023

A handwritten signature in black ink, appearing to read "Dicky darmawan, S.Kom".

Dicky darmawan, S.Kom



### **LEMBAR PERSETUJUAN RESPONDEN**

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : Maulidani Mahmud

Umur : 22 tahun

Pekerjaan : Mobile Developer

Alamat : Perumnas Sudiang

Nomor Hp : 0853-4212-8158

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI

Nim : 192941021

Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS *MOBILE* ”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

Makassar, 08 Desember 2023

A handwritten signature in black ink, appearing to read "Maulidani Mahmud", is placed below the date.

Maulidani Mahmud



### **LEMBAR PERSETUJUAN RESPONDEN**

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : Rahmat Hidayat  
Umur : 23 tahun  
Pekerjaan : Mobile Developer  
Alamat : Btn. Asal Mula  
Nomor Hp : 0821-9469-2848

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI  
Nim : 192941021  
Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS *MOBILE* ”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

Makassar, 08 Desember 2023

Rahmat Hidayat



## **LEMBAR PERSETUJUAN RESPONDEN**

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : Rama ismaya Sofian, S.Kom

Umur : 23 tahun

Pekerjaan : Product Enginer

Alamat : Btn. Graha Mustika

Nomor Hp : 0822-9225-6219

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI

Nim : 192941021

Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS *MOBILE*”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

Makassar, 08 Desember 2023

Rama ismaya Sofian, S.Kom



### **LEMBAR PERSETUJUAN RESPONDEN**

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : Yeheskiel P R

Umur : 23 tahun

Pekerjaan : Web Developer

Alamat : Jl. Inpeksi PAM nipa-nipa

Nomor Hp : 0896-6958-2305

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI

Nim : 192941021

Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba  
Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS *MOBILE* ”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

Makassar, 08 Desember 2023



Yeheskiel P R



## **LEMBAR PERSETUJUAN RESPONDEN**

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : Ricky Heri Sappa

Umur : 22

Pekerjaan : -

Alamat : Sulawesi Selatan, Makassar, Kec. Rappocini, Kel. Mappala,  
Jl. Tidung VI, Stp 3 No.53

Nomor Hp : 08875366094

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI

Nim : 192941021

Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba  
Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS *MOBILE*”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

Makassar, November 2023

A handwritten signature in black ink, appearing to read "Ricky Heri Sappa".

Ricky Heri Sappa

	<b><u>LEMBAR PERSETUJUAN RESPONDEN</u></b>
---	--

Dengan surat ini saya yang bertanda tangan di bawah ini :

Nama : Muh. Muhammin Nur

Umur : 22

Pekerjaan : Mahasiswa

Alamat : Jalan Pendidikan

Nomor Hp : 081342400303

Telah menanggapi surat permohonan dan menyetujui menjadi responden dalam penelitian yang dilakukan oleh :

Nama : MUHAMMAD KIDFARI

Nim : 192941021

Alamat : BTN Pao-pao Blok F2 No.5, Paccinonggang, Kec. Somba

Opu, Kabupaten Gowa, Sulawesi Selatan

Dengan judul “ANALISIS TINGKAT EFEKTIVITAS ANTARA *REACT NATIVE* DAN *FLUTTER* DALAM PEMBUATAN APLIKASI *TO DO LIST* BERBASIS MOBILE”. Demikian surat persetujuan ini disepakati tanpa ada paksaan dari pihak manapun.

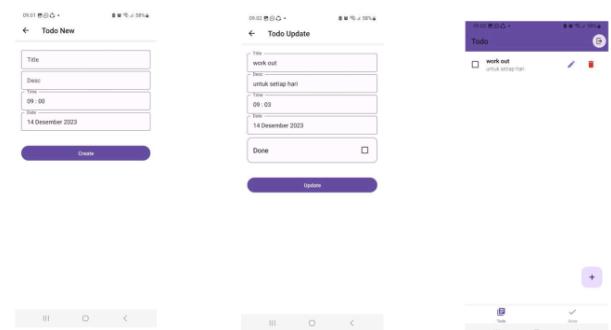
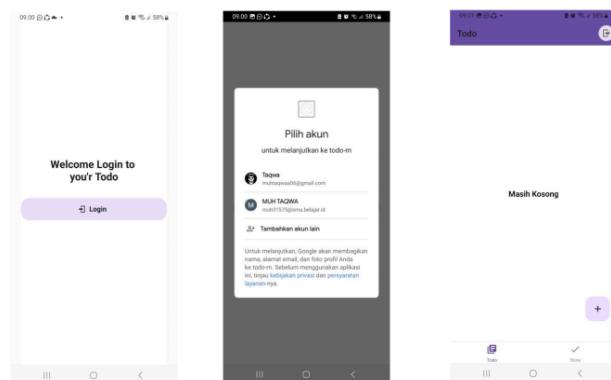
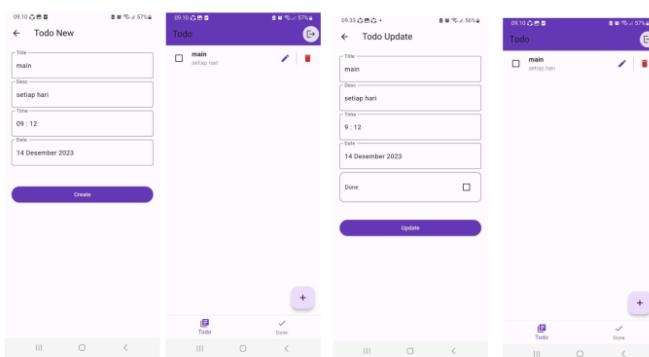
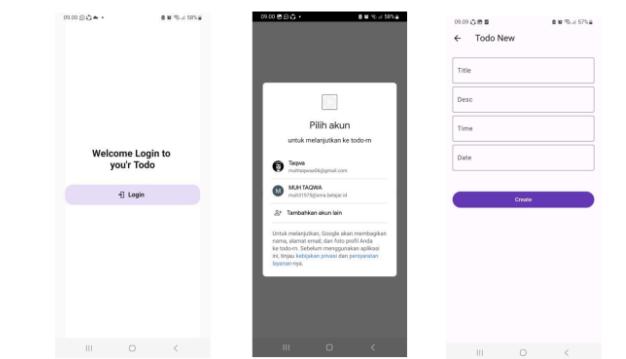
Makassar, 30 November 2023



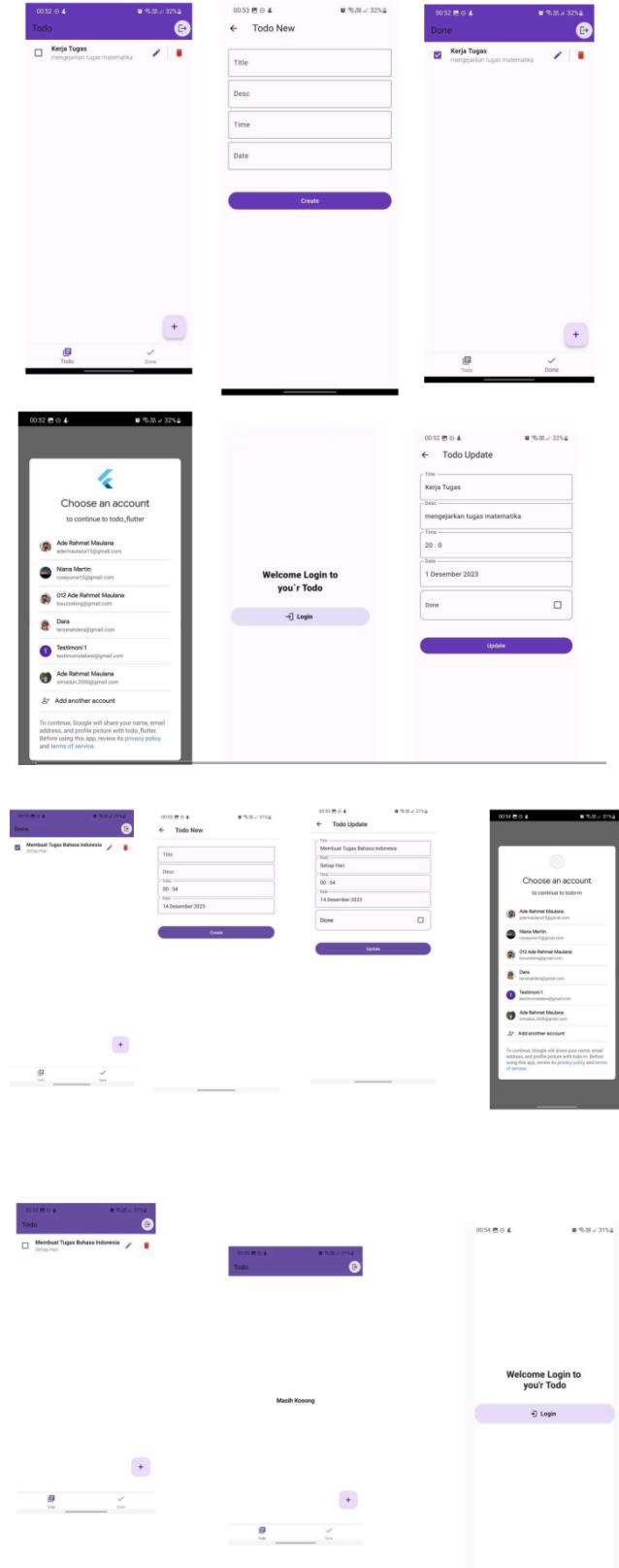
Muh. Muhammin Nur

## Lampiran 2.5 Data Hasil Screenshot Aplikasi *To-do List* Responden

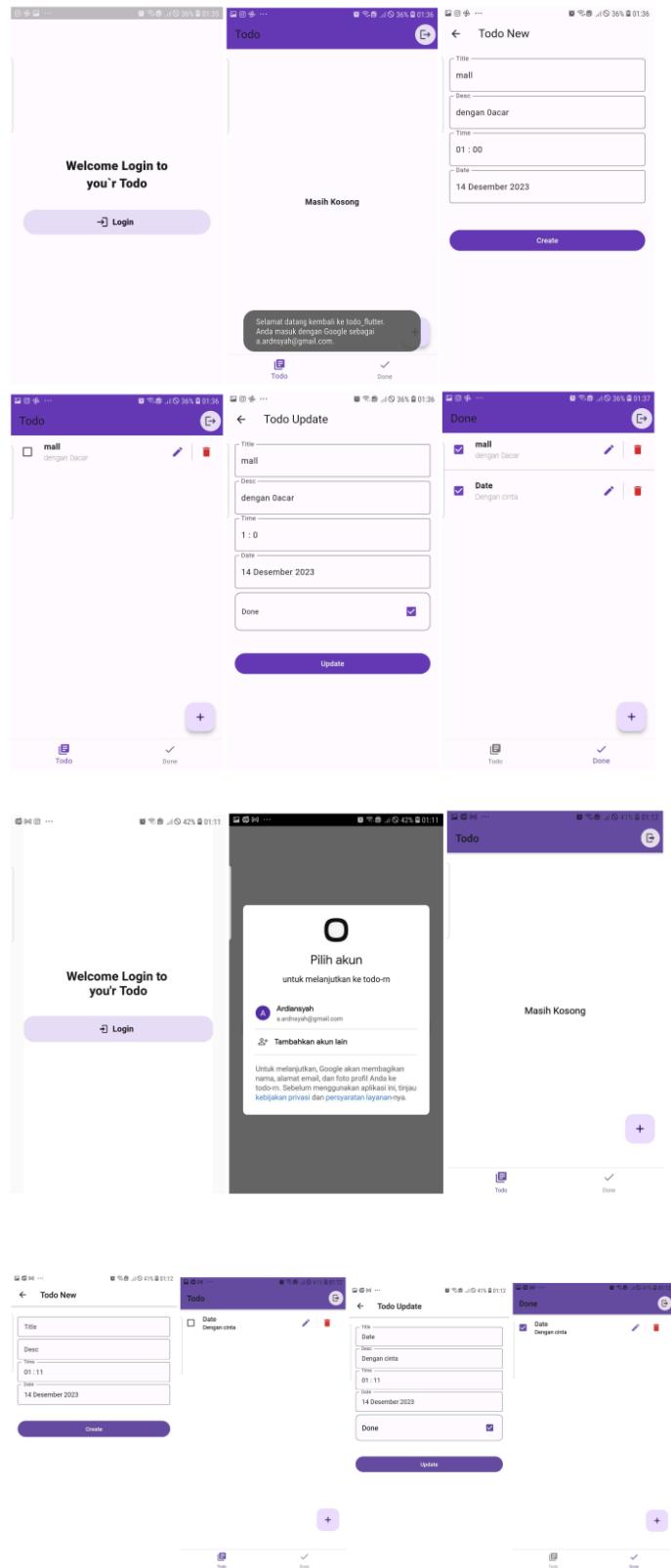
### 1. Maulidani Mahmud



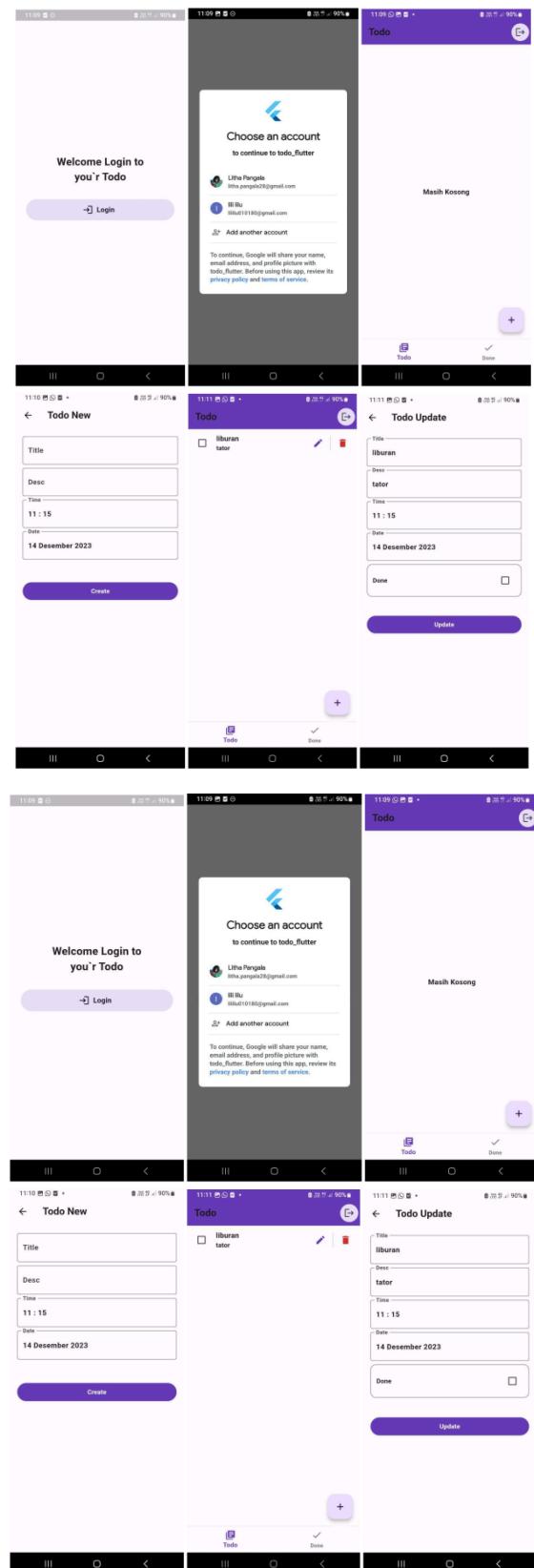
## 2. Ade Rahmat Maulana



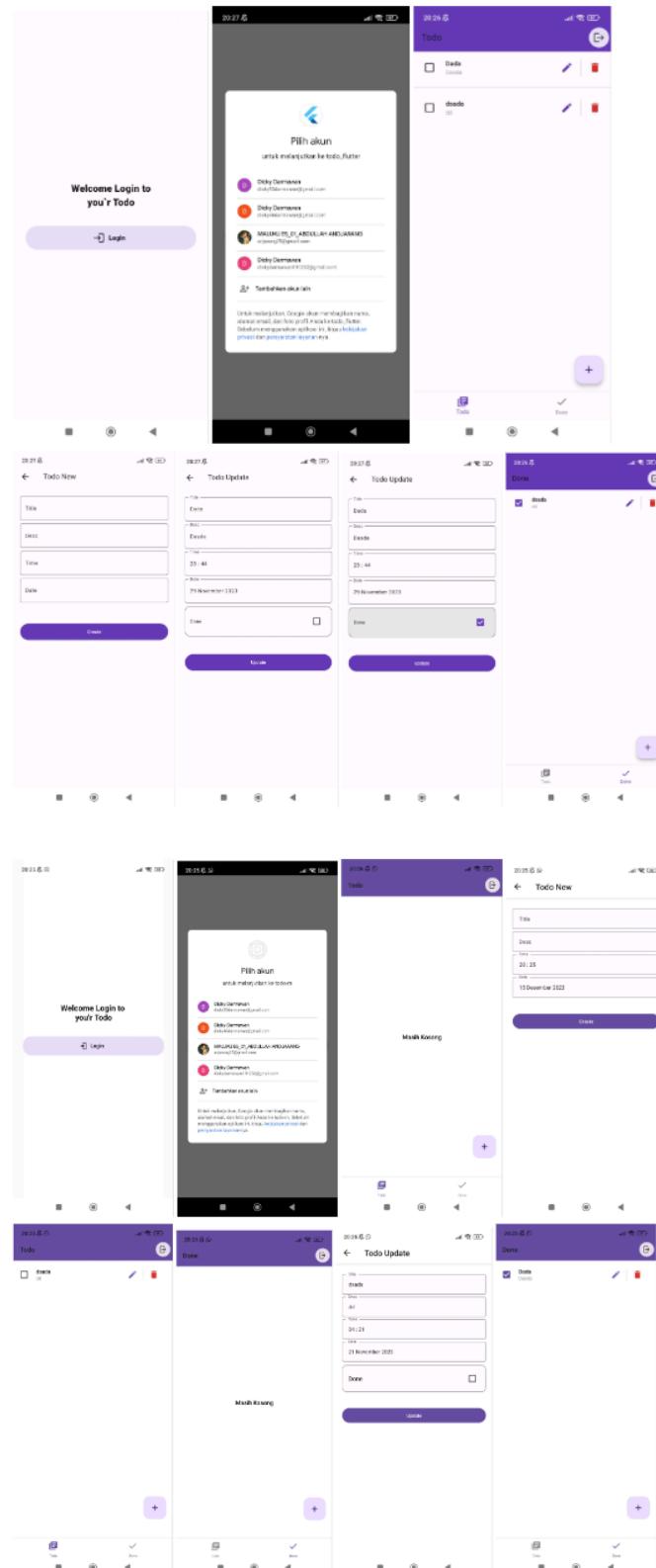
### 3. A. Ardiansyah, S.Kom



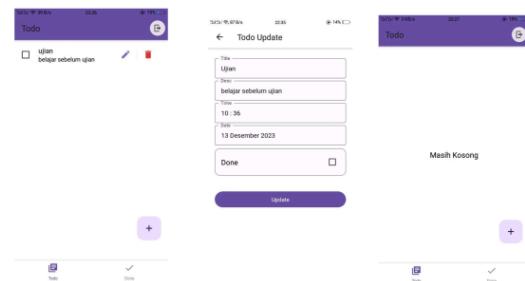
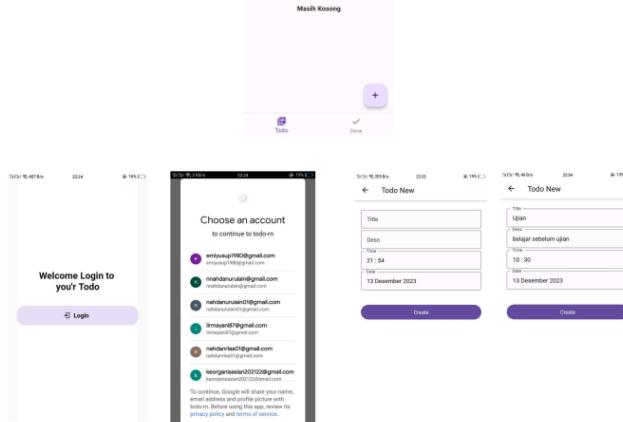
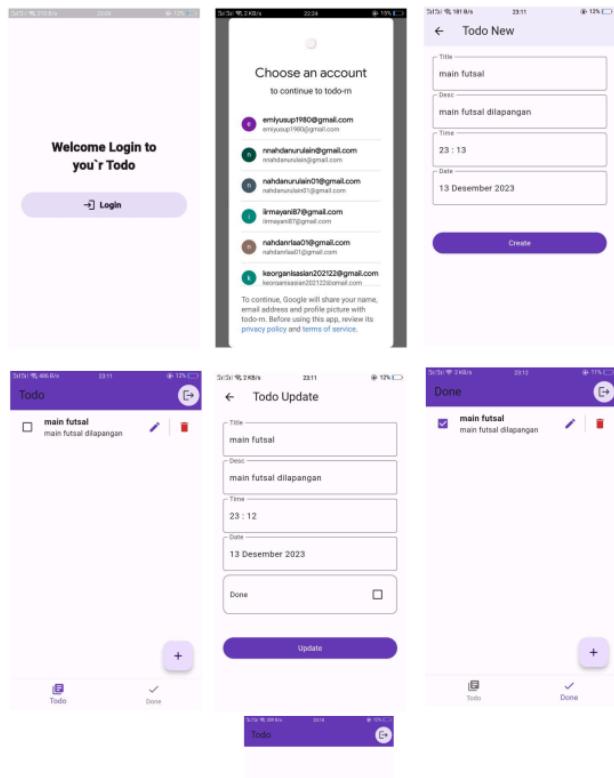
#### 4. Dandi Saputra



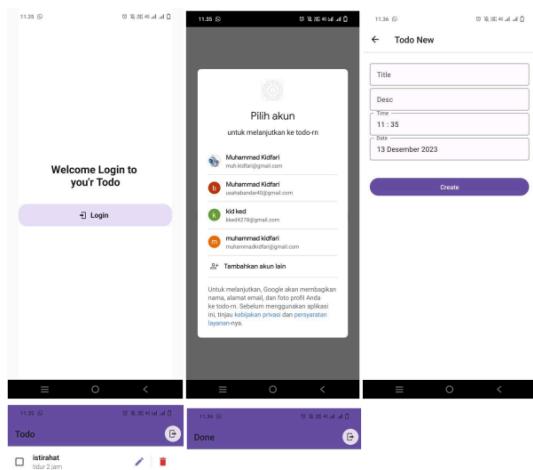
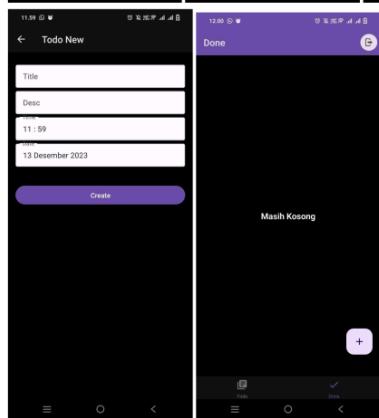
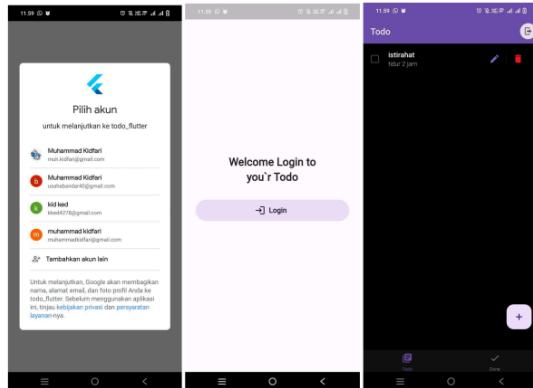
## 5. Dicki Darmawan, S.Kom



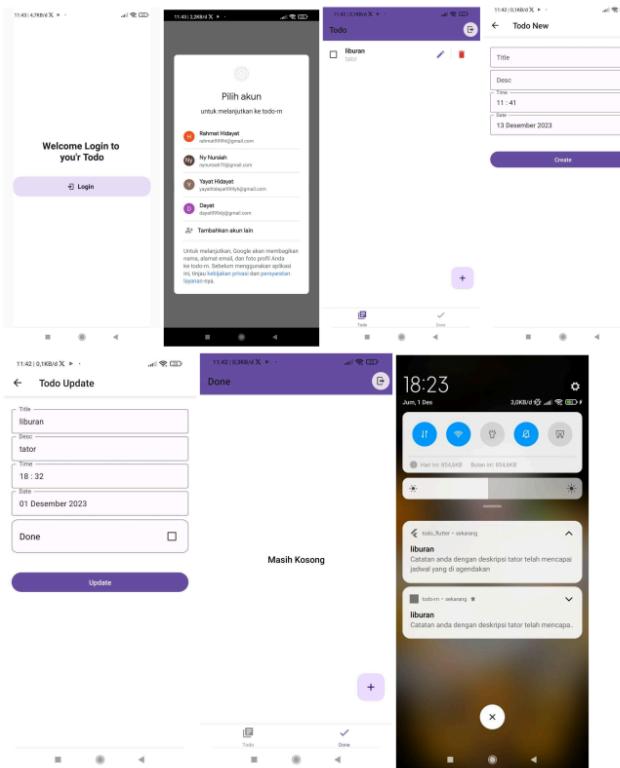
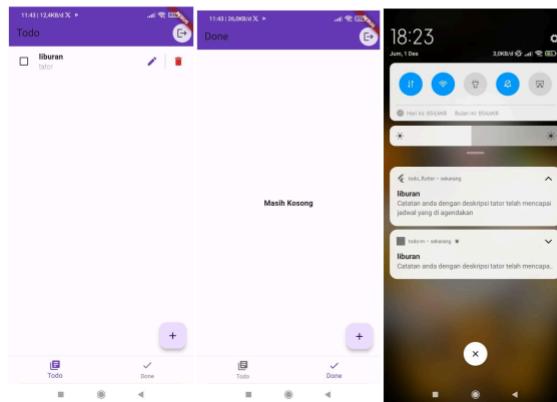
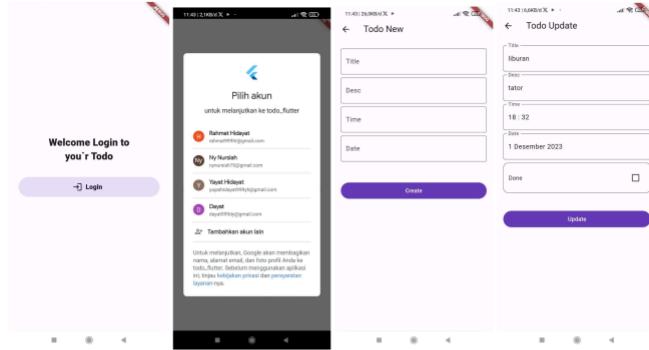
## 6. Yeheskiel P R



## 7. Rama Ismaya Sopian, S.Kom



## 8. Rahmat Hidayat



## 9. Muh. Muhammin Nur, S.Kom

The image consists of two side-by-side screenshots of a mobile application interface, likely a todo list application.

**Top Left Screenshot:** Shows a "Todo Update" screen at 12:15 AM. The form fields are:

- Title: Belanja
- Desc: Ke pasar
- Time: 08:00
- Date: 27 November 2023

A "Done" button with a checkmark icon is present. Below the form is a purple "Update" button.

**Bottom Left Screenshot:** Shows a list of completed tasks (checked) with edit and delete icons. The tasks are:

- Kampus
- Belanja
- Olahraga
- Membersihkan
- Belajar
- Kampus
- Nonton
- Olahraga
- Tugas
- Olahraga
- Olahraga

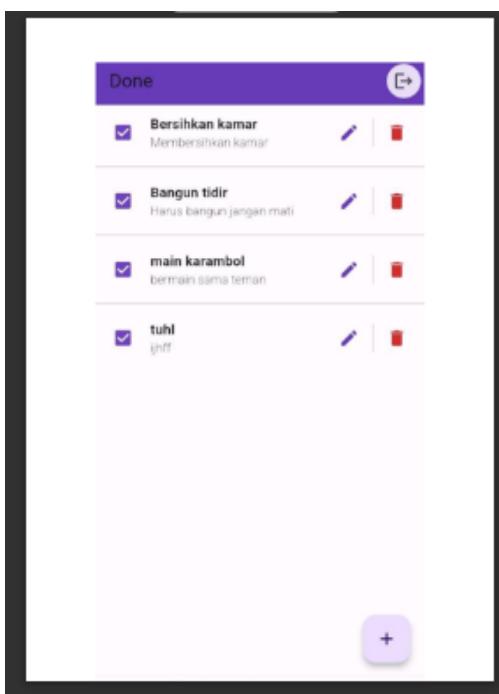
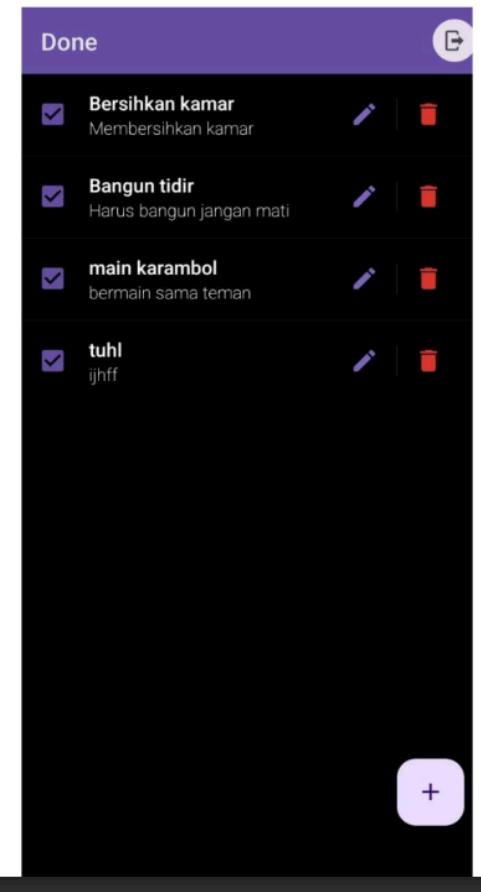
**Top Right Screenshot:** Shows a "Todo Update" screen at 10:37 PM. The form fields are:

- Title: Olahraga
- Desc: Workout
- Time: 6:00
- Date: 2 Desember 2023

A "Done" button with a checkmark icon is present. Below the form is a purple "Update" button.

**Bottom Right Screenshot:** Shows a list of completed tasks (checked) with edit and delete icons. The tasks are identical to the ones in the bottom left screenshot.

## 10. Ricki Hery Sappa



## Lampiran 2.6 Kode Program dari Task *To-do List Flutter*

No	Fungsi	Kode Program
1	Login	<pre> FilledButton.tonalIcon(     style: FilledButton.styleFrom(         fixedSize: Size(size.width, 48),     ),     onPressed: () =&gt; signInWithGoogle(context),     label: const Text("Login",         style: TextStyle(             fontSize: 16,             fontWeight: FontWeight.w900,         ),         icon: const Icon(Icons.login),     ), )  Future signInWithGoogle(BuildContext context) async {     // Trigger the authentication flow     final GoogleSignInAccount? googleUser = await     GoogleSignIn().signIn();      // Obtain the auth details from the request     final GoogleSignInAuthentication? googleAuth =         await googleUser?.authentication;      // Create a new credential     final credential = GoogleAuthProvider.credential(         accessToken: googleAuth?.accessToken,         idToken: googleAuth?.idToken,     );      // Once signed in, return the UserCredential     await     FirebaseAuth.instance.signInWithCredential(credential).then((auth) {         if(auth.user != null) {             PageNavigationOpacity.toPageReplacement(context, const RootNavBarPage());         }     }); } </pre>

No	Fungsi	Kode Program
		<pre>     }  });  return; } </pre>
2	Add To-Do	<pre> ElevatedButton(     style: ElevatedButton.styleFrom(         fixedSize: Size(size.width, 40),         backgroundColor: Colors.deepPurple,     ),     onPressed: onSubmit,     child: Text(widget.todo != null ? "Update" : "Create",         style: const TextStyle(color: Colors.white),     ), ), onSubmit() async {     TimeOfDay jam = TimeOfDay(         hour: int.parse(time.text.split(':')[0]),         minute: int.parse(time.text.split(':')[1]),     ); }  DateFormat inputFormat = DateFormat("dd MMMM yyyy", "id_ID"); DateTime tanggal = inputFormat.parse(date.text);  DateTime initDateTime = DateTime(     tanggal.year,     tanggal.month,     tanggal.day,     jam.hour,     jam.minute, ); Timestamp dateTime = Timestamp.fromDate(initDateTime);  final user = FirebaseAuth.instance.currentUser; final data = TodoModel(     id: widget.todo?.id ?? "_", </pre>

No	Fungsi	Kode Program
		<pre> userId: user?.uid ?? "-", isDone: isDone, title: title.text, descripsion: descripsion.text, dateTime: dateTime);  log(data.toMap().toString());  final isError = descripsion.text.isEmpty    title.text.isEmpty;  if(title.text.isEmpty) {     isErrorTitle = true; }  if(descripsion.text.isEmpty) {     isErrorDescripsion = true; }  if(isError) {     setState(() {});     return; }  if(widget.todo != null) {     await TodoFirestoreService.updateTodo(data).then(         () =&gt; Navigator.pop(context),     );     return; }  await TodoFirestoreService.createTodo(data).then(     () =&gt; Navigator.pop(context), ); } </pre>
3	Check To-Do	<p>Checkbox(</p> <p>    value: todo.isDone,</p> <p>    onChanged: (v) {</p> <p>        TodoFirestoreService.updateTodo(</p>

No	Fungsi	Kode Program
		<pre>todo.copyWith(isDone: v ?? todo.isDone),     ); }, ), static Future&lt;void&gt; updateTodo(TodoModel todo){ return todos.doc(todo.id).update(todo.toMap()); }</pre>
4	<i>Undo Check</i>	<pre>Checkbox(   value: todo.isDone,   onChanged: (v) {     TodoFirestoreService.updateTodo(       todo.copyWith(isDone: v ?? todo.isDone),     ); }, ), static Future&lt;void&gt; updateTodo(TodoModel todo){ return todos.doc(todo.id).update(todo.toMap()); }</pre>
5	<i>Edit To-Do</i>	<pre>ElevatedButton(   style: ElevatedButton.styleFrom(     fixedSize: Size(size.width, 40),     backgroundColor: Colors.deepPurple,   ),   onPressed: onSubmit,   child: Text(widget.todo != null ? "Update" : "Create",   style: const TextStyle(color: Colors.white),   ), ), onSubmit() async {   TimeOfDay jam = TimeOfDay(     hour: int.parse(time.text.split(':')[0]),     minute: int.parse(time.text.split(':')[1]),   );    DateFormat inputFormat = DateFormat("dd MMMM yyyy",   "id_ID");   DateTime tanggal = inputFormat.parse(date.text); }</pre>

No	Fungsi	Kode Program
		<pre> DateTime initDateTime = DateTime(     tanggal.year,     tanggal.month,     tanggal.day,     jam.hour,     jam.minute, ); Timestamp dateTime = Timestamp.fromDate(initDateTime);  final user = FirebaseAuth.instance.currentUser; final data = TodoModel(     id: widget.todo?.id ?? "_",     userId: user?.uid ?? "-",     isDone: isDone,     title: title.text,     descripsion: descripsion.text,     dateTime: dateTime);  log(data.toMap().toString());  final isError = descripsion.text.isEmpty    title.text.isEmpty;  if(title.text.isEmpty) {     isErrorTitle = true; }  if(descripsion.text.isEmpty) {     isErrorDescripsion = true; }  if(isError) {     setState(() {});     return; }  if(widget.todo != null) { </pre>

No	Fungsi	Kode Program
		<pre> await TodoFirestoreService.updateTodo(data).then(     () =&gt; Navigator.pop(context), ); </pre>
6	Delete	<pre> SizedBox(     height: 32,     width: 32,     child: IconButton(         padding: const EdgeInsets.all(0.0),         color: Colors.red.shade700,         splashRadius: 24,         splashColor: Colors.red.shade700,         icon: const Icon(Icons.delete, ),         onPressed: onDelete,     ), ), ), onDelete() {     TodoFirestoreService.deleteTodo(todo.id); } </pre>
7	Notification	<pre> Future&lt;void&gt; firebaseMessagingBackgroundHandler(RemoteMessage message) async { await Firebase.initializeApp(     options: DefaultFirebaseOptions.currentPlatform, ); }  AndroidNotificationChannel? channel;  FlutterLocalNotificationsPlugin? flutterLocalNotificationsPlugin; // late FirebaseMessaging messaging;  final GlobalKey&lt;NavigatorState&gt; navigatorKey = GlobalKey&lt;NavigatorState&gt;();  void notificationTapBackground(NotificationResponse notificationResponse) { </pre>

No	Fungsi	Kode Program
		<pre> log('notification(\${notificationResponse.id}) action tapped: '   +'\${notificationResponse.actionId} with'   +' payload: \${notificationResponse.payload}'); if(notificationResponse.input?.isEmpty ?? false) {   log('notification action tapped with input: \${notificationResponse.input}'); } }  void showCustomSnackBar(String title, String body) { BuildContext context = navigatorKey.currentContext!; ScaffoldMessenger.of(context).showSnackBar( SnackBar( content: Column( crossAxisAlignment: CrossAxisAlignment.start, mainAxisSize: MainAxisSize.min, children: [ Text( title, style: const TextStyle( fontWeight: FontWeight.bold, ), ), const SizedBox(height: 4.0), Text(body), ], ), ), duration: const Duration(seconds: 3), // Adjust the duration as needed // behavior: SnackBarBehavior, // Customize the background color ), ); }  void main() async { WidgetsFlutterBinding.ensureInitialized(); } </pre>

No	Fungsi	Kode Program
		<pre> await Firebase.initializeApp(   options: DefaultFirebaseOptions.currentPlatform, );  final user = FirebaseAuth.instance.currentUser; FirebaseMessaging messaging = FirebaseMessaging.instance;  await messaging.requestPermission(   alert: true,   announcement: false,   badge: true,   carPlay: false,   criticalAlert: false,   provisional: false,   sound: true, );  if(user != null) {   await messaging.subscribeToTopic(user.uid); }  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler); FirebaseMessaging.onMessage.listen((message) {   log('Got a message whilst in the foreground!');   log('Message data: \${message.data}');    if(message.notification != null) {     log('Message also contained a notification: \${message.notification?.body ?? "kosong"}');     String title = message.notification?.title ?? "Notif";     String body = message.notification?.body ?? "kosong";     // menampilkan alert notifikasi ketika aplikasi sedang terbuka     showCustomSnackBar(       title,       body,     );   } }) </pre>

No	Fungsi	Kode Program
		<pre>     ); } });  if(!kIsWeb) {   channel = const AndroidNotificationChannel(     'flutter_notification', // id     'flutter_notification_title', // title     importance: Importance.high,     enableLights: true,     enableVibration: true,     showBadge: true,     playSound: true); }  flutterLocalNotificationsPlugin = FlutterLocalNotificationsPlugin();  const android =   AndroidInitializationSettings('@drawable/ic_notifications_icon'); const iOS = DarwinInitializationSettings(); const initSettings = InitializationSettings(android: android, iOS: iOS);  await flutterLocalNotificationsPlugin!.initialize(initSettings,   onDidReceiveNotificationResponse: notificationTapBackground,   onDidReceiveBackgroundNotificationResponse:   notificationTapBackground);  await messaging.setForegroundNotificationPresentationOptions(   alert: true,   badge: true,   sound: true, ); }  await initializeDateFormatting('id_ID', null).then(   (_) =&gt; runApp(App(     isAuth: user != null,   ) )); </pre>

No	Fungsi	Kode Program
		<pre>     navigatorKey: navigatorKey,   )), ); } </pre>
8	Navigation bar	<pre> bottomNavigationBar: BottomNavigationBar(   items: const &lt;BottomNavigationBarItem&gt;[     BottomNavigationBarItem(       icon: Icon(Icons.library_books_roud),       label: 'Todo',     ),     BottomNavigationBarItem(       icon: Icon(Icons.check_rounded),       label: 'Done',     ),   ],   currentIndex: _selectedIndex,   onTap: _onItemTapped, ), void _onItemTapped(int index) {   setState(() {     _selectedIndex = index;   }); } </pre>
9	Logout	<pre> IconButton.filledTonal(   onPressed: () =&gt; signOut(context),   icon: const Icon(Icons.logout), ) Future signOut(BuildContext context) async {   // Trigger the authentication flow   final auth = FirebaseAuth.instance;    await GoogleSignIn().signOut();   await auth.signOut().then((_) =&gt;     PageNavigationOpacity.     toPageReplacement(       context, </pre>

No	Fungsi	Kode Program
		<pre>const LoginPage(),     ); return; }</pre>

### Lampiran 2.7 Kode Program dari Task *To-do List React Native*

No	Fungsi	Kode Program
1	<i>Login</i>	<pre> async function onGoogleButtonPress() {     // Check if your device supports Google Play     await GoogleSignin.hasPlayServices({         showPlayServicesUpdateDialog: true });     // Get the users ID token     const { tokenId } = await GoogleSignin.signIn();      // Create a Google credential with the token     const googleCredential =         auth.GoogleAuthProvider.credential(tokenId);      // Sign-in the user with the credential     const userCredensial = await         auth().signInWithCredential(googleCredential);     setUser(userCredensial.user); } </pre>
2	<i>Add To-Do</i>	<pre> &lt;Button     mode="contained"     style={{         marginTop: 40,     }}     onPress={onSubmit} &gt;     {isNew ? "Create" : "Update"} &lt;/Button&gt;  const onSubmit = async () =&gt; {     const newDate = new Date(         date.getFullYear(),         date.getMonth(),         date.getDate(),         time.getHours(),         time.getMinutes(),         time.getSeconds() ) </pre>

No	Fungsi	Kode Program
		<pre> ); const dateTIme = firebase.firestore.Timestamp.fromDate(newDate);  const todo: ITodo = {   id: <i>data</i>.id,   title: title,   descripsion: descripsion,   dateTIme: dateTIme,   userId: user?.uid ?? "-",   isDone: isDone, }; if(   todo.title === defaultData.title      todo.descripsion === defaultData.descripsion ) {   if(todo.title === defaultData.title) {     setIsError((v) =&gt; ({       ...v,       title: true,     }));   }   if(todo.descripsion === defaultData.descripsion) {     setIsError((v) =&gt; ({       ...v,       descripsion: true,     }));   }   return; } else {   setIsError((v) =&gt; ({     title: false,     descripsion: false,   }));   setLoading(true);   if(<i>data</i>.id === "New") {     await addTodo(todo);   } } </pre>

No	Fungsi	Kode Program
		<pre> } else {     await updateTodo(todo); } onToggleSnackBar(); setData(todo);  setLoading(false); router.back(); } }; </pre>
3	<i>Check To-Do</i>	<pre> &lt;Checkbox     onPress={() =&gt; {         checkedTodo(todo);     }}     status={todo.isDone ? "checked": "unchecked"} /&gt; const checkedTodo = async (item: ITodo) =&gt; {     const newData = {         ...item,         isDone: !item.isDone,     };     await updateTodo(newData);     setData((data) =&gt;         data.map((dataItem) =&gt; {             if(dataItem === item) {                 return newData;             }             return dataItem;         })     );     list.current?.prepareForLayoutAnimationRender();      LayoutAnimation.configureNext(LayoutAnimation.Presets.easeInEaseOut); }; </pre>
4	<i>Undo Check</i>	<Checkbox

No	Fungsi	Kode Program
		<pre> onPress={() =&gt; {     checkedTodo(todo); }} status={todo.isDone ? "checked": "unchecked"} /&gt; const checkedTodo = async (item: ITodo) =&gt; {     const newData = {         ...item,         isDone: !item.isDone,     };     await updateTodo(newData);     setData((data) =&gt;         data.map((dataItem) =&gt; {             if(dataItem === item) {                 return newData;             }             return dataItem;         })     );     list.current?.prepareForLayoutAnimationRender();      LayoutAnimation.configureNext(LayoutAnimation.Presets.easeInEase Out); }; </pre>
5	Edit To-Do	<pre> &lt;Button     mode="contained"     style={{{         marginTop: 40,     }}}     onPress={onSubmit} &gt;     {isNew ? "Create" : "Update"} &lt;/Button&gt;  const onSubmit = async () =&gt; {     const newDate = new Date( </pre>

No	Fungsi	Kode Program
		<pre> date.getFullYear(), date.getMonth(), date.getDate(), time.getHours(), time.getMinutes(), time.getSeconds() );  const dateTime = firebase.firestore.Timestamp.fromDate(newDate);  const todo: ITodo = {   id: <i>data</i>.id,   title: title,   descripsion: descripsion,   dateTime: dateTime,   userId: user?.uid ?? "-",   isDone: isDone, }; if(   todo.title === defaultData.title      todo.descripsion === defaultData.descripsion ) {   if(todo.title === defaultData.title) {     setIsError((v)=&gt; ({       ...v,       title: true,     }));   }   if(todo.descripsion === defaultData.descripsion) {     setIsError((v)=&gt; ({       ...v,       descripsion: true,     }));   } } return; } else {   setIsError((v)=&gt; ({ </pre>

No	Fungsi	Kode Program
		<pre>         title: false,         descpision: false,       });       setLoading(true);       if(data.id === "New") {         await addTodo(todo);       } else {         await updateTodo(todo);       }       onToggleSnackBar();       setData(todo);       setLoading(false);       router.back();     }   }; </pre>
6	<i>Delete</i>	<pre> &lt;IconButton   iconColor={MD3Colors.error50}   icon="delete"   onPress={() =&gt; {     removeTodo(todo);   }} /&gt; const removeTodo = async (item: ITodo) =&gt; {   await deleteTodo(item);   setData((data) =&gt;     data.filter((dataItem) =&gt; {       return dataItem !== item;     })   );   list.current?.prepareForLayoutAnimationRender();    LayoutAnimation.configureNext(LayoutAnimation.Presets.easeInEaseOut); }; </pre>
7	<i>Notification</i>	<pre> messaging().setBackgroundMessageHandler(async (remoteMessage) =&gt; { </pre>

No	Fungsi	Kode Program
		<pre>console.log("Message handled in the background!", remoteMessage); });  function requestAndroidPermission() {    PermissionsAndroid.request(PermissionsAndroid.PERMISSIONS.POST_NOTIFICATIONS); }  async function requestUserPermission() {   const authorizationStatus = await messaging().requestPermission();    if(authorizationStatus) {     console.log("Permission status:", authorizationStatus.toString());   } }</pre>
8	<i>Navigation bar</i>	<pre>&lt;Tabs.Screen   name="index"   options={{     title: "Todo",     tabBarIcon: ({ color })=&gt;(       &lt;MaterialIcons name="library-books" size={24} color={color}&gt;         /&gt;       ),     })   /&gt; &lt;Tabs.Screen   name="done"   options={{     title: "Done",     tabBarIcon: ({ color })=&gt;(       &lt;MaterialIcons name="check" size={24} color={color}&gt; /&gt;     ),   })   /&gt;</pre>
9	<i>Logout</i>	async function onLogout() {

No	Fungsi	Kode Program
		<pre>// Check if your device supports Google Play await GoogleSignin.hasPlayServices({     showPlayServicesUpdateDialog: true });  // Get the users ID token await GoogleSignin.signOut(); await auth().signOut(); await auth().currentUser?.reload(); setUser(null); }</pre>

### Lampiran 2.8 Rekapitulasi Pengujian *Usability Framework Flutter*

No	Pertanyaan	Item Jawa ban	Responden ke-										Tota 1 Nilai	Skor Mak s	Pers en (100 %)	Bobot Pertanya an
			1	2	3	4	5	6	7	8	9	10				
1	Saya menganggap tampilan aplikasi mudah dikenali	LR	0	1	1	1	1	1	1	1	1	1	9	10	90	77,5
		EF	0	1	1	1	1	1	1	1	1	1	9	10	90	
		MR	1	1	0	1	0	0	1	1	1	0	6	10	60	
		SF	1	1	1	1	0	0	1	1	1	0	7	10	70	
2	Saya merasa aplikasi ini mudah dioperasikan	LR	0	1	1	1	1	0	1	1	1	1	8	10	80	72,5
		EF	1	1	1	1	0	1	1	1	1	1	9	10	90	
		MR	1	1	1	1	1	0	1	0	0	0	6	10	60	
		SF	1	1	0	0	0	1	1	1	1	0	6	10	60	
3	Menurut saya, tampilan warna pada aplikasi enak dilihat dan tidak membosankan	LR	1	1	1	1	1	1	1	1	1	1	10	10	100	77,5
		EF	1	1	1	1	0	1	0	1	1	0	7	10	70	
		MR	1	1	0	0	0	0	1	1	1	0	5	10	50	
		SF	1	1	1	1	1	0	1	1	1	1	9	10	90	
4	Saya merasa fungsi-fungsi dalam aplikasi terintegrasi dengan baik	LR	1	1	1	1	0	1	1	1	1	1	9	10	90	75
		EF	1	1	1	0	0	0	1	1	1	1	7	10	70	
		MR	1	1	1	1	0	0	1	1	1	0	7	10	70	
		SF	1	1	0	1	1	0	1	1	1	0	7	10	70	
5	Saya menganggap tampilan menu dalam aplikasi mudah dikenali	LR	1	1	1	1	1	1	1	1	1	1	10	10	100	77,5
		EF	1	1	1	1	1	0	1	1	1	1	9	10	90	
		MR	1	1	0	0	0	0	1	1	1	0	5	10	50	
		SF	0	1	1	1	0	0	1	1	1	1	7	10	70	
6	Menurut saya, aplikasi ini mudah dibaca	LR	1	1	1	1	1	1	1	1	1	1	10	10	100	82,5
		EF	1	1	1	1	1	1	1	1	1	1	9	10	90	
		MR	1	1	0	1	0	0	1	1	1	0	6	10	60	
		SF	1	1	1	1	0	0	1	1	1	1	8	10	80	
7	Saya merasa aplikasi mudah diinstal	LR	0	1	1	1	1	1	1	1	1	0	8	10	80	75
		EF	1	1	1	1	1	0	1	1	1	1	9	10	90	
		MR	1	1	1	1	0	0	0	1	0	0	5	10	50	
		SF	0	1	1	1	0	1	1	1	1	1	8	10	80	

No	Pertanyaan	Item Jawa ban	Responden ke-										Total Nilai	Skor Mak s	Pers en (100 %)	Bobot Pertanya an
			1	2	3	4	5	6	7	8	9	10				
8	Menurut saya, simbol-simbol pada aplikasi mudah dipahami	LR	1	1	1	1	0	1	1	1	1	1	9	10	90	87,5
		EF	1	1	1	1	1	1	1	1	1	1	10	10	100	
		MR	1	1	1	1	0	0	1	1	1	0	7	10	70	
		SF	1	1	1	1	1	1	1	1	1	0	9	10	90	
9	Saya merasa menu dan tampilan aplikasi mudah diingat	LR	0	0	1	1	1	1	1	1	1	1	8	10	80	75
		EF	1	1	1	1	1	1	1	1	1	0	9	10	90	
		MR	1	1	1	0	0	0	1	1	1	0	6	10	60	
		SF	1	0	1	1	0	1	1	1	0	1	7	10	70	
10	Menurut saya, aplikasi ini mudah dipelajari	LR	1	1	1	1	0	1	1	1	1	0	8	10	80	77,5
		EF	1	1	1	1	1	1	1	1	1	1	10	10	100	
		MR	0	1	1	1	1	0	1	1	0	0	6	10	60	
		SF	0	0	1	1	0	1	1	1	1	1	7	10	70	
11	Saya menganggap aplikasi ini praktis digunakan	LR	1	0	1	1	1	1	1	1	1	1	9	10	90	80
		EF	1	1	1	1	0	1	1	1	1	1	9	10	90	
		MR	1	1	1	1	0	0	1	0	1	0	6	10	60	
		SF	1	1	1	1	0	1	1	1	1	0	8	10	80	
Total												343	440	3430	857,5	

### Lampiran 2.8 Rekapitulasi Pengujian Usability Framework React Native

No	Pertanyaan	Item Jawa ban	Responden ke-										Total Nilai	Skor Mak s	Pers en (100 %)	Bobot Pertanya an
			1	2	3	4	5	6	7	8	9	10				
1	Saya menganggap tampilan aplikasi mudah dikenali	LR	0	0	1	1	1	1	1	1	1	1	8	10	80	65
		EF	0	0	1	1	1	1	1	1	1	1	8	10	80	
		MR	1	1	0	0	1	1	0	0	0	1	5	10	50	
		SF	0	1	0	1	1	1	0	1	0	0	5	10	50	
2	Saya merasa aplikasi ini mudah dioperasikan	LR	1	1	1	1	1	1	0	1	0	1	8	10	80	57,5
		EF	0	1	0	0	1	1	1	1	1	1	7	10	70	
		MR	0	0	0	1	1	1	1	0	0	1	5	10	50	
		SF	0	0	1	0	0	0	0	0	1	1	3	10	30	

No	Pertanyaan	Item Jawaban	Responden ke-										Total Nilai	Skor Makas	Persen (100%)	Bobot Pertanyaan
			1	2	3	4	5	6	7	8	9	10				
3	Menurut saya, tampilan warna pada aplikasi enak dilihat dan tidak membosankan	LR	0	0	1	0	1	1	0	1	1	1	6	10	60	57,5
		EF	1	1	1	1	1	1	1	1	0	1	9	10	90	
		MR	0	1	0	0	0	0	0	0	1	1	3	10	30	
		SF	0	0	0	1	1	1	1	1	0	0	5	10	50	
4	Saya merasa fungsi-fungsi dalam aplikasi terintegrasi dengan baik	LR	1	1	1	0	1	1	1	1	0	1	8	10	80	57,5
		EF	0	0	1	1	1	1	1	1	1	1	8	10	80	
		MR	0	0	0	0	0	1	0	0	0	1	2	10	20	
		SF	0	1	0	0	1	1	0	1	1	0	5	10	50	
5	Saya menganggap tampilan menu dalam aplikasi mudah dikenali	LR	0	0	1	1	1	1	1	1	1	1	8	10	80	55
		EF	0	0	0	0	1	1	0	1	0	1	4	10	40	
		MR	0	1	0	1	0	1	1	0	1	1	6	10	60	
		SF	1	0	0	0	1	1	0	0	0	1	4	10	40	
6	Menurut saya, aplikasi ini mudah dibaca	LR	0	0	1	0	1	1	1	1	0	1	6	10	60	62,5
		EF	0	1	0	0	1	1	0	1	1	1	6	10	60	
		MR	0	1	0	0	1	1	1	0	0	1	5	10	50	
		SF	1	0	1	1	1	1	1	1	0	1	8	10	80	
7	Saya merasa aplikasi mudah diinstal	LR	0	0	1	1	1	1	1	0	0	1	6	10	60	47,5
		EF	1	1	0	1	1	1	0	1	0	1	7	10	70	
		MR	0	0	0	0	1	1	0	1	0	1	4	10	40	
		SF	0	0	0	0	0	0	0	0	1	1	2	10	20	
8	Menurut saya, simbol-simbol pada aplikasi mudah dipahami	LR	0	0	1	0	1	1	0	1	0	1	5	10	50	52,5
		EF	0	1	1	1	0	1	1	1	0	1	7	10	70	
		MR	0	0	0	0	1	1	0	0	0	0	2	10	20	
		SF	1	1	0	1	1	1	0	0	1	1	7	10	70	
9	Saya merasa menu dan tampilan aplikasi mudah diingat	LR	0	1	1	0	1	1	1	1	0	1	7	10	70	62,5
		EF	1	0	0	1	1	1	1	0	1	1	7	10	70	
		MR	0	0	0	0	0	1	0	0	0	1	2	10	20	
		SF	0	1	1	1	1	1	1	1	1	1	9	10	90	
10	Menurut saya, aplikasi ini mudah dipelajari	LR	0	0	1	0	1	1	1	1	0	1	6	10	60	65
		EF	0	1	1	1	0	1	1	1	1	1	8	10	80	
		MR	1	1	0	1	1	1	0	0	0	1	6	10	60	

No	Pertanyaan	Item Jawa ban	Responden ke-										Total Nilai	Skor Mak s	Pers en (100 %)	Bobot Pertanya an
			1	2	3	4	5	6	7	8	9	10				
		SF	0	0	0	0	1	1	1	1	1	1	6	10	60	
11	Saya menganggap aplikasi ini praktis digunakan	LR	0	0	0	0	1	1	1	0	1	1	5	10	50	55
		EF	1	1	0	1	1	1	1	1	1	1	9	10	90	
		MR	0	0	1	0	0	1	1	0	0	1	4	10	40	
		SF	0	0	0	0	1	1	0	1	0	1	4	10	40	
Total																

## **RIWAYAT HIDUP**



**MUH. KIDFARI**, lahir di Kirambu, Desa Ponggiha, Kecamatan Lasusua, Kabupaten Kolaka Utara pada tanggal 19 September 2001. Anak Kedua dari pasangan Arsono S.Sos. M.Si dan Dra. Jahidah. Penulis pertama kali menempuh pendidikan formal pada tahun 2007 di SDN 1 Ponggiha dan tamat pada tahun 2013. Selanjutnya penulis melanjutkan pendidikan ke jenjang Sekolah Menengah Pertama di SMPN 1 Lasusua dan lulus pada tahun 2016, kemudian melanjutkan pendidikan di SMKN 1 Lasusua dan lulus pada tahun 2019. Selanjutnya penulis terdaftar sebagai mahasiswa di Universitas Negeri Makassar, Fakultas Teknik, Jurusan Teknik Informatika dan Komputer, Prodi Pendidikan Teknik Informatika dan Komputer melalui Seleksi Bersama Masuk Perguruan Tinggi Negeri atau SBMPTN.