

**SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN *E-WALLET*
MENGUNAKAN METODE *WEIGHTED PRODUCT*
BERBASIS ANDROID**



Skripsi ini diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer

OLEH:
PUTRI MAHARANI
NIM: 0402191015

FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS NAHDLATUL ULAMA CIREBON
2023 M / 1445 H



UNIVERSITAS NAHDLATUL ULAMA CIREBON

PENGESAHAN STATUS SKRIPSI

JUDUL : SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN
E-WALLET MENGGUNAKAN METODE *WEIGHTED*
PRODUCT BERBASIS ANDROID

PENYUSUN : PUTRI MAHARANI

NIM : 0402191015

Mengijinkan Skripsi Sarjana Komputer ini disimpan di Perpustakaan Universitas Nahdlatul Ulama Cirebon dengan syarat-syarat kegunaan sebagai berikut:

1. Skripsi adalah hak milik Universitas Nahdlatul Ulama.
2. Perpustakaan Universitas Nahdlatul Ulama dibenarkan membuat salinan untuk referensi saja.
3. Perpustakaan juga dibenarkan membuat salinan Skripsi ini sebagai bahan pertukaran antar institusi pendidikan tinggi.
4. Berikan tanda $\sqrt{\quad}$ sesuai dengan kategori Skripsi
 - ☐ Sangat Rahasia (Mengandung isi tentang keselamatan atau kepentingan negara Indonesia).
 - ☐ Rahasia (Mengandung isi tentang kerahasiaan dari suatu organisasi/badan dimana penelitian Skripsi ini dikerjakan).

$\sqrt{\quad}$ Biasa

Disahkan Oleh:

ADE JOHAR MATURIDL, M.KOM

Tanggal:

ABDUL KOHAR, M.KOM

Tanggal:



UNIVERSITAS NAHDLATUL ULAMA CIREBON

PERNYATAAN PENULIS

JUDUL : SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN
E-WALLET MENGGUNAKAN METODE *WEIGHTED*
PRODUCT BERBASIS ANDROID

PENYUSUN : PUTRI MAHARANI

NIM : 0402191015

” Saya menyatakan dan bertanggung jawab dengan sebenarnya bahwa Skripsi ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya. Jika pada waktu selanjutnya ada pihak lain yang mengklaim bahwa Skripsi ini sebagai karyanya, yang disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk membatalkan gelar Sarjana Komputer saya beserta segala hak dan kewajiban yang melekat pada gelar tersebut”.

Cirebon, 5 Agustus 2023

Materai 10.000

PUTRI MAHARANI

NIM: 0402191015



UNIVERSITAS NAHDLATUL ULAMA CIREBON

PERSETUJUAN SKRIPSI

JUDUL : SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN
E-WALLET MENGGUNAKAN METODE *WEIGHTED*
PRODUCT BERBASIS ANDROID

PENYUSUN : PUTRI MAHARANI

NIM : 0402191015

Skripsi ini telah diperiksa dan disetujui,
Cirebon, 5 Agustus 2023

Pembimbing I

Pembimbing II

ADE JOHAR MATURIDI, M.KOM

NIDN: 0422037605

ABDUL KOHAR, M.KOM

NIDN: 0412068704



UNIVERSITAS NAHDLATUL ULAMA CIREBON

PENGESAHAN SKRIPSI

JUDUL : SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN
E-WALLET MENGGUNAKAN METODE WEIGHTED
PRODUCT BERBASIS ANDROID

PENYUSUN : PUTRI MAHARANI

NIM : 0402191015

Skripsi ini telah diujikan dan dipertahankan dihadapan Dewan Penguji pada Sidang Skripsi tanggal 22 Juli 2023. Menurut pandangan kami, Skripsi ini memadai untuk penganugrahan gelar Sarjana Komputer (S. Kom).

Tanggal

Tanda tangan

Ketua Jurusan,

HERI HERYANTO, M.KOM

NIDN: 0416068608

Pembimbing 1,

ADE JOHAR MATURIDI, M.KOM

NIDN: 0422037605

Pembimbing 2,

ABDUL KOHAR, M.KOM

NIDN: 0412068704

Penguji 1,

ROSIDIN, M.KOM

NIDN: 0415028303

Penguji 2,

DICKY ANDIKA SULAEMAN, M.KOM

NIDN: 0409069402

Mengetahui,

Dekan Fakultas Ilmu Komputer
Universitas Nahdlatul Ulama Cirebon

ROSIDIN, M.KOM

NIDN: 0415028303

ABSTRACT

E-Wallet (digital wallet) is a service or platform that offers a type of assistance like a wallet, namely a place to store cash, but the money storage is sophisticated (on the web). Due to the increasing number of E-Wallet applications in everyday life, this creates problems in determining which E-Wallet application is the most effective to use. To overcome this problem, research was carried out using a decision support system to determine an E-Wallet by knowing what factors support the use of an E-Wallet, so that it can determine the ideal E-Wallet application for use by many people. In this research, the design and creation of an Android-based E-Wallet selection decision support system uses the weighted product method. This method uses several weighted criteria, namely: admin fees, ease of top up, features, promos and services. The results of this research show that ShopeePay is the best alternative with the highest value, namely 0.21769, followed by OVO (0.21111), DANA (0.19638), GoPay (0.19302) and LinkAja (0.18180).

Keywords: E-Wallet, Weighted Product, decision support system, Android

ABSTRAK

E-Wallet (dompet digital) adalah layanan atau platform yang menawarkan jenis bantuan seperti dompet, yaitu tempat menyimpan uang tunai, tetapi penyimpanan uangnya canggih (di web). Karena semakin banyaknya aplikasi *E-Wallet* dalam kehidupan sehari-hari, hal ini menimbulkan permasalahan dalam menentukan aplikasi *E-Wallet* mana yang paling efektif untuk digunakan. Untuk mengatasi permasalahan tersebut maka dilakukan penelitian menggunakan sistem pendukung keputusan untuk menentukan *E-Wallet* dengan mengetahui faktor apa saja yang mendukung penggunaan *E-Wallet*, sehingga dapat menentukan aplikasi *E-Wallet* yang ideal untuk digunakan oleh banyak orang. Dalam penelitian ini desain dan pembuatan sistem pendukung keputusan pemilihan *E-Wallet* berbasis android menggunakan metode weighted product. Metode ini menggunakan beberapa bobot kriteria yaitu: biaya admin, kemudahan top up, fitur, promo dan layanan. Hasil penelitian ini menunjukkan bahwa ShopeePay menjadi alternatif terbaik dengan nilai tertinggi yaitu 0.21769 yang diikuti oleh OVO (0.21111), GoPay (0.19638), DANA (0.19302) dan LinkAja (0.18180).

Kata kunci: E-Wallet, Weighted Product, Sistem Pendukung Keputusan, Android

KATA PENGANTAR



Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah kepada penulis, sehingga dalam penyusunan skripsi ini penulis dapat menyelesaikan tanpa mengalami hambatan yang berarti.

Tersusunnya skripsi ini merupakan hasil partisipasi dan bantuan serta bimbingan dari berbagai pihak baik secara moril maupun materil, oleh karena itu pada kesempatan yang berharga ini, penulis memohon kepada Allah SWT semoga Allah memberikan balasan kepada Bapak, Ibu serta rekan-rekan yang telah memberikan bantuan kepada penulis.

Selanjutnya penulis mengucapkan terima kasih dan penghargaan yang sedalam-dalamnya kepada:

1. Tuhan Yang Maha Esa, yang telah memberikan rahmat, hidayah serta kesehatan sehingga penulis dapat menyelesaikan skripsi ini.
2. Kedua Orang tua yang selalu memberikan doa dan dukungan.
3. Untuk diri saya sendiri, terima kasih karena telah mau berjuang dan bertahan sampai saat ini dan mampu berada di titik ini.
4. Bapak Dr. Agus Sugiarto, SH., MH., MM selaku Rektor Universitas Nahdlatul Ulama Cirebon
5. Bapak Rosidin, M. Kom selaku Dekan Fakultas Ilmu Komputer Universitas Nahdlatul Ulama Cirebon.
6. Bapak Heri Heryanto, M. Kom selaku Ketua Program Studi Teknik Informatika Universitas Nahdlatul Ulama Cirebon.
7. Bapak Ade Johar Maturidi, M. Kom selaku Dosen Pembimbing I yang meluangkan waktu, tenaga dan pikiran untuk memberikan bimbingan, berbagai pengalaman, arahan, nasehat, motivasi dan arahan dalam pengembangan program hingga penyusunan skripsi ini.
8. Bapak Abdul Kohar, M. Kom selaku Dosen Pembimbing II yang selalu memberi masukan dan arahan dalam penyusunan laporan skripsi ini.

9. Dosen-dosen Fakultas Ilmu Komputer Universitas Nahdlatul Ulama Cirebon.
10. Seluruh sivitas akademika Universitas Nahdlatul Ulama Cirebon khususnya para dosen, terima kasih atas segala ilmu dan bimbingannya.
11. Teman-teman angkatan 2019, yang berjuang bersama untuk meraih mimpi, terima kasih atas kenangan indah yang dijalin bersama.
12. Kepada pemilik NIM 0402191004 terima kasih sudah senantiasa meluangkan waktu, tenaga dan dukungan yang luar biasa sehingga penulis dapat menyelesaikan skripsi tepat waktu.
13. Semua pihak yang tidak dapat penulis sebutkan satu persatu atas bantuan, masukan, dukungan dan motivasi kepada penulis.

Segala rintangan dan hambatan yang penulis alami selama penyusunan skripsi ini tidaklah membuat penulis putus asa dan putus harapan. Penulis mengharap kritik dan saran yang bersifat membangun demi kemanfaatan skripsi ini.

Akhir kata hanya kepada Allah penulis berserah diri semoga petunjuknya selalu menyertai kita semua sehingga kita selalu berada di jalan dan ridho-Nya.

Semoga segala kebaikan dan pertolongan semuanya mendapatkan berkah dari Allah SWT. Akhir kata penulis mohon maaf apabila masih banyak kekurangan dalam penyusunan skripsi ini. Semoga skripsi ini dapat bermanfaat bagi semua pihak yang memerlukan. Aamiiiin.

Cirebon 5 Agustus 2023

Penulis

DAFTAR ISI

PENGESAHAN STATUS SKRIPSI	ii
PERNYATAAN PENULIS.....	iii
PERSETUJUAN SKRIPSI.....	iv
PENGESAHAN SKRIPSI	v
ABSTRAK.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metode Penelitian	4
1.6.1 Metode Pengumpulan Data	4
1.6.2 Metode Analisis Data	4
1.6.3 Metode Pengembangan Sistem.....	4
1.7 Sistematika Penulisan.....	6
1.8 Kerangka Berfikir	7
BAB II LANDASAN TEORI.....	8
2.1 Keterkaitan Dengan Ayat Al-Qur'an	8
2.2 Tinjauan Pustaka.....	8
2.3 Sistem Pendukung Keputusan	11
2.3.1 Pengertian Sistem Pendukung Keputusan	11
2.3.2 Tahapan Pengambilan Keputusan.....	12
2.3.3 Komponen Sistem Pendukung Keputusan	12
2.3.4 Manfaat Sistem Pendukung Keputusan.....	13
2.4 Metode <i>Weighted Product</i> (WP)	13

2.5 E-Wallet	15
2.5.1 Pengertian <i>E-Wallet</i>	15
2.5.2 Jenis-Jenis Dompot Digital	16
2.6 Android	20
2.6.1 Sejarah Android	20
2.6.2 Versi Android	20
2.7 Android Studio	21
2.7.1 Pengertian Android Studio	21
2.7.2 Antarmuka pengguna	22
2.7 Bahasa Pemograman Java	23
2.8 JDK	23
2.9 Basis Data	24
2.10 Firebase	24
2.10.1 <i>Firebase Realtime database</i>	25
2.11 BalsamiqMockup	26
2.12 UML (Unified Modelling Language)	27
2.12.1 Pengertian UML.....	27
2.12.2 <i>Use Case Diagram</i>	27
2.12.3 <i>Activity Diagram</i>	29
2.12.4 <i>Sequence Diagram</i>	30
2.12.4 <i>Class Diagram</i>	31
2.13 Flowchart	32
2.14 Black Box	34
2.14.1 Pengertian <i>Black Box</i>	34
2.14.2 Kelebihan <i>Black Box</i>	34
2.14.3 Kekurangan <i>Black Box</i>	34
BAB III ANALISIS DAN PERANCANGAN SISTEM	35
3.1 Analisis Masalah	35
3.2 Analisa Kebutuhan	35
3.2.1 Analisa Kebutuhan Perangkat Keras.....	35
3.2.2 Analisa Kebutuhan Perangkat Lunak.....	35
3.3 Tujuan Perancangan Sistem	36

3.4 Flowchart Perancangan Sistem	36
3.5 Perancangan Sistem	37
3.5.1 Perancangan Sistem Pendukung Keputusan.....	37
3.5.2 Perancangan Sistem	44
3.5.3 Perancangan Database.....	47
3.5.4 Perancangan <i>Interface</i> (Antarmuka)	48
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	58
4.1 Implementasi Sistem.....	58
4.1.1 Implementasi Tampilan <i>Interface</i>	58
4.1.2 Implementasi Metode <i>Weighted Product</i>	69
4.1.3 Implementasi <i>Firebase Realtime database</i>	71
4.2 Pengujian Sistem	75
4.2.1 Pengujian <i>Black Box</i>	75
4.2.2 Pengujian Akurasi	76
BAB V PENUTUP.....	78
5.1 Kesimpulan	78
5.2 Saran	79
DAFTAR PUSTAKA.....	80

DAFTAR GAMBAR

Gambar 1. 1 Metode Waterfall	5
Gambar 1. 2 Kerangka Berfikir	7
Gambar 1. 3 Android Studio	21
Gambar 1. 4 Antarmuka Android Studio	22
Gambar 2. 1 Dompot Digital	15
Gambar 2. 2 Tampilan OVO	16
Gambar 2. 3 Tampilan GoPay	17
Gambar 2. 4 Tampilan DANA	18
Gambar 2. 5 Tampilan LinkAja.....	19
Gambar 2. 6 Tampilan ShopeePay	19
Gambar 2. 7 Android	20
Gambar 2. 8 Versi Android	21
Gambar 2. 9 Android Studio	21
Gambar 2. 10 Interface Android Studio.....	22
Gambar 2. 11 Java	23
Gambar 2. 12 JDK	23
Gambar 2. 13 Fitur Firebase.....	26
Gambar 2. 14 Balsamiq Mockup.....	26
Gambar 2. 15 Contoh Pemanfaatan Use Case Diagram	28
Gambar 2. 16 Contoh Pemanfaatan Activity Diagram	30
Gambar 2. 17 Contoh Pemanfaatan Sequence Diagram.....	31
Gambar 2. 18 Contoh Pemanfaatan Class Diagram	32
Gambar 3. 1 <i>Flowchart</i> Perancangan Sistem	36
Gambar 3. 2 Use Case Diagram	44
Gambar 3. 3 Activity Diagram	45
Gambar 3. 4 Sequence Diagram	46
Gambar 3. 5 Rancangan Halaman Login	48

Gambar 3. 6 Rancangan Halaman Register	49
Gambar 3. 7 Rancangan Halaman Utama	50
Gambar 3. 8 Rancangan Halaman Penilaian ShopeePay	51
Gambar 3. 9 Rancangan Halaman AlertDialog	52
Gambar 3. 10 Rancangan Halaman Penilaian OVO	53
Gambar 3. 11 Rancangan Halaman Penilaian GoPay	54
Gambar 3. 12 Rancangan Halaman Penilaian DANA	55
Gambar 3. 13 Rancangan Halaman Penilaian LinkAja	56
Gambar 3. 14 Rancangan Halaman Hasil Perangkingan	57
Gambar 4. 1 Tampilan Halaman Login	59
Gambar 4. 2 Tampilan Halaman Register	60
Gambar 4. 3 Tampilan Halaman Utama	61
Gambar 4. 4 Tampilan Halaman Penilaian ShopeePay	62
Gambar 4. 5 Tampilan Halaman AlertDialog	63
Gambar 4. 6 Tampilan Halaman Penilaian OVO	64
Gambar 4. 7 Tampilan Halaman Penilaian GoPay	65
Gambar 4. 8 Tampilan Halaman Penilaian DANA	66
Gambar 4. 9 Tampilan Halaman Penilaian LinkAja	67
Gambar 4. 10 Tampilan Halaman Hasil Perangkingan	68
Gambar 4. 11 Rules firebase realtime database Aplikasi Pemilihan E-Wallet	72

DAFTAR TABEL

Tabel 1 Tinjauan Pustaka	11
Tabel 2 Simbol Use Case Diagram.....	28
Tabel 3 Simbol Activity Diagram.....	29
Tabel 4 Simbol Sequence Diagram	30
Tabel 5 Simbol Class Diagram.....	32
Tabel 6 Simbol Flowchart	34
Tabel 7 Perancangan Model Weighted Product	37
Tabel 8 Bobot Kriteria	38
Tabel 9 Bobot Kepentingan.....	38
Tabel 10 Perbaikan Bobot Setiap Kriteria	39
Tabel 11 Nilai alternatif setiap kriteria	41
Tabel 12 Hasil Pemeringkatan.....	43
Tabel 13 Perancangan Database	47
Tabel 14 Implementasi Merode Weighted Product	71
Tabel 15 Add Single Value Event Listener.....	73
Tabel 16 Add Event Event Listener.....	74
Tabel 17 Pengujian Black Box	76
Tabel 18 Pengujian Akurasi	76

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring berjalannya waktu, perkembangan teknologi semakin hari semakin meningkat pesat, salah satu dampaknya yaitu timbul kebiasaan baru dalam hal adanya perubahan metode pembayaran transaksi dari tunai menjadi non tunai. Dengan adanya metode pembayaran melalui kartu dimana kartu ini bisa merupakan kartu kredit maupun kartu debit yang diterbitkan oleh sebuah Bank, negeri maupun swasta, di bawah pengawasan Bank Indonesia. Saat ini sedang terjadi adanya metode pembayaran yang dikeluarkan oleh sebuah startup-startup, yang dikenal dengan Electronic Wallet (*E-Wallet*). (Sitinjak & Nurlela, 2022).

Munculnya *E-Wallet* membuat transaksi menjadi lebih mudah, artinya tidak perlu lagi membawa uang tunai atau kartu. *E-Wallet* telah diatur oleh Bank Indonesia melalui Peraturan Bank Indonesia. Peraturan tersebut dengan jelas menyatakan bahwa *E-Wallet* sebagai salah satu metode transaksi pembayaran wajib mendapatkan izin penyelenggaraan dari Bank Indonesia sehingga *E-Wallet* dapat disebut sebagai metode pembayaran yang sah saat ini, tidak hanya tunai, kartu kredit dan kartu debit. Jumlah *E-Wallet* saat ini sudah ada 38 *E-Wallet* yang telah mendapatkan izin dari Bank Indonesia. Dimana 5 *E-Wallet* tersebut adalah OVO, GoPay, DANA, LinkAja dan ShopeePay.

Karena meningkatnya penggunaan *E-Wallet* dalam kehidupan sehari-hari dan banyaknya aplikasi yang beredar, hal ini menimbulkan permasalahan dalam menentukan aplikasi *E-Wallet* mana yang paling efektif untuk digunakan. Untuk mengatasi permasalahan tersebut maka dilakukan penelitian menggunakan sistem pendukung keputusan untuk menentukan alternatif *E-Wallet* dengan mengetahui faktor apa saja yang mendukung penggunaan alternatif *E-Wallet*. itu, sehingga Bisa menentukan aplikasi dompet elektronik yang sangat ideal untuk digunakan oleh banyak orang.

Metode yang dapat digunakan untuk membantu pengambilan keputusan dalam proses pemilihan *E-Wallet* adalah *Weighted Product*. Menurut (Sya'id Maftuh, 2022). Metode *weighted product* merupakan salah satu metode sistem pendukung keputusan yang dapat dilakukan. Metode *weighted product* merupakan sebuah metode yang didalamnya memuat sebuah keputusan dengan cara perkalian untuk dapat menghubungkan nilai atribut, dimana nilai atribut harus dipangkatkan dengan bobot atribut yang bersangkutan. Metode ini dapat menentukan bobot kepentingan dari masing-masing kriteria, apakah itu sangat penting, penting, cukup penting, tidak penting, atau sangat tidak penting.

Berdasarkan hasil penelitian yang dilakukan oleh (Nadila dkk., 2021) metode *Weighted Product* (WP) memiliki tingkat kesesuaian yang lebih tinggi sebesar 99,99857%. dibandingkan dengan metode Moore dengan nilai 99,99779% dan metode Promethee dengan nilai 99.99679%. Metode ini dipilih karena bisa tentukan nilai bobot dari setiap atribut, kemudian dilanjutkan dengan proses pemeringkatan yang akan menentukan Dompot Digital yang sesuai dengan kriteria. Informasi yang dihasilkan dari sistem ini berdasarkan data kriteria yaitu: (Biaya Admin, Kemudahan Top Up, Fitur, Promo dan Layanan). Peringkat yang dihasilkan dapat digunakan untuk membantu masyarakat dalam mengambil keputusan dalam menentukan dompet digital yang terbaik. Menggunakan metode *Weighted Product* (WP) lebih efisien karena waktu diperlukan dalam perhitungan yang lebih singkat.

Dari permasalahan diatas pada penelitian ini penulis membuat suatu sistem berbasis android untuk proses pemilihan Dompot Digital terbaik, sebagai solusi dari permasalahan tersebut yang ada. Oleh karena itu, dalam penelitian ini penulis mencoba mengangkat masalah dengan judul **“SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN *E-WALLET* MENGGUNAKAN METODE *WEIGHTED PRODUCT* BERBASIS ANDROID”**.

1.2 Rumusan Masalah

Berdasarkan dari latar belakang yang dijelaskan diatas, maka dapat dirumuskan rumusan masalah sebagai berikut:

1. Bagaimana merancang sistem pembuatan aplikasi pemilihan *E-Wallet* menggunakan Android Studio.

2. Bagaimana mengimplementasikan metode *weighted product* pada sistem pendukung keputusan pemilihan *E-Wallet*.

1.3 Batasan Masalah

Dalam perancangan sistem pendukung keputusan untuk menentukan Dompot Digital ini penulis membatasi masalah sebagai berikut:

1. Kriteria yang digunakan dalam aplikasi ini berupa: Biaya Admin, Kemudahan Top Up, Fitur, Promo dan Layanan.
2. Aplikasi ini dirancang hanya untuk pengguna android.
3. Metode yang digunakan pada aplikasi adalah metode *Weighted Product* (WP).
4. Sistem yang dibangun berbasis android dengan menggunakan bahasa pemograman Java.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Untuk dapat merancang dan membangun sistem pembuatan aplikasi pemilihan *E-Wallet* menggunakan software Android Studio.
2. Penerapan sistem pendukung keputusan pemilihan *E-Wallet* dengan menggunakan metode *Weighted Product* (WP) yang menghasilkan *E-Wallet* yang efektif digunakan.

1.5 Manfaat Penelitian

1. Bagi Penulis
 - Menambah pengetahuan dan wawasan bagi penulis melalui perancangan dan pembuatan aplikasi sistem pendukung keputusan dan metode yang digunakan.
 - Untuk menerapkan dan mengembangkan ilmu yang telah di pelajari di perkuliahan.

2. Bagi Universitas

Dapat dijadikan sebagai sarana pengenalan atau penambahan informasi serta referensi di Perpustakaan Universitas Nahdlatul Ulama Cirebon mengenai permasalahan yang terkait dengan sistem tersebut.

3. Bagi Masyarakat Umum

- Membantu pengguna dan masyarakat pada umumnya untuk mempermudah dalam memilih *E-Wallet*.
- Aplikasi yang bersifat mobile dapat digunakan pada semua smartphone dengan sistem operasi android.

1.6 Metode Penelitian

Metode penelitian adalah salah satu cara atau prosedur yang digunakan untuk dapat mengumpulkan data, dengan perantara teknis tertentu. Metode penelitian yang digunakan adalah sebagai berikut:

1.6.1 Metode Pengumpulan Data

1. Kuesioner

Mengumpulkan data atau informasi mengenai *E-Wallet* apa saja yang sering digunakan dengan cara memberikan pertanyaan seputar *E-Wallet*.

2. Tinjauan Literatur

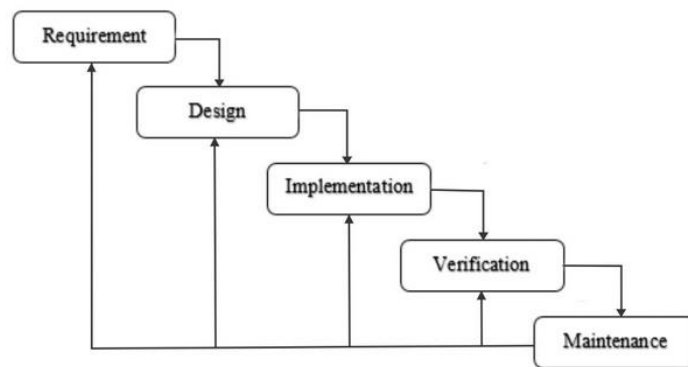
Mencari literatur yang terkait dengan *E-Wallet* melalui buku dan internet.

1.6.2 Metode Analisis Data

Metode analisis data yang digunakan dalam penelitian ini adalah metode Unified Modelling Language (UML) merupakan salah satu metode yang dapat digunakan untuk memahami dan mendokumentasikan sistem perangkat lunak secara visual. Jenis-jenis UML yang digunakan antara lain yaitu *Use Case Diagram*, *Sequence Diagram*, *Activity Diagram* dan *Class Diagram*.

1.6.3 Metode Pengembangan Sistem

Metode yang digunakan untuk membangun sistem ini adalah Model *WaterFall*. Model ini merupakan pendekatan pembangunan perangkat lunak sistematis, dengan beberapa tahapan yaitu: Sistem Rekayasa, Analisis, Desain, Pengkodean, Pengujian dan Pemeliharaan.



Gambar 1. 1 Metode Waterfall

Penjelasan dari Metode Waterfall adalah sebagai berikut:

1. Rekayasa Sistem Pemodelan (*System Engineering*) ini diawali dengan pencarian kebutuhan keseluruhan sistem yang akan diterapkan pada bentuk perangkat lunak. Ini sangat penting, mengingat perangkat lunak harus dapat berinteraksi dengan elemen lain seperti perangkat keras, database, dan sebagainya.
2. Analisis adalah tahapan menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pengembangan perangkat lunak. Dalam hal ini dilakukan dengan menganalisis dokumen-dokumen yang digunakan dalam informasi data.
3. Pada tahap ini bertujuan untuk memberikan gambaran tentang tampilan yang akan dibuat dilakukan beserta gambaran tahapan yang akan dilakukan. Pada tahap perancangan penulis menggunakan UML (Unified Model Language) sebagai pemodelan sistem, UML yang digunakan adalah: *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. Di panggung desain penulis juga menjelaskan desain database dan desain antarmuka menggunakan perangkat lunak Balsamiq yang nantinya akan menjadi acuan dalam pembuatan kode program.
4. Coding adalah tahap menerjemahkan data/pemecahan masalah perangkat lunak yang telah dirancang dengan bahasa pemrograman yang telah

ditentukan dan digunakan dalam pembuatan sistem dengan menggunakan perangkat lunak Android Studio dengan bahasa pemrograman Java.

5. Testing merupakan tahap pengujian terhadap program yang telah dibuat. Tes ini dimulai dengan membuat test case untuk masing-masing fungsi dalam perangkat lunak untuk sistem informasi pemrosesan data. Data tersebut kemudian dilanjutkan dengan pengujian modul dan terakhir di antarmuka untuk memastikannya tidak ada kesalahan dan semua berjalan dengan baik dan input diberikan hasilnya seperti yang diinginkan.
6. Pemeliharaan adalah perangkat lunak yang telah tersedia perubahan sesuai dengan permintaan pengguna. Pemeliharaan bisa dilakukan jika ada permintaan penambahan fungsi sesuai dengan keinginan pengguna atau adanya pertumbuhan dan perkembangan baik perangkat lunak maupun perangkat keras.

1.7 Sistematika Penulisan

Adapun sistematika penulisan ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, sistematika penulisan dan kerangka berpikir.

BAB II LANDASAN TEORI

Pada bab ini berisi tentang pengertian dan teori-teori yang digunakan sebagai landasan atau dasar dari merancang sistem yang berhubungan dengan sistem informasi pendukung keputusan pemilihan *E-Wallet* dengan metode weighted product.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Pada bab menjelaskan tentang sistem serta deskripsi dari analisis sistem yang akan dijadikan sebagai petunjuk untuk perancangan sistem selanjutnya.

BAB IV IMPLEMENTASI SISTEM DAN PENGUJIAN

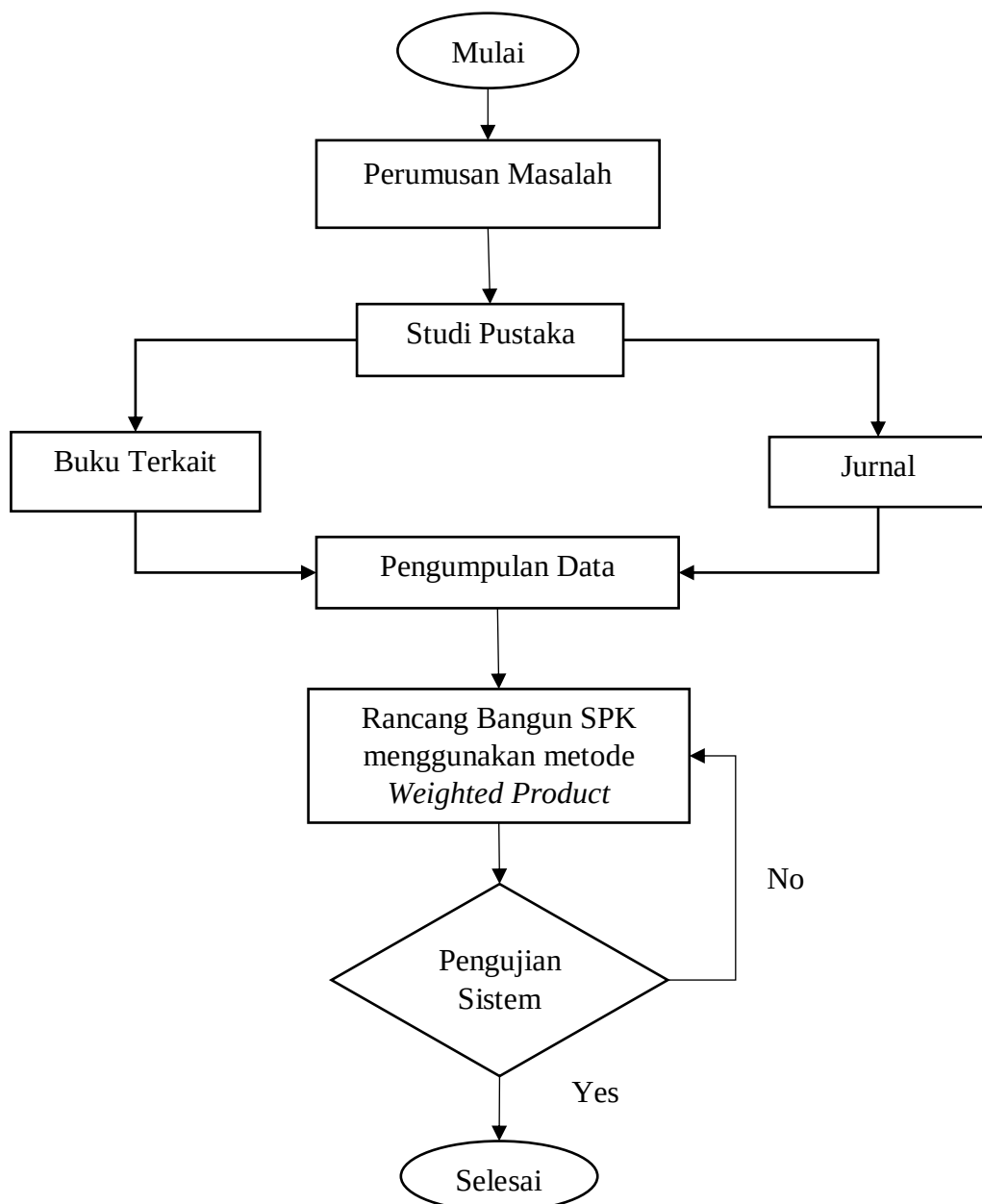
Bab ini menguraikan langkah-langkah dalam implementasi sistem, disertai dengan komponen-komponen kebutuhan sistem.

BAB V PENUTUP

Mengemukakan kesimpulan yang diambil dari hasil penelitian dan perancangan sistem, serta saran-saran untuk pengembangan selanjutnya, agar dapat dilakukan perbaikan-perbaikan di masa yang akan datang.

1.8 Kerangka Berfikir

Berikut adalah gambaran dari framework pada penelitian ini yang ditunjukkan pada gambar di bawah ini.



Gambar 1. 2 Kerangka Berfikir

BAB II

LANDASAN TEORI

2.1 Keterkaitan Dengan Ayat Al-Qur'an

Prinsip pertama dan yang terpenting dalam pengambilan keputusan adalah adil. Dari segi keadilan dapat diartikan sebagai tidak memihak dan seimbang. Prinsip keadilan ini sangat penting karena dengan adanya keadilan putusan yang diambil tidak merugikan orang lain.

Dalam QS. An - Nisa ayat 58, Allah berfirman:

﴿إِنَّ اللَّهَ يَأْمُرُكُمْ أَنْ تُؤَدُّوا الْأَمَانَاتِ إِلَىٰ أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوا بِالْعَدْلِ إِنَّ اللَّهَ نِعِمَّا يَعِظُكُمْ بِهِ إِنَّ اللَّهَ كَانَ سَمِيعًا بَصِيرًا﴾

Innallāha ya`murukum an tu`addul-amānāti ilā ahlihā wa izā ḥakamtum bainan-nāsi an taḥkumu bil-'adl, innallāha ni'immā ya'izukum bih, innallāha kāna samī'am baṣīrā

Artinya: Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha Mendengar lagi Maha Melihat.

2.2 Tinjauan Pustaka

Setelah melakukan beberapa studi literatur, terdapat beberapa penelitian yang memiliki kesamaan dan relevansi dengan penelitian yang penulis lakukan.

Berikut beberapa penelitian terdahulu yang digunakan oleh penulis sebagai acuan dalam penyusunan Sistem Pendukung Keputusan Pemilihan *E-Wallet* Menggunakan Metode *Weighted Product* Berbasis Android.

No.	Jurnal	Permasalahan	Solusi	Hasil Akhir
1.	Judul: Sistem Pendukung Keputusan Pemilihan <i>E-Wallet</i> Menggunakan Metode Analytic Hierarchy Process (AHP) Penulis: Windy Amelia Putri, Dyana Rachmawati, dan Widya Sarah Silalahi Penerbit: MALCOM: Indonesian Journal of Machine Learning and Computer Service	Karena meningkatnya penggunaan <i>E-Wallet</i> dalam kehidupan sehari-hari dan banyaknya aplikasi yang sudah beredar, hal ini menimbulkan permasalahan dalam menentukan aplikasi <i>E-Wallet</i> mana yang paling efektif untuk digunakan.	Untuk mengatasi masalah ini maka dilakukan penelitian untuk menentukan alternatif <i>E-Wallet</i> dengan mengetahui faktor apa saja yang mendukung alternatif penggunaan <i>E-Wallet</i>, sehingga dapat menentukan aplikasi <i>E-Wallet</i> yang ideal untuk digunakan oleh banyak orang.	Terbentuknya Sistem pendukung Keputusan Pemilihann <i>E-Wallet</i> Menggunakan metode Analytic Hierarchy Process (AHP) sebagai solusi untuk dapat memberikan rekomendasi dalam pemilihan aplikasi <i>E-Wallet</i> yang paling efektif.
2.	Judul: Penerapan Metode <i>Weighted</i>	Kurang tercapainya efisiensi kerja yang kondusif dan hasil laporan yang	Dari permasalahan tersebut peneliti membuat sistem	Sistem Pendukung Keputusan untuk pemilihan Siswa

	<i>Product</i> (WP) Dalam Menentukan Pemilihan Siswa Teladan Pada SD Negeri 3 Jatipohon Kecamatan Grobogan Berbasis Web Penulis: Rinda Mesra Hayu Penerbit: Universitas Semarang	kurang akurat serta sering terjadi eror atau kesalahan hilangnya data. Selain itu, input manual juga dimungkinkan menyebabkan data dapat diubah sehingga laporan penilaian siswa yang kurang objektif.	berbasis web untuk proses penilaian siswa teladan.	Teladan dengan metode <i>Weighted Product</i> (WP) yang terbentuk akan memudahkan sekolah dalam menentukan siswa teladan pada SD Negeri 3 Jatipohon.
3.	Judul: Sistem Pendukung Keputusan Untuk Menentukan Kualitas Beras Berbasis Android Pada Kelompok Tani Kabupaten Aceh Selatan Menggunakan Metode <i>Weighted Product</i>	Banyaknya faktor yang membuat kualitas beras berbeda-beda tersebut, dibutuhkan penilaian pada kualitas beras yang dipanen. Kelompok tani pada Kabupaten Aceh Selatan memiliki banyak petani padi yang bercocok tanam	Berdasarkan permasalahan tersebut, maka dibutuhkan suatu program aplikasi sistem pendukung keputusan untuk menentukan kualitas beras pada kelompok tani Kabupaten Aceh Selatan yang mampu memberikan penilaian terhadap	Terbentuknya Sistem Pendukung Keputusan Untuk Menentukan Kualitas Beras Berbasis Android Pada Kelompok Tani Kabupaten Aceh Selatan dapat membantu para petani dalam penilaian pada beras yang dihasilkannya.

	Penulis: Andrian Purba Penerbit: Universitas Pembangunan Panca Budi Medan	tanaman padi tersebut. Jenis kualitas beras yang dihasilkan juga berbeda-beda tiap daerah kelompok tani pada Kabupaten Aceh Selatan.	kualitas beras yang dihasilkan	
--	---	--	-----------------------------------	--

Tabel 1 Tinjauan Pustaka

2.3 Sistem Pendukung Keputusan

2.3.1 Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan manipulasi data. SPK dirancang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, dan menentukan pendekatan yang digunakan dalam proses pengambilan keputusan, sampai mengevaluasi pemilihan alternatif. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. (Noviansyah dkk., t.t.)

Agar berhasil mencapai tujuannya, sistem harus sederhana, kuat, mudah dikendalikan, mudah beradaptasi sepenuhnya dengan hal-hal penting dan mudah dikomunikasikan. Secara implisit juga berarti bahwa sistem ini harus berbasis komputer dan digunakan selain kemampuan pemecahan masalah seseorang. Sistem Pendukung Keputusan secara intelektual memanfaatkan sumber daya individu dengan kemampuan komputer untuk meningkatkan kualitas keputusan. Jadi ini adalah sistem pendukung berbasis komputer untuk manajemen pengambilan keputusan terkait masalah semi terstruktur.

2.3.2 Tahapan Pengambilan Keputusan

Menurut Herbert A. Simon, tahapan dalam pengambilan keputusan adalah sebagai berikut:

- **Tahap Pemahaman (Tahap Intelijen)**

Tahapan ini merupakan proses penelusuran dan pendeteksian ruang lingkup permasalahan serta proses identifikasi masalah. Data input diperoleh, diproses dan diuji untuk mengidentifikasi masalah.

- **Tahap Desain (Fase Desain)**

Tahapan ini merupakan proses pengembangan dan pencarian alternatif tindakan/solusi yang dapat diambil. Ini merupakan representasi dari kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam memeriksa permasalahan yang ada.

- **Tahap Seleksi (Fase Pilihan)**

Tahap ini dilakukan dengan memilih diantara berbagai alternatif solusi yang dimunculkan pada tahap perencanaan sehingga ditentukan/memperhatikan kriteria berdasarkan tujuan yang ingin dicapai.

- **Tahap Implementasi (Implementation Phase)**

Tahapan ini dilakukan dengan mengimplementasikan rancangan sistem yang telah dibuat pada tahap perancangan dan mengimplementasikan alternatif tindakan yang telah dipilih pada tahap pemilihan.

2.3.3 Komponen Sistem Pendukung Keputusan

- a. Manajemen data. Termasuk Database, yang berisi data relevan untuk berbagai situasi dan diatur oleh sistem itu disebut Database Management System (DBMS).
- b. Model Manajemen. Melibatkan keuangan, statistik, ilmu manajemen atau berbagai model kuantitatif lainnya, sehingga memberikan sistem kemampuan analitis dan manajemen sistem yang diperlukan.
- c. Komunikasi (Subsistem Dialog). Pengguna dapat berkomunikasi dan perintah kepada DSS Melalui Subsistem ini. Itu berarti menyediakan antarmuka.

2.3.4 Manfaat Sistem Pendukung Keputusan

SPK dapat memberikan berbagai manfaat dan keuntungan. Manfaat yang dapat diambil dari SPK adalah:

- SPK memperluas kemampuan pengambil keputusan untuk memproses data/informasi bagi pengguna.
- SPK membantu pengambil keputusan untuk memecahkan masalah, terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
- SPK dapat menghasilkan solusi lebih cepat dan hasilnya dapat diandalkan.

Walaupun sebuah SPK belum tentu dapat menyelesaikan permasalahan yang dihadapi oleh pengambil keputusan, namun dapat menjadi stimulan bagi pengambil keputusan dalam memahami permasalahan, karena mampu menghadirkan berbagai alternatif solusi.

2.4 Metode *Weighted Product* (WP)

Weighted Product (WP) adalah suatu metode yang menggunakan perkalian untuk menghubungkan rating atribut, di mana rating setiap atribut harus dipangkatkan dulu dengan bobot yang bersangkutan. Proses ini sama halnya dengan proses normalisasi. *Weighted Product* (WP) merupakan salah satu metode yang digunakan untuk menyelesaikan masalah *Multi Attribute Decision Making* (MADM). MADM adalah suatu metode yang digunakan untuk mencari alternatif paling optimal dari sejumlah alternatif optimal dengan kriteria tertentu. Inti dari MADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perangkingan yang akan menyeleksi alternatif yang sudah diberikan. (Noviansyah dkk., t.t.)

Pembobotan metode *weighted product* dihitung berdasarkan level minat. Tingkat kepentingan metode *Weighted Product* yaitu:

- A. Sangat tidak penting
- B. Tidak penting
- C. Cukup penting
- D. Penting

E. Sangat penting

Proses normalisasi bobot kriteria (W), $\sum W = 1$ adalah:

$$W_J = \frac{W_J}{\sum W_J}$$

Keterangan:

W_j : Bobot atribut

$\sum W_j$: Jumlah bobot atribut

Preferensi untuk alternatif diberikan:

$$S_i = \prod_{j=1}^n X_{ij}^{w_j}$$

Keterangan:

S_i = Hasil normalisasi keputusan pada alternatif ke-i

X_{ij} = Peringkat Alternatif per atribut

W_j = Berat atribut

i = Alternatif

J = Atribut

Dalam alternatif ini di mana $\sum W_j = 1$. W_j adalah pangkat positif untuk atribut keuntungan, dan negatif untuk atribut biaya. Preferensi relatif dari setiap alternatif (V), diberikan:

$$V_i = \frac{\prod_{j=1}^n X_{ij}^{w_j}}{\prod_{j=1}^n X_{ij} * W_j} \quad \text{atau} \quad V_i = \frac{S_i}{\sum S_i}$$

Keterangan:

V_i = Hasil preferensi alternatif - i

X_{ij} = Peringkat alternatif per atribut

W_j = Berat atribut

i = Alternatif

J = Atribut

2.5 E-Wallet

2.5.1 Pengertian E-Wallet



Gambar 2. 1 Dompot Digital

Dompot digital atau *E-Wallet* adalah layanan elektronik yang berperan dalam proses penyimpanan data sekaligus sebagai alternatif sistem pembayaran yang dibuat untuk memudahkan pengguna dalam melakukan transaksi.

Sebagai salah satu financial technology, *E-Wallet* digunakan untuk menyimpan uang secara digital yang dapat digunakan untuk bertransaksi baik online maupun offline dengan menggunakan kode QR. Dompot digital tidak hanya digunakan untuk melakukan transaksi online, tetapi juga dapat digunakan untuk melakukan transfer ke bank lain, bahkan antar bank.

Pembayaran dompet digital telah populer dan paling diterima sebagai metode pembayaran yang muncul di negara maju dan berkembang. Dompot digital terus tumbuh dan mempengaruhi banyak faktor seperti peningkatan penyebaran, penetrasi seluler, inklusi keuangan, lebih nyaman, lebih cepat, dan lebih ekonomis. Pembayaran secara digital seperti ini mempunyai fungsi yang penting dalam pembentukan ekosistem mulai dari pengaturan, institusi keuangan, manufaktur alat, pengecer atau penjual, hingga konsumennya sendiri. (Karsen dkk., 2019)

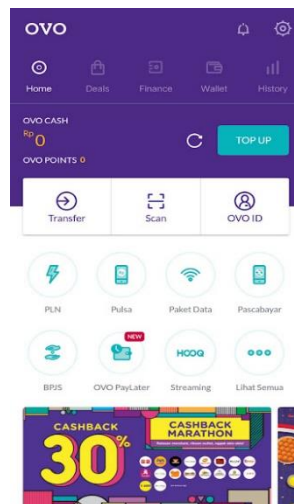
Data yang dirilis pada Agustus 2020, GoPay dan OVO masih dominan sebagai aplikasi dompet digital. DANA naik posisi menggantikan Link Aja. Pengguna aktif dompet digital sebagai pengguna aktif masih didominasi oleh GoPay. Di Indonesia terdapat 38 dompet digital yang diakui oleh Bank Indonesia

secara resmi. Transaksi dompet digital di Indonesia mencapai angka US\$ 1,5 miliar pada tahun 2018 dan akan terus meningkat menjadi US\$ 25 miliar pada tahun 2023. Sementara, GoPay merupakan aplikasi berupa uang elektronik yang dapat dijadikan kegiatan transaksi keuangan melalui aplikasi gojek dan dapat digunakan oleh customer, maupun driver itu sendiri. GoPay dapat digunakan untuk bertransaksi secara online maupun secara langsung di tenant-tenant yang sudah bekerja sama dengan GoJek. (Aulia, 2020)

2.5.2 Jenis-Jenis Dompet Digital

Berikut 5 jenis dompet digital yang paling populer di Indonesia:

1. OVO



Gambar 2. 2 Tampilan OVO

Salah satu jenis *E-Wallet* yang populer di Indonesia adalah OVO yang diklaim telah menguasai lebih dari 70 persen transaksi dan pusat perbelanjaan di Indonesia. Awalnya, OVO dimiliki oleh Grup Lippo, namun kini telah bermitra dengan Grab. Dompet digital ini juga memiliki beragam fitur menarik mulai dari belanja online, transportasi, delivery order, transfer bank, dan lain sebagainya.

2. GoPay

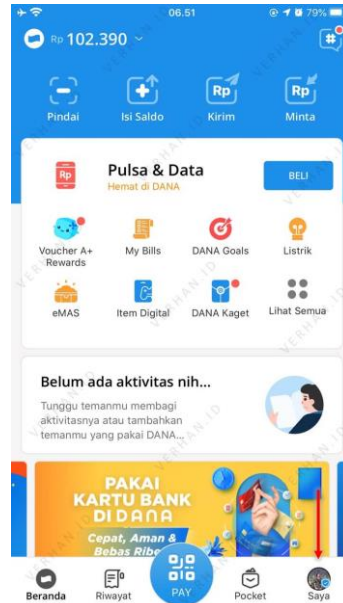


Gambar 2. 3 Tampilan GoPay

GoPay yang merupakan layanan di bawah GoJek juga menjadi salah satu *E-Wallet* terpopuler di Indonesia. Dompot digital ini telah mendapat izin dari Bank Indonesia, sehingga setiap pengguna tidak perlu lagi khawatir atau ragu akan keamanannya.

GoPay bisa digunakan untuk membayar transaksi transportasi, pesan makanan, bayar tagihan, isi pulsa, investasi, bayar jasa pengiriman, bahkan ada penawaran promo, diskon, dan cashback bagi pengguna yang sering menggunakannya.

3. DANA



Gambar 2. 4 Tampilan DANA

Dana juga merupakan *E-Wallet* andal lainnya. Dompot digital buatan anak Indonesia ini telah mendapat izin dari Bank Indonesia sebagai Lembaga Keuangan Digital, bahkan Dananya terkoneksi dengan data dari Dinas Kependudukan dan Catatan Sipil sehingga waktu pengesahannya lebih cepat dari dompet digital lainnya.

Menariknya, Dana juga terhubung dengan ATM Bersama dan BPJS yang dapat digunakan untuk bertransaksi dengan memindai kode QR, mentransfer uang, dan sebagainya dengan sangat mudah.

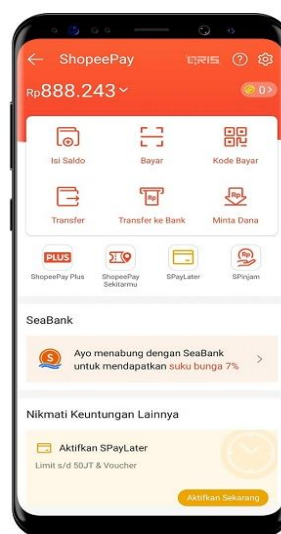
4. LinkAja



Gambar 2. 5 Tampilan LinkAja

LinkAja merupakan produk bank digital hasil sinergi beberapa perusahaan besar di Indonesia yaitu Asosiasi Bank BUMN, Pertamina dan Telkomsel. Karena banyaknya sinergi yang ada pada produk ini tentunya mendukung fitur-fitur yang dihadirkan untuk kenyamanan transaksi digital bagi para penggunaanya.

5. ShopeePay



Gambar 2. 6 Tampilan ShopeePay

Shopee juga telah meluncurkan dompet digital yaitu ShopeePay. Pertumbuhan dompet digital ini cukup pesat dibandingkan dengan dompet digital lainnya, karena ShopeePay terhubung langsung dengan Shopee sebagai marketplace. Menariknya, ShopeePay tidak hanya bisa digunakan untuk membayar tagihan atau berbelanja di Shopee, tapi juga bisa digunakan untuk membayar ShopeeFood dan lain sebagainya.

2.6 Android



Gambar 2. 7 Android

Ikon Android menggambarkan robot hijau bercahaya itu memiliki sepasang tangan dan kaki. Dalam bahasa Inggris istilah android berarti "Robot mirip manusia". Android sendiri adalah sebuah sistem operasi berbasis linux yang berfungsi sebagai penghubung antara pengguna dengan perangkat keras pada perangkat elektronik tertentu agar pengguna dapat berinteraksi dan menjalankan berbagai aplikasi seluler. (Mia Nurjannah, 2021)

2.6.1 Sejarah Android

Android lahir pertama kali pada bulan Oktober 2003 dengan penggagasnya Andy Rubin, Rich Miner, Nick Sears dan Chris White di bawah naungan perusahaan Android Inc. yang mana di Palo Alto, California. Kemudian pada tahun 2005 Android diakuisisi oleh Google. Tahun 2007 tepatnya di bulan November Android meluncurkan SDK atau Software Development Kit.

2.6.2 Versi Android

Android terus memperbarui sistem operasinya. Berbagai versi android yang telah dirilis dengan nama yang berbeda untuk setiap versi. versi android dapat dilihat pada gambar berikut:



Gambar 2. 8 Versi Android

2.7 Android Studio

2.7.1 Pengertian Android Studio



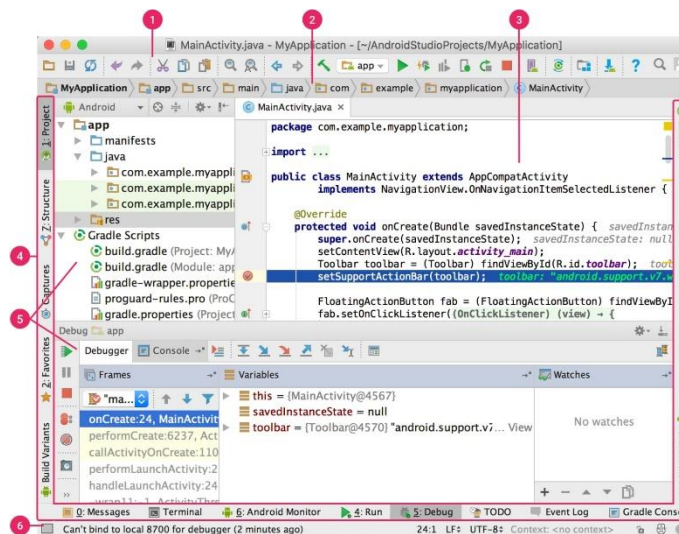
Gambar 2. 9 Android Studio

Android Studio adalah lingkungan pengembangan terintegrasi (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain menjadi editor kode dan alat pengembang IntelliJ yang andal, Dengan menggunakan Android Studio proses menciptakan aplikasi anroid menjadi lebih mudah. Itu karena banyaknya fitur-fitur yang tersedia, salah satu diantaranya ialah lingkungan yang menyatu untuk mengembangkan aplikasi android bagi semua perangkat android (Smartphone, Tablet, Smarttv, dan Smartwachth). (Syach Nuralam dkk., t.t.)

Android Studio sendiri sudah mendukung beberapa bahasa pemrograman untuk mengembangkan aplikasi android. Beberapa bahasa pemrograman yang bisa kita pakai untuk membuat aplikasi android dan support menggunakan IDE Android Studio yaitu Java, Kotlin, dan C++.

2.7.2 Antarmuka pengguna

Jendela utama Android Studio terdiri dari beberapa area logis yang ditunjukkan pada gambar di bawah ini.



Gambar 2. 10 Interface Android Studio

1. Toolbar: Melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.
2. Menu navigasi: Buka proyek Anda dan buka file untuk diedit. Menu ini memberikan tampilan yang lebih ringkas dari struktur yang terlihat di jendela Proyek.
3. Jendela editor: Membuat dan mengedit kode. Bergantung pada jenis file yang ada, editor ini dapat berubah. Misalnya, saat melihat file tata letak, editor menampilkan Editor Tata Letak.
4. Kolom jendela alat: Gunakan tombol di bagian luar jendela IDE untuk memperluas atau menciutkan setiap jendela alat.
5. Jendela alat: Akses tugas tertentu seperti manajemen proyek, penelusuran, kontrol versi, dan lainnya. Anda dapat meluaskan dan menciutkan jendela ini.

6. Status Bar: Menampilkan status proyek Anda dan IDE itu sendiri, serta peringatan atau pesan apa pun.

2.7 Bahasa Pemrograman Java



Gambar 2. 11 Java

Java adalah bahasa pemrograman yang dapat dijalankan diberbagai perangkat komputer, termasuk pada ponsel. Dikembangkan oleh Sun Microsystem dan dirilis pada 1995. Dalam pembuatan aplikasi, bahasa pemrograman Java memiliki banyak fungsi sehingga membuatnya lebih sering digunakan oleh para programmer. Beberapa di antaranya adalah daftar library yang lengkap, dapat digunakan di berbagai objek, dan bahasa yang relatif sederhana. Bahasa Java juga memiliki banyak kelebihan yang salah satunya ialah dapat dilakukan dalam berbagai perangkat (multiplatform) yang memudahkan pengembang. Namun, Java juga memiliki kekurangan yang mengganggu pengembang dalam melakukan proses pembuatan aplikasi salah satunya ialah Java memakan memori lebih besar.(Syach Nuralam dkk., t.t.)

2.8 JDK



Gambar 2. 12 JDK

Java Development Kit (JDK) adalah seperangkat perangkat lunak yang dapat Anda gunakan untuk mengembangkan perangkat lunak berbasis Java, sedangkan

JRE adalah implementasi dari Java Virtual Machine yang sebenarnya digunakan untuk menjalankan program java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lainnya seperti sumber kompiler java, bundling, debugger, pustaka pengembangan, dan sebagainya.

2.9 Basis Data

Dalam istilah sederhana *Database* atau dalam bahasa Indonesia disebut database dapat dianggap sebagai organisasi data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Basis data juga dapat diartikan sebagai kumpulan dari banyak file yang dimiliki keterkaitan antara satu file dengan file lainnya sehingga membentuk suatu bangunan data yang dapat menginformasikan perusahaan atau instansi dalam batas-batasnya yakin. Jika ada file yang tidak digabungkan atau ditautkan file lain berarti file tersebut bukan grup dari satu Database, file akan dapat membentuk database sendiri.

Database Bisa didefinisikan sebagai sesuatu membentuk penyimpanan dari informasi terpusat sehingga Informasi mudah ditemukan, dikelola, dan digunakan kembali. Database biasanya terdiri dari baris dan kolom, yang disebut baris dalam database dengan record dan kolom dalam database disebut field. (Teguh Kurniawan, 2021)

2.10 Firebase

Firebase adalah penyedia layanan cloud dengan back-end lengkap sebagai layanan yang divisi perusahaannya berlokasi di San Francisco, California. Firebase membuat sejumlah produk atau layanan untuk pengembangan aplikasi seluler atau web. Firebase didirikan oleh Andrew Lee dan James Tamplin pada tahun pada tahun 2011 dan meluncurkan basis data cloud waktu nyata pada tahun 2012. Perusahaan ini diakuisisi oleh Google pada Oktober 2014. (Teguh Kurniawan, 2021).

Firebase adalah penyedia database realtime dan layanan backend melayani. Aplikasi yang memungkinkan pengembang untuk membuat API (Application Antarmuka Pemrograman) untuk disinkronkan untuk klien yang berbeda dan disimpan di cloud Firebase. Firebase memiliki banyak library yang mungkin untuk mengintegrasikan melayani ini dengan Android, iOS, Javascript, Java, Objective-C

dan Node.JS. Beberapa kemampuan Firebase Lain-lain yang ditawarkan adalah sebagai berikut:

1. Responsif bahkan saat offline. Database Realtime dengan SDK untuk menyimpan data ke disk lokal. Jadi saat offline pengguna bisa melakukan akses data.
2. Firebase memiliki tingkat keamanan yang tinggi, karena dimiliki langsung oleh Google.
3. Firebase menyediakan layanan gratis untuk pengembang aplikasi.

2.10.1 *Firebase Realtime database*

Firebase Realtime database adalah database realtime disimpan di cloud dan dukungan multiplatform seperti Android, iOS dan Web. Data aktif firebase akan disimpan dalam struktur JSON (Java Notasi Objek Skrip). Database Firebase akan sinkronisasi secara otomatis terhadap aplikasi klien yang terhubung dengannya. Aplikasi multiplatform menggunakan SDK Android, iOS dan JavaScript akan menerima pembaruan data terbaru secara otomatis ketika aplikasi terhubung ke server pangkalan api.(Firman Maulana, 2020).

Firebase Realtime database adalah sebuah platform Database digunakan dalam aplikasi realtime. Kapan terjadi perubahan data, maka aplikasi terhubung dengan firebase itu akan diperbarui secara otomatis melalui setiap perangkat (device) baik website maupun seluler. Firebase memiliki perpustakaan itu lengkap untuk sebagian besar platform web dan seluler. Firebase dapat digabungkan dengan framework lainnya seperti node, java, javascript, dan lainnya. Ada beberapa fitur yang disediakan oleh firebase adalah sebagai berikut:

1. Analytics, fitur ini digunakan untuk mengamati perilaku pengguna dalam menggunakan aplikasi dan ditampilkan dalam satu dasbor.
2. Kembangkan, fitur ini berupa cloud messaging, autentikasi, database waktu nyata, penyimpanan, hosting, testlab dan pelaporan kerusakan.
3. Grow, fitur ini digunakan untuk publish sebuah produk aplikasi. Berikut adalah beberapa fitur firebase.



Gambar 2. 13 Fitur Firebase

2.11 Balsamiq Mockup

Balsamiq Mockup adalah aplikasi wireframing yang digunakan untuk membuat konsep antarmuka pengguna suatu aplikasi. Software ini sudah menyediakan tools yang dapat mempermudah untuk merancang aplikasi prototype yang akan dibuat. (Hendri & Mochammad Arief Sutisna, 2021)



Gambar 2. 14 Balsamiq Mockup

- Cara pemasangan mockup balsamiq adalah sebagai berikut:
 1. Ekstrak "Balsamiq Mockup" yang diunduh.
 2. Matikan koneksi internet dan antivirus.
 3. Install software Balsamiq Mockup seperti biasa.
 4. Perangkat lunak Balsamiq Mockup dapat digunakan.
- Fitur dan keunggulan Balsamiq Mockup adalah sebagai berikut:
 1. Membuat antarmuka pengguna dan komponen lainnya dengan mudah.
 2. Impor dan ekspor hasil proyek ke pdf atau png.
 3. Bisa digunakan secara offline beserta fitur sharing dengan pengguna lain.

4. Dapat dengan mudah menggunakan drag and drop.
5. Fitur yang lebih detail dibandingkan dengan software low fidelity lainnya.

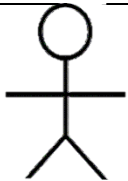
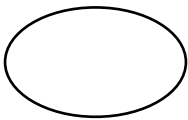
2.12 UML (Unified Modelling Language)

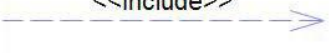
2.12.1 Pengertian UML

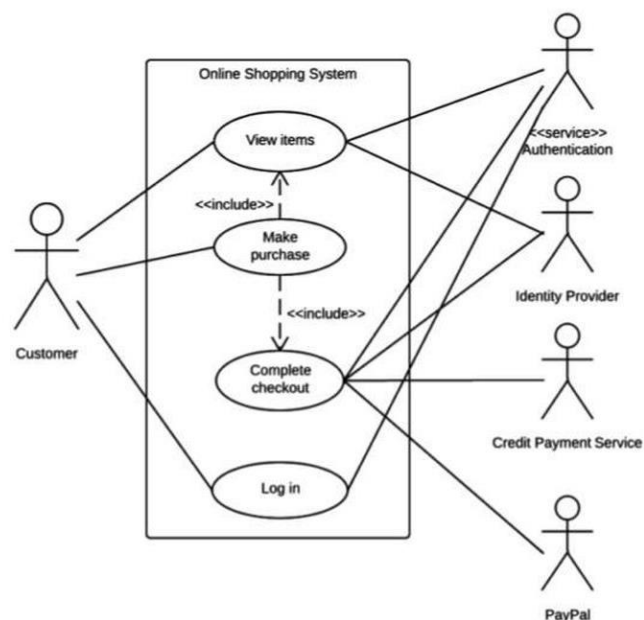
UML (Unified Modelling Language) adalah penggantian metode analisis berorientasi objek dan desain berorientasi objek (OOAD&D/object berorientasi analisis dan desain) yang muncul sekitar akhir 80an dan awal 90-an. UML adalah kombinasi dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tapi UML mencakup lebih banyak lebih luas dari OOAD. Di pertengahan waktu Pengembangan UML, dilakukan standardisasi proses dengan OMG (Grup Manajemen Objek) dengan Saya berharap UML akan menjadi bahasa standar pemodelan masa depan (mis sekarang banyak digunakan oleh berbagai lingkaran).(Feby Prasetya & Lestari Dewi Putri, 2022)

2.12.2 Use Case Diagram

Use case diagram adalah salah satu dari beberapa jenis diagram UML (Unified Modeling Language) yang menjelaskan hubungan interaksi antara sistem dan aktor. Use Case dapat menggambarkan jenis interaksi antar pengguna sistem dengan sistem. Simbol pada *Use Case Diagram* dapat dilihat pada tabel berikut ini:

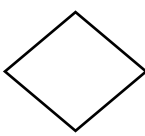
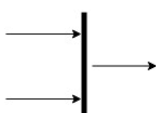
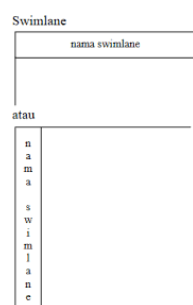
No.	Simbol	Keterangan
1.		Aktor Simbol ini mewakili set peran pengguna saat menghubungkan atau interaksi dengan kasus penggunaan.
2.		<i>Use Case</i> Deskripsi interaksi atau hubungan antara aktor dengan sistem.

3		Asosiasi Penghubung antara objek dengan objek lainnya.
4.		<i>Ektend</i> Menjelaskan jika use case memiliki kondisi terpenuhi maka use case lainnya adalah tambahan fungsional
5.		<i>Include</i> Menjelaskan jika use case sudah selesai adalah fungsionalitas dari use case lain
6.		Generalisasi Menggambarkan spesialisasi aktor untuk dapat berpartisipasi dengan use case.

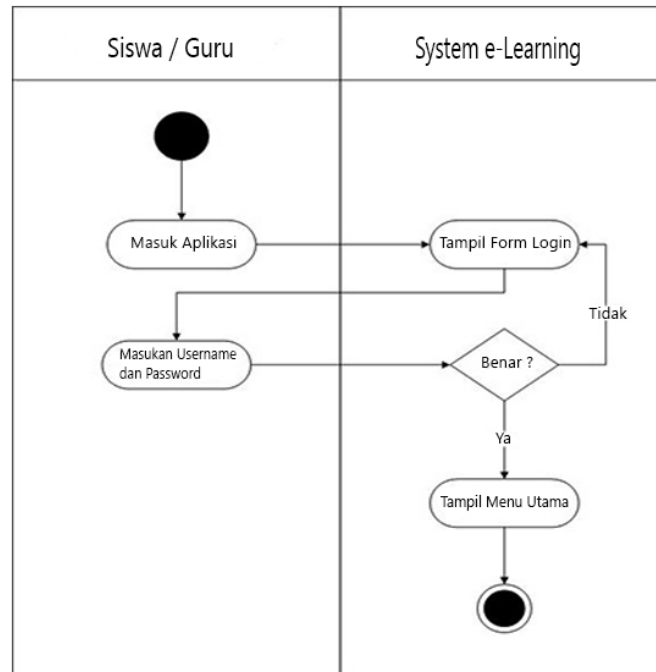
Tabel 2 Simbol *Use Case Diagram*Gambar 2. 15 Contoh Pemanfaatan *Use Case Diagram*

2.12.3 Activity Diagram

Activity Diagram merupakan penggambaran alur kerja (workflow) dan juga aktivitas lain dari suatu sistem atau proses bisnis yang ada pada perangkat tersebut perangkat lunak.

No.	Simbol	Keterangan
1.		Node Awal Keadaan awal suatu objek pada sistem.
2.		Aktivitas Aktivitas satu sistem agar menjadi berinteraksi satu sama lain.
3.		Percabangan / <i>Decision</i> Cabang yang digunakan jika ada tindakan atau apa pun keputusan lebih dari satu akan diambil.
4.		Penggabungan / <i>Join</i> Menggabungkan lebih dari satu aktivitas.
5.		<i>Final</i> Keadaan akhir suatu objek pada sistem
6.		<i>Swimlane</i> Tunjukkan organisasi penjamin emisi bertanggung jawab atas aktivitas masa lalu terjadi.

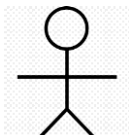
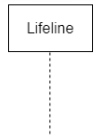
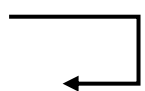
Tabel 3 Simbol Activity Diagram



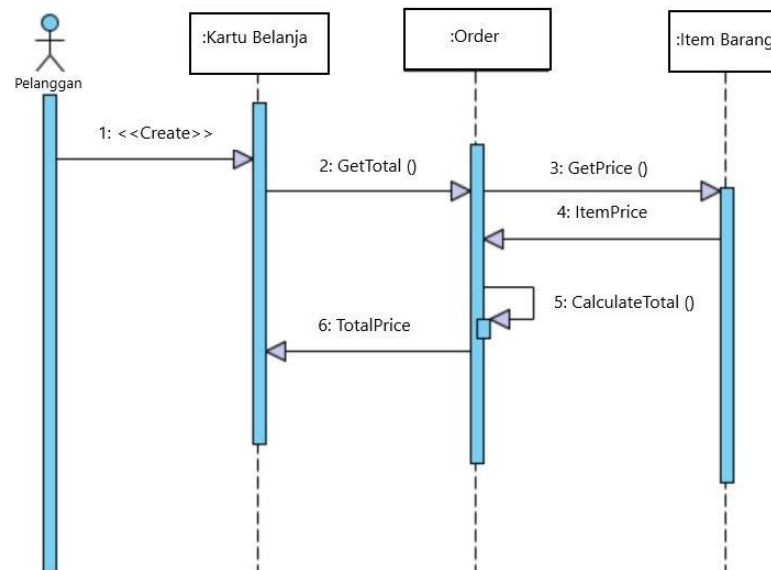
Gambar 2. 16 Contoh Pemanfaatan *Activity Diagram*

2.12.4 *Sequence Diagram*

Sequence Diagram menggambarkan aktivitas antara suatu objek dengan objek lain melalui pesan yang dikirim dan diterima antar objek.

No.	Simbol	Keterangan
1.		<i>Actor</i> Pengguna yang berinteraksi dengan sistem.
2.		<i>LifeLine</i> Antarmuka untuk awal dan akhir pesan.
3.		<i>Message</i> Sebuah pesan dari satu objek ke objek lain.
4.		<i>Message to self</i> Aktivitas atau pesan yang berisi aktivitas diri.

Tabel 4 Simbol *Sequence Diagram*



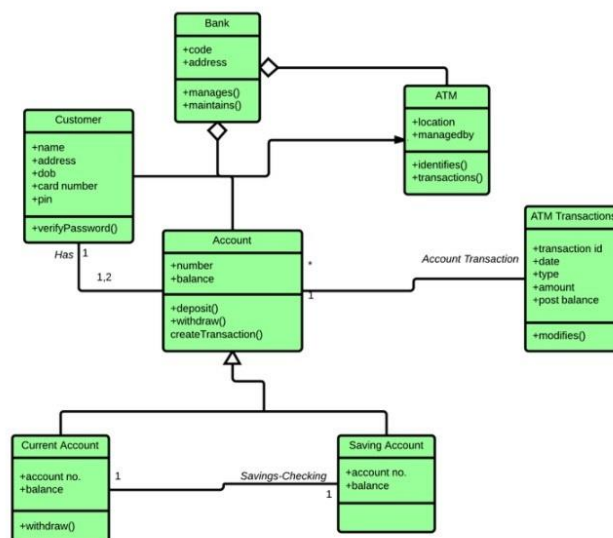
Gambar 2. 17 Contoh Pemanfaatan Sequence Diagram

2.12.4 Class Diagram

Class Diagram digunakan untuk menggambarkan hubungan antar kelas dalam model desain suatu sistem, perbedaannya cukup mendasar antara satu kelas dengan kelas lainnya, dan juga menjelaskan di mana subsistem itu berada kelas itu. Diagram kelas menunjukkan tanggung jawab entitas yang akan menentukan perilaku sistem.

No.	Simbol	Keterangan
1.		<i>Generalization</i> Ini adalah relasi atau hubungan antar kelas memiliki arti umum.
2.		<i>Collaboration</i> Deskripsi urutan tindakan ditampilkan oleh sistem yang menghasilkan suatu hasil yang terukur bagi seorang aktor.
3.		<i>Class</i> <i>Class</i> dalam struktur sistem.

4.		<i>Realization</i> Suatu kelas harus diwajibkan untuk mengikuti aturan yang telah ditetapkan kelas lain karena sudah memiliki hubungan.
5.		<i>Dependency</i> Kelas relasi yang memiliki ketergantungan dengan kelas lain karena tidak memiliki elemen mandiri.
6.		<i>Association</i> Hubungan antara satu kelas dengan kelas lainnya.

Tabel 5 Simbol *Class Diagram*

Gambar 2. 18 Contoh Pemanfaatan Class Diagram

2.13 Flowchart

Flowchart adalah bagian dengan simbol (kode) tertentu yang menjelaskan dan menggambarkan langkah-langkah proses secara rinci, dan hubungan antara proses (metode) dengan proses lain dalam suatu program.

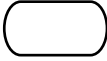
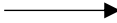
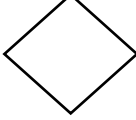
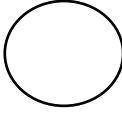
Flowchart terbagi menjadi 5 jenis, diantaranya:

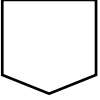
1. *Flowchart* Sistem (System *Flowchart*)

2. *Flowchart* Dokumen
3. *Flowchart* Skema
4. Diagram Program (*Flowchart* Program)
5. *Flowchart* Proses (*Process Flowchart*)

➤ Simbol dan Fungsi *Flowchart*

Berikut ini adalah beberapa simbol yang digunakan dalam menggambar *Flowchart* pada tabel di bawah ini.

Simbol	Nama	Fungsi
	Terminator	Permulaan/akhir program
	Garis alir	Arah aliran program
	Preparation	Proses ini sialisasi (pemberian harga awal)
	Proses	Proses perhitungan (pengolahan data).
	Input/output data	Proses input/output data, parameter, informasi.
	Predefined Process (sub program)	Permulaan sub program/ proses menjalankan sub program.
	Decision	Perbandingan pernyataan, penyeleksian data yang memberikan pilhan untuk langkah selanjutnya.
	On Page Connector	Penghubung bagian-bagian <i>Flowchart</i> yang berada pada satu halaman.

	Off Page Connector	Penghubung bagian-bagian <i>Flowchart</i> yang berada pada halaman berbeda.
---	-----------------------	---

Tabel 6 Simbol *Flowchart*

2.14 *Black Box*

2.14.1 Pengertian *Black Box*

Black Box adalah kebalikan dari *White box* yang dimana Software tidak memiliki akses ke dalam mengetahui implementasi atau source code dari program tersebut untuk mencari tahu apakah ada kesalahan dalam program dan tidak diharuskan memiliki pengetahuan menjadi seorang programming dan implementasinya. Ketika menjalankan *Black Box* maka akan bertemu dengan user *Interface* kemudian memeriksa output dan menyediakan input, juga akan menguji function-function atau menguji performa program yang tidak bekerja dengan benar.

2.14.2 Kelebihan *Black Box*

1. Efisien dan cocok untuk kode sumber skala besar.
2. Menguji program dilihat dari sudut pandang pengguna.
3. Mampu menggunakan perangkat lunak dalam jumlah banyak dalam pengujian program seperti itu jadi tidak harus memiliki pengetahuan seperti seorang pemrogram.

2.14.3 Kekurangan *Black Box*

1. Perangkat lunak hanya mampu menjalankan beberapa skenario tes yang akan dipilih nanti.
2. Pengujian yang salah karena penguji Perangkat Lunak memiliki pengetahuan yang terbatas tentang program yang digunakan.
3. Pengujian yang tidak tepat karena perangkat lunak yang tidak dimiliki persetujuan penuh terhadap kode sumber yang dipilih.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Masalah

Analisis masalah adalah penguraian dari suatu masalah yang utuh menjadi bagian-bagian komponennya untuk tujuan mengidentifikasi dan mengevaluasi masalah, peluang, hambatan yang terjadi dan kebutuhan yang terjadi diharapkan dapat diusulkan perbaikannya.

Karena semakin meningkatnya penggunaan dompet digital dalam kehidupan sehari-hari dan banyaknya aplikasi yang beredar, hal ini menimbulkan permasalahan dalam menentukan aplikasi *E-Wallet* mana yang paling efektif untuk digunakan. Untuk mengatasi permasalahan tersebut maka dilakukan penelitian menggunakan sistem pendukung keputusan untuk menentukan alternatif *E-Wallet* dengan mengetahui faktor-faktor apa saja yang mendukung penggunaan *E-Wallet*, sehingga dapat menentukan aplikasi mana yang ideal untuk digunakan oleh banyak orang. Salah satu solusi untuk membantu kesulitan masyarakat adalah dengan membuat sistem pendukung keputusan.

Oleh karena itu berdasarkan analisis permasalahan yang terjadi, maka melalui Sistem ini diharapkan dapat menjadi alternatif pilihan masyarakat dalam melakukan pemilihan *E-Wallet*.

3.2 Analisa Kebutuhan

3.2.1 Analisa Kebutuhan Perangkat Keras

Berikut ini adalah perangkat keras yang digunakan untuk menerapkan sistem rekomendasi ini adalah sebagai berikut:

- 1) Processor Intel Celeron 1.10 GHz
- 2) RAM 4 GB

3.2.2 Analisa Kebutuhan Perangkat Lunak

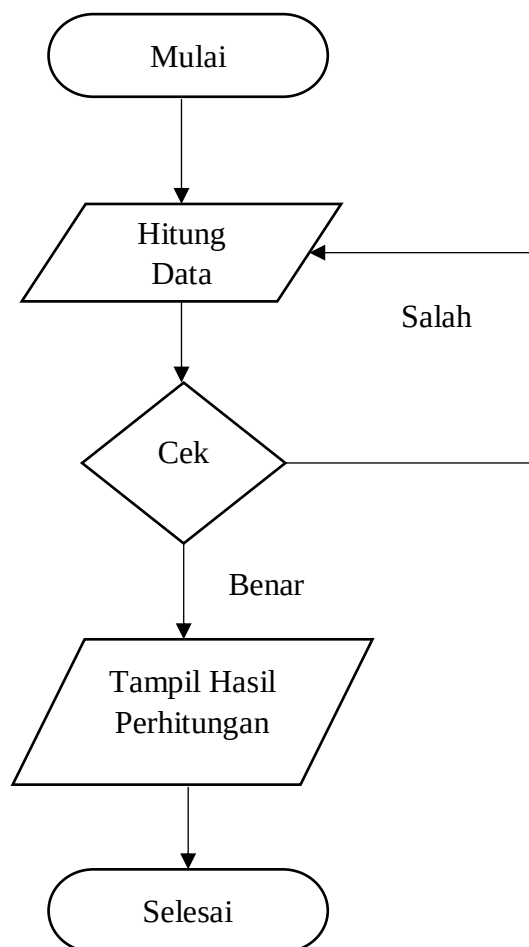
Perangkat Lunak yang digunakan pada penelitian ini adalah sebagai berikut:

- 1) Android Studio
- 2) Android JDK
- 3) Microsoft Word 2010
- 4) Notepad++
- 5) Balsamiq Mockup
- 6) Visual Paradigm

3.3 Tujuan Perancangan Sistem

Perancangan sistem ini bertujuan untuk membuat sistem pendukung keputusan pemilihan dompet digital berbasis android agar proses pemilihan dapat dilakukan lebih cepat dan dapat mengurangi kesalahan serta dapat memberikan alternatif terbaik dalam pengambilan keputusan.

3.4 Flowchart Perancangan Sistem



Gambar 3. 1 *Flowchart* Perancangan Sistem

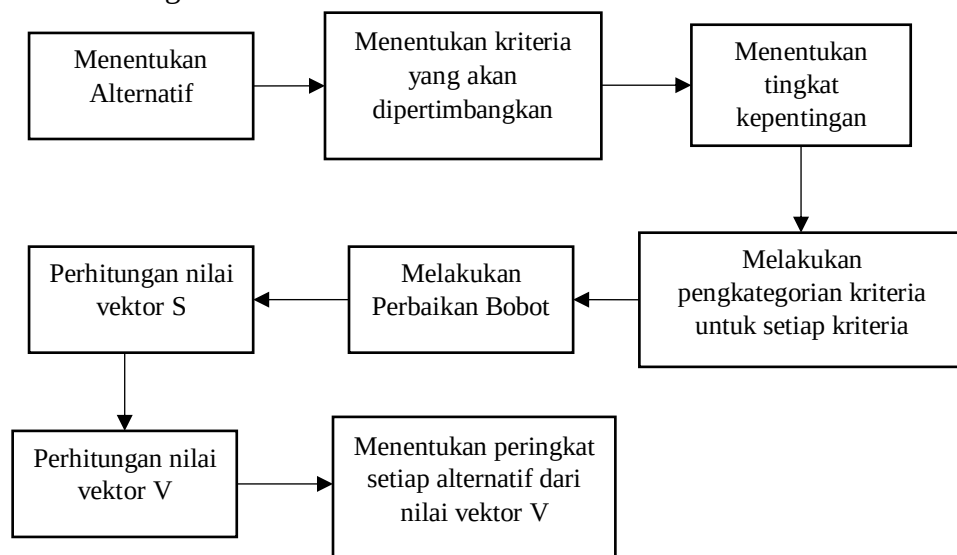
3.5 Perancangan Sistem

3.5.1 Perancangan Sistem Pendukung Keputusan

Pada tahap perancangan ini bertujuan untuk mendefinisikan alternatif-alternatif yang ada, mulai dari menentukan kriteria yang akan dipertimbangkan hingga menentukan bobot kriteria, juga menjelaskan cara perhitungan dengan menggunakan metode *weighted product*. Berikut adalah perancangan sistem pendukung keputusan pemilihan *E-Wallet* dengan menggunakan metode *weighted product* berbasis Android.

1) Perancangan Model *Weighted Product*

Tahapan rancangan model *weighted product* (wp) pada penelitian ini adalah sebagai berikut:



Tabel 7 Perancangan Model *Weighted Product*

Sistem pendukung keputusan pemilihan *E-Wallet* dengan metode *weighted product*, alternatif-alternatif yang digunakan adalah sebagai berikut:

- OVO
- GoPay
- DANA
- LinkAja
- ShopeePay

- b) Kriteria yang digunakan dalam pengambilan keputusan pemilihan *E-Wallet* ini adalah Biaya Admin, Kemudahan Top Up, Promo, Fitur dan Layanan.
- c) Sistem Pendukung Keputusan dalam penelitian ini memberikan bobot kepentingan pada kriteria yaitu:

Bobot	Kepentingan
1	Tidak Penting
2	Kurang Penting
3	Cukup Penting
4	Penting
5	Sangat Penting

Tabel 9 Bobot Kepentingan

No.	Kriteria	Bobot
1.	Biaya Admin	5
2.	Kemudahan Top Up	4
3.	Promo	4
4.	Fitur	4
5.	Layanan	3

Tabel 8 Bobot Kriteria

- d) Mengkategorikan kriteria untuk setiap kriteria yang digunakan. Kriteria yang bersifat keuntungan bernilai positif dan kriteria yang bersifat biaya bernilai negatif.
- Kriteria Biaya Admin merupakan kriteria yang bernilai negatif. Semakin besar nilai kriteria ini maka akan menjadi tidak baik.
 - Kriteria Kemudahan Top Up, Promo, Fitur dan Layanan merupakan kriteria yang bernilai positif. Maka jika semakin besar nilainya akan semakin baik.
- e) Melakukan perbaikan bobot untuk setiap bobot kriteria adalah sebagai berikut:

No.	Kriteria	Bobot	Perbaikan Bobot
1.	Biaya Admin	5	$5/20 = 0.25$
2.	Kemudahan Top Up	4	$4/20 = 0.2$
3.	Fitur	4	$4/20 = 0.2$
4.	Promo	4	$4/20 = 0.2$
5.	Layanan	3	$3/20 = 0.15$
Jumlah		20	1

Tabel 10 Perbaikan Bobot Setiap Kriteria

Karena Biaya Admin adalah kriteria yang bernilai negatif, maka nilai perbaikan bobotnya adalah negatif.

f) Perhitungan nilai vektor S dengan rumus:

$$S_i = \prod_{j=1}^n X_{ij}^{w_j}$$

Keterangan:

S_i = hasil normalisasi keputusan pada alternatif ke-i

X_{ij} = rating alternatif per atribut

i = alternatif

j = atribut

n = banyaknya kriteria

g) Nilai V yang terbesar adalah alternatif terbaik, dengan rumus sebagai berikut:

$$V_i = \frac{\prod_{j=1}^n X_{ij}^{w_j}}{\prod_{j=1}^n X_{ij} * W_j}$$

Dimana:

V_i = preferensi alternatif

n = banyaknya kriteria

X_{ij} = rating alternatif per atribut

W_j = bobot awal

2) Perhitungan Metode *Weighted Product*

WP (*Weighted Product*) adalah analisis multi keputusan kriteria yang sangat terkenal (MCDA) atau metode pengambilan keputusan multi kriteria. Disini penulis akan membahas tentang perhitungan metode *Weighted Product* untuk menentukan pemilihan *E-Wallet*. Berikut contoh penghitungan *Weighted Product*.

1. Menentukan kriteria yang dijadikan acuan dalam pengambilan keputusan. 5 kriteria yang digunakan untuk melakukan penilaian adalah:

C1 = Biaya Admin

C2 = Kemudahan Top Up

C3 = Promo

C4 = Fitur

C5 = Layanan

Adapun alternatif yang digunakan adalah sebagai berikut:

R1 = OVO

R2 = GoPay

R3 = DANA

R4 = LinkAja

R5 = ShopeePay

2. Memberikan nilai bobot pada setiap kriteria tingkat kepentingan setiap kriteria juga dinilai dengan nilai 1 sampai 5, itu adalah:

1 = Tidak Penting

2 = Kurang Penting

3 = Cukup Penting

4 = Penting

5 = Sangat Penting

3. Pengambilan keputusan memberikan bobot preferensi untuk setiap kriteria adalah: $W = (5, 4, 4, 4, 3)$

No.	Wallet	Biaya Admin	Ttopup	Jtopup	Tpromo	Jpromo	Tfitur	Jfitur	Tlayan	Jlayan
1.	OVO	1200	84	22	73	22	76	22	72	22
2.	GoPay	2000	89	22	78	22	80	22	73	22
3.	DANA	2500	101	22	84	22	90	22	78	22
4.	LinkAja	1500	74	22	64	22	63	22	70	22
5.	ShopeePay	2000	97	22	95	22	92	22	93	22

Untuk mencari rumus nilai masing-masing kriteria adalah sebagai berikut:

$$\text{Rata-rata} = \frac{\text{Tdata}}{\text{Jdata}}$$

Keterangan:

Tdata: Total data dari setiap kriteria

Jdata: Jumlah data (responden)

No.	Alternatif	Kriteria				
		C1	C2	C3	C4	C5
1.	OVO	1200	84/22=3,82	73/22=3,32	76/22=3,45	72/22 = 3,27
2.	GoPay	2000	89/22=4,05	78/22=3,55	80/22=3,64	73/22=3,32
3.	DANA	2500	101/22=4,59	84/22=3,82	90/22=4,09	78/22=3,55
4.	LinkAja	1500	74/22=3,36	64/22=2,91	63/22=2,86	70/22=3,18
5.	ShopeePay	2000	97/22=4,41	95/22=4,32	92/22=4,18	93/22=4,23

Tabel 11 Nilai alternatif setiap kriteria

4. Melakukan normalisasi Bobot

$$W = (5, 4, 4, 4, 3)$$

Maka perbaikan bobotnya adalah:

$$W_1 = 5 / (5+4+4+4+3) = 5/20 = 0,25$$

$$W_2 = 4 / (5+4+4+4+3) = 4/20 = 0,2$$

$$W_3 = 4 / (5+4+4+4+3) = 4/20 = 0,2$$

$$W_4 = 4 / (5+4+4+4+3) = 4/20 = 0,2$$

$$W_5 = 3 / (5+4+4+4+3) = 3/20 = 0,15$$

$$W_1 + W_2 + W_3 + W_4 + W_5$$

$$= 0,25 + 0,2 + 0,2 + 0,2 + 0,15 = 1$$

$$\Sigma = 1$$

Pangkat	-0,25	0,2	0,2	0,2	0,15
----------------	-------	-----	-----	-----	------

Karena Biaya Admin adalah kriteria yang bernilai negatif, maka nilai perbaikan bobotnya adalah negatif.

5. Menghitung Nilai Vektor S

$$S_1 = (1200^{-0.25}) (3,82^{0.2}) (3,32^{0.2}) (3,45^{0.2}) (3,27^{0.15})$$

$$= 0,4321$$

$$S_2 = (2000^{-0.25}) (4,05^{0.2}) (3,55^{0.2}) (3,64^{0.2}) (3,32^{0.15})$$

$$= 0,3951$$

$$S_3 = (2500^{-0.25}) (4,59^{0.2}) (3,82^{0.2}) (4,09^{0.2}) (3,55^{0.15})$$

$$= 0,4019$$

$$S_4 = (1500^{-0.25}) (3,36^{0.2}) (2,91^{0.2}) (2,86^{0.2}) (3,18^{0.15})$$

$$= 0,3721$$

$$S_5 = (2000^{-0.25}) (4,41^{0.2}) (4,32^{0.2}) (4,18^{0.2}) (4,23^{0.15})$$

$$= 0,4456$$

6. Menghitung Nilai Vektor V

$$V_1 = \frac{0,432}{(0,432) + (0,395) + (0,401) + (0,372) + (0,445)} = 0,211$$

$$V_2 = \frac{0,395}{(0,432) + (0,395) + (0,401) + (0,372) + (0,445)} = 0,193$$

$$V_3 = \frac{0,401}{(0,432) + (0,395) + (0,401) + (0,372) + (0,445)} = 0,196$$

$$V_4 = \frac{0,372}{(0,432) + (0,395) + (0,401) + (0,372) + (0,445)} = 0,181$$

$$V_5 = \frac{0,445}{(0,432) + (0,395) + (0,401) + (0,372) + (0,445)} = 0,217$$

7. Menentukan Peringkat dari Nilai Vektor V

Dengan melihat hasil di atas, nilai V_5 lebih besar dari pada nilai V lainnya, dengan urutan sebagai berikut:

Peringkat	Vektor V	Nilai
1	V_5	0,217
2	V_1	0,211
3	V_3	0,196
4	V_2	0,193
5	V_4	0,181

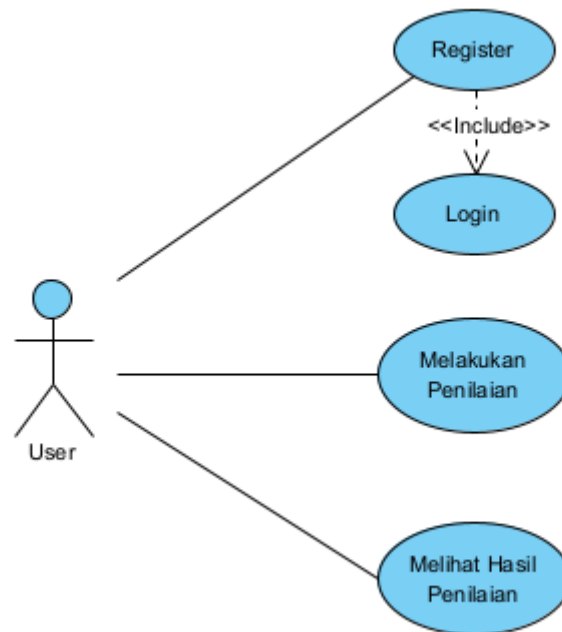
Tabel 12 Hasil Pemeringkatan

Berdasarkan pemeringkatan di atas, dapat disimpulkan bahwa nilai vektor tertinggi yaitu, V_5 (ShopeePay) dengan nilai 0,217.

3.5.2 Perancangan Sistem

3.5.2.1 Use Case Diagram

Use Case Diagram menggambarkan fungsional sistem yang dijalankan oleh User. Berikut use case diagram dalam Pemilihan *E-Wallet*:

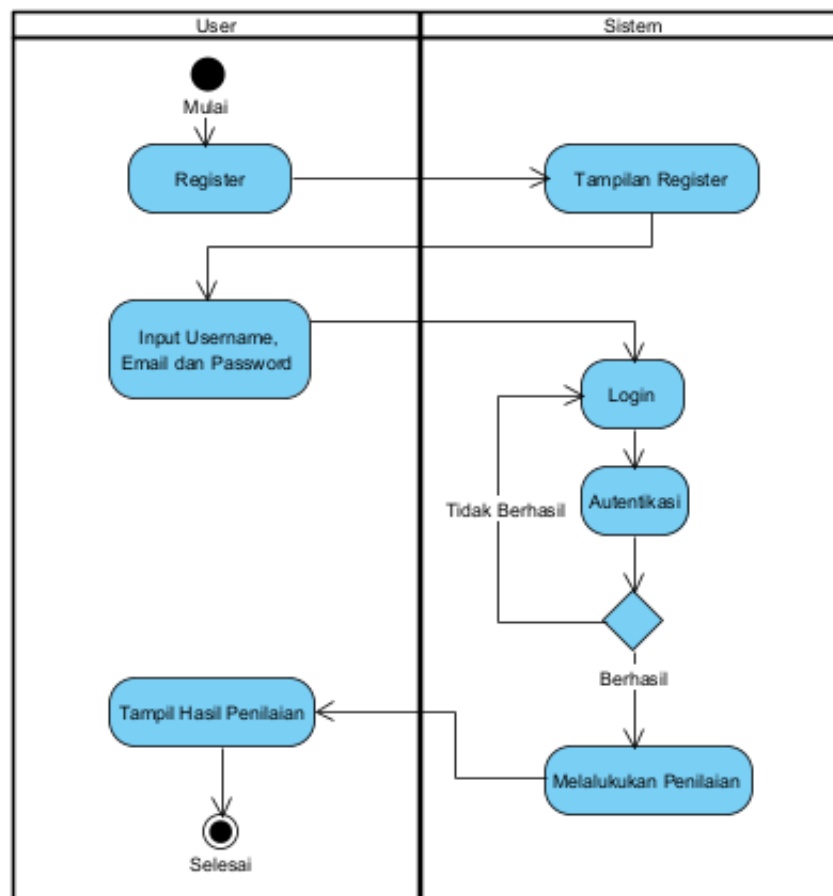


Gambar 3. 2 Use Case Diagram

Gambar diatas menunjukkan bahwa pengguna harus melakukan registrasi dan kemudian login untuk dapat mengakses sistem. Kriteria bobot use case *E-Wallet* mempunyai use case fungsional yaitu case penilaian skala kriteria. *Use Case* pemilihan *E-Wallet* menjelaskan hubungan antar pengguna yang melakukan aktivitas pemilihan *E-Wallet* dari 5 kriteria yaitu Biaya Admin, Kemudahan Top Up, Promo, Fitur dan Layanan.

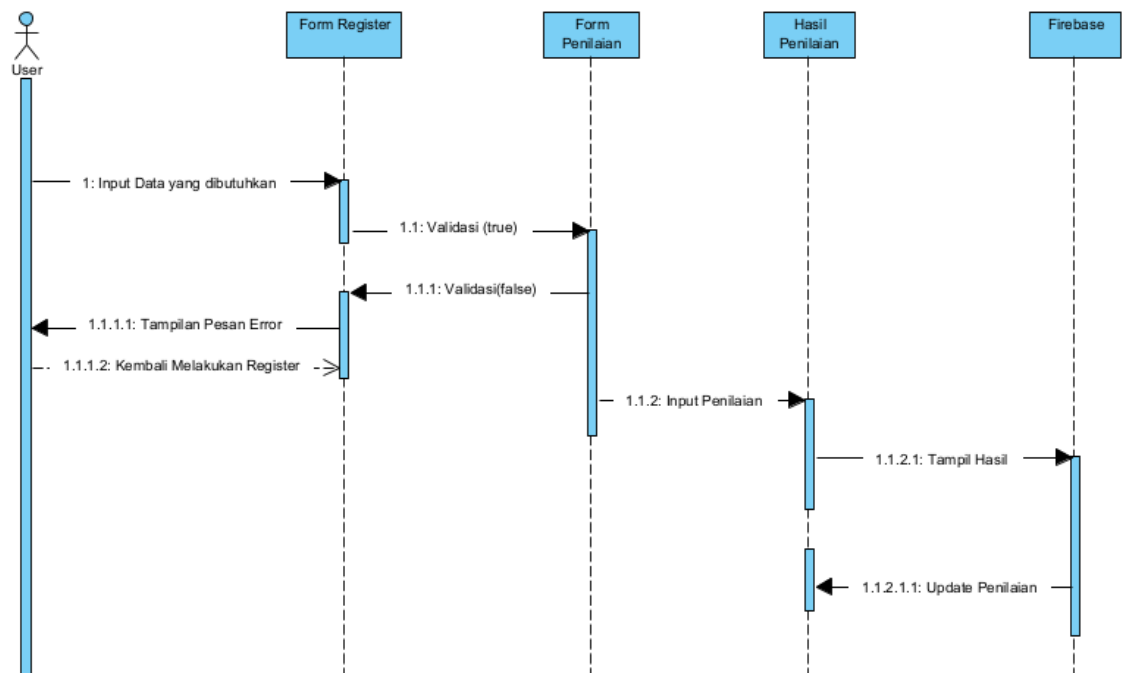
3.5.2.2 Activity Diagram

Sebelum menganalisis kriteria, hal pertama yang harus dilakukan pengguna adalah melakukan registrasi terlebih dahulu lalu login untuk masuk ke sistem dan melakukan penilaian kriteria yang ada, setelah melakukan penilaian pengguna dapat melihat hasil penilaian.



Gambar 3. 3 Activity Diagram

3.5.2.3 Sequence Diagram



Gambar 3. 4 Sequence Diagram

Gambar di atas menyajikan rangkaian pesan antar objek saat pengguna melakukan proses registrasi atau pendaftaran akun yang akan digunakan aplikasi. Diagramnya dimulai dari aktivitas penginputan data pribadi ke dalam aplikasi, maka di balik layar sistem akan melakukannya memvalidasi apakah pengguna benar-benar telah melengkapi formulir, jika validasi berjalan baik maka sistem akan meminta permintaan data ke firebase dan menjawab bahwa data pengguna telah berhasil didaftarkan di dalam database, akan tetapi jika validasi ditolak oleh sistem jadi sistem akan memberikan return/callback yang menginformasikan bahwa pengguna harus melengkapi data formulir terlebih dahulu untuk masuk ke dalam sistem.

Setelah melakukan pendaftaran, pengguna dapat menilai kriteria yang ada, kemudian setelah terisi, pengguna dapat melihat hasil penilaian yang telah dilakukan. Pengguna hanya dapat melakukan penilaian satu kali.

3.5.3 Perancangan Database

Struktur file digunakan dalam desain sistem untuk menentukan nilai atau tipe data suatu atribut dalam file yang terdapat dalam database. Secara bertahap perancangan struktur file untuk memudahkan dalam mengetahui suatu nilai atau tipe data yang terdapat pada file penyimpanan ini akan dijelaskan mengenai merancang database yang akan digunakan. Penyusunan tabel ini pada dasarnya digunakan untuk memudahkan pemasukan dengan menyimpan data sesuai dengan kelompok data atau informasi.

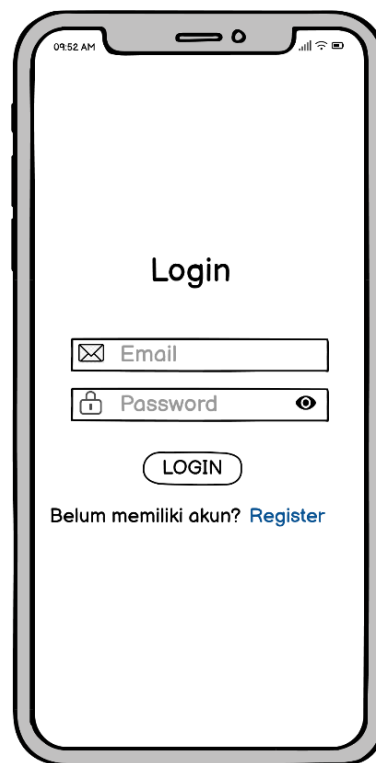
No.	Nama Field	Tipe	Keterangan
1.	Username	<i>String</i>	Nama lengkap pengguna
2.	Email	<i>String</i>	Alamat email pengguna
3.	Password	<i>String</i>	Password pengguna
4.	Biaya admin	<i>Integer</i>	Biaya admin masing-masing aplikasi <i>E-Wallet</i>
5.	Appname	<i>String</i>	Nama dari aplikasi <i>E-Wallet</i>
6.	jtopup	<i>Integer</i>	Jumlah nilai top up dari kriteria kemudahan top up
7.	jpromo	<i>Integer</i>	Jumlah nilai promo dari kriteria promo
8.	jitur	<i>Integer</i>	Jumlah nilai fitur dari kriteria fitur
9.	jlayanan	<i>Integer</i>	Jumlah nilai layanan dari kriteria layanan
10.	Poin aplikasi	<i>Integer</i>	Jumlah dari semua kriteria
11.	Rating aplikasi	<i>Integer</i>	Rating dari tiap aplikasi <i>E-Wallet</i>

Tabel 13 Perancangan Database

3.5.4 Perancangan *Interface* (Antarmuka)

Desain antarmuka adalah proses pengembangan sederhana dari sistem yang akan dibuat dalam Sistem Pendukung Keputusan Pemilihan Dompok Digital. Sedangkan untuk mendesain Antarmuka yang dirancang untuk sistem ini adalah sebagai berikut:

3.5.4.1 Rancangan Halaman Login



Gambar 3. 5 Rancangan Halaman Login

Gambar diatas merupakan gambar desain antarmuka aplikasi pemilihan E-Wallet yang menjelaskan tentang aktivitas untuk melakukan login menggunakan email. Rancangan antarmuka ini terlihat dari komponen tampilan Android seperti *Text Display*, *Text Edit*, dan *Buttons*. Setiap komponen tampilan mempunyai perannya masing-masing untuk dapat berperan dan membantu pengguna dalam proses login ke dalam aplikasi.

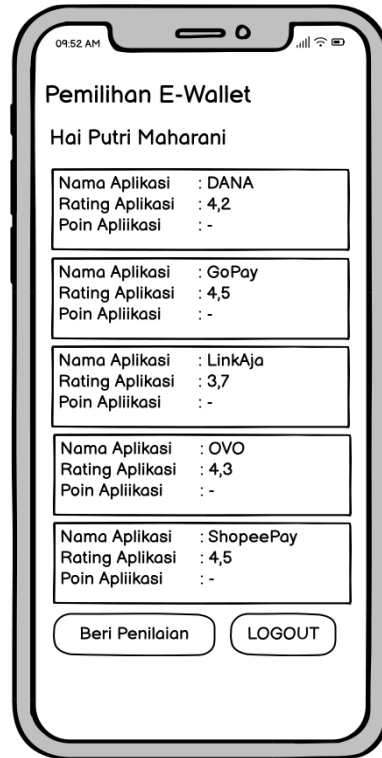
3.5.4.2 Rancangan Halaman Register



Gambar 3. 6 Rancangan Halaman Register

Pada gambar di atas merupakan gambar desain antarmuka di aplikasi pemilihan E-Wallet, dimana antarmuka tersebut menjelaskan mengenai pendaftaran untuk masuk ke dalam sistem, dimana dalam perancangan antarmuka ini pengguna diharuskan melengkapi data diri yang diperlukan dengan memasukkan data ke dalam kolom pada formulir yang telah disediakan. Adapun data pribadi yang terdapat pada desain antarmuka ini berupa username, email dan password. Komponen tampilan pada desain ini adalah *textview*, *edittext*, dan *button*.

3.5.4.3 Rancangan Halaman Utama



Gambar 3. 7 Rancangan Halaman Utama

Pada gambar di atas merupakan gambar desain antarmuka pada Aplikasi Pemilihan E-Wallet, di mana antarmuka menjelaskan tentang antarmuka halaman utama atau biasa di sebut dengan dashboard. Pada desain antarmuka halaman utama ini terdapat komponen view seperti tampilan teks dan tampilan tombol yang diatur seperti pada gambar di atas.

3.5.4.4 Rancangan Halaman Penilaian

1) Rancangan Halaman Penilaian ShopeePay

The image shows a mobile phone screen with a wireframe for an evaluation page titled "Pemilihan E-Wallet". The status bar at the top shows "09:52 AM" and signal icons. The main heading is "Pemilihan E-Wallet". Below it, the text reads "Mohon Berikan Penilaian Anda Mengenai ShopeePay". A legend for the rating scale is provided: "Keterangan : 1 - Tidak Memenuhi Kriteria, 2 - Kurang Baik, 3 - Cukup/Netral, 4 - Baik, 5 - Sangat Baik". The evaluation items are listed with their respective admin fees: "Biaya Admin : Rp 2.000", "Bagaimana Penilaian Anda tentang Kemudahan Top Up pada ShopeePay", "Bagaimana Penilaian Anda tentang Promo pada ShopeePay", "Bagaimana Penilaian Anda tentang Fitur-Fitur pada ShopeePay", and "Bagaimana Penilaian Anda tentang Layanan pada ShopeePay". Each item has a dropdown menu with the number "1" selected. At the bottom, there is a button labeled "Selanjutnya".

Gambar 3. 8 Rancangan Halaman Penilaian ShopeePay

Gambar diatas merupakan rancangan antarmuka halaman penilaian untuk *E-Wallet* ShopeePay. Pada desain antarmuka halaman penilaian ini terdapat komponen view seperti tampilan teks, *combo box* dan tampilan *button*.

The image shows a mobile application interface for 'Pemilihan E-Wallet'. At the top, the status bar shows '09:52 AM'. The app title is 'Pemilihan E-Wallet'. Below it, the text reads 'Mohon Berikan Penilaian Anda Mengenai ShopeePay'. A legend for ratings is provided: 1 - Tidak Memenuhi Kriteria, 2 - Kurang Baik, 3 - Cukup/Netral, 4 - Baik, 5 - Sangat Baik. The 'Biaya Admin' is listed as 'Rp 2.000'. The main question is 'Bagaimana Penilaian Anda tentang Keandalan Fitur-Fitur pada ShopeePay'. An alert dialog box is overlaid on the screen with the title 'SPK E-Wallet' and the message 'Apakah anda yakin ingin keluar dari kuisioner?'. The dialog has 'No' and 'Yes' buttons. Below the dialog, there are three more rating questions: 'Bagaimana Penilaian Anda tentang Fitur-Fitur pada ShopeePay', 'Bagaimana Penilaian Anda tentang Layanan pada ShopeePay', and a 'Selanjutnya' button at the bottom.

09:52 AM

Pemilihan E-Wallet

Mohon Berikan Penilaian Anda Mengenai ShopeePay

Keterangan : 1 - Tidak Memenuhi Kriteria
2 - Kurang Baik
3 - Cukup/Netral
4 - Baik
5 - Sangat Baik

Biaya Admin : Rp 2.000

Bagaimana Penilaian Anda tentang Keandalan Fitur-Fitur pada ShopeePay

SPK E-Wallet

Apakah anda yakin ingin keluar dari kuisioner?

[No](#) [Yes](#)

1

Bagaimana Penilaian Anda tentang Fitur-Fitur pada ShopeePay

1

Bagaimana Penilaian Anda tentang Layanan pada ShopeePay

1

Selanjutnya

Gambar 3. 9 Rancangan Halaman AlertDialog

Jika pengguna keluar dari kuisioner, terdapat peringatan tampilan komponen yaitu tampilan *alertdialog* seperti pada gambar di atas.

2) Rancangan Halaman Penilaian OVO

The image shows a mobile phone screen with a wireframe for an evaluation page titled "Pemilihan E-Wallet". The status bar at the top shows "09:52 AM" and signal icons. The main heading is "Pemilihan E-Wallet". Below it, the text reads "Mohon Berikan Penilaian Anda Mengenai OVO". A legend for the rating scale is provided: "Keterangan : 1 - Tidak Memenuhi Kriteria, 2 - Kurang Baik, 3 - Cukup/Netral, 4 - Baik, 5 - Sangat Baik". The evaluation questions and their corresponding dropdown menus are: "Biaya Admin : Rp 1.200 Bagaimana Penilaian Anda tentang Kemudahan Top Up pada OVO" (dropdown shows 1), "Bagaimana Penilaian Anda tentang Promo pada OVO" (dropdown shows 1), "Bagaimana Penilaian Anda tentang Fitur-Fitur pada OVO" (dropdown shows 1), and "Bagaimana Penilaian Anda tentang Layanan pada OVO" (dropdown shows 1). At the bottom, there is a button labeled "Selanjutnya".

Gambar 3. 10 Rancangan Halaman Penilaian OVO

Gambar diatas merupakan rancangan antarmuka halaman penilaian untuk *E-Wallet* OVO. Pada desain antarmuka halaman penilaian ini terdapat komponen view seperti tampilan teks, *combo box* dan tampilan *button*.

3) Rancangan Halaman Penilaian GoPay

The image shows a wireframe of a smartphone screen displaying an evaluation form for GoPay's E-Wallet. The screen has a status bar at the top showing '09:52 AM' and signal icons. The main title is 'Pemilihan E-Wallet'. Below it, a subtitle asks for an evaluation of GoPay. A legend explains the rating scale from 1 to 5. The form includes a fixed admin fee, followed by four rating questions: ease of top-up, promo availability, features, and service quality. Each question has a dropdown menu currently showing '1'. A 'Selanjutnya' button is at the bottom.

09:52 AM

Pemilihan E-Wallet

Mohon Berikan Penilaian Anda Mengenai GoPay

Keterangan : 1 - Tidak Memenuhi Kriteria
2 - Kurang Baik
3 - Cukup/Netral
4 - Baik
5 - Sangat Baik

Biaya Admin : Rp 2.000

Bagaimana Penilaian Anda tentang Kemudahan Top Up pada GoPay

1

Bagaimana Penilaian Anda tentang Promo pada GoPay

1

Bagaimana Penilaian Anda tentang Fitur-Fitur pada GoPay

1

Bagaimana Penilaian Anda tentang Layanan pada GoPay

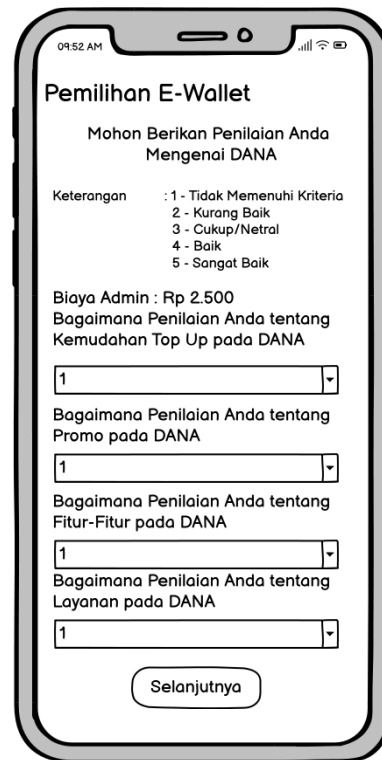
1

Selanjutnya

Gambar 3. 11 Rancangan Halaman Penilaian GoPay

Gambar diatas merupakan rancangan antarmuka halaman penilaian untuk *E-Wallet* GoPay. Pada desain antarmuka halaman penilaian ini terdapat komponen view seperti tampilan teks, *combo box* dan tampilan *button*.

4) Rancangan Halaman Penilaian DANA



The image shows a mobile application interface for evaluating DANA E-Wallet. The screen is titled "Pemilihan E-Wallet" and asks the user to provide a rating for DANA. It includes a rating scale from 1 to 5, a list of evaluation criteria, and a "Selanjutnya" (Next) button.

09:52 AM

Pemilihan E-Wallet

Mohon Berikan Penilaian Anda Mengenai DANA

Keterangan : 1 - Tidak Memenuhi Kriteria
2 - Kurang Baik
3 - Cukup/Netral
4 - Baik
5 - Sangat Baik

Biaya Admin : Rp 2.500
Bagaimana Penilaian Anda tentang Kemudahan Top Up pada DANA

1

Bagaimana Penilaian Anda tentang Promo pada DANA

1

Bagaimana Penilaian Anda tentang Fitur-Fitur pada DANA

1

Bagaimana Penilaian Anda tentang Layanan pada DANA

1

Selanjutnya

Gambar 3. 12 Rancangan Halaman Penilaian DANA

Gambar diatas merupakan rancangan antarmuka halaman penilaian untuk *E-Wallet* DANA. Pada desain antarmuka halaman penilaian ini terdapat komponen view seperti tampilan teks, *combo box* dan tampilan *button*.

5) Rancangan Halaman Penilaian LinkAja

The image shows a mobile application interface for evaluating LinkAja's E-Wallet. The screen is titled "Pemilihan E-Wallet" and asks the user to provide a rating. It includes a legend for ratings from 1 to 5, a fixed admin fee of Rp 1.500, and four dropdown menus for rating different aspects: ease of top-up, promo, features, and service. Each dropdown menu currently shows the value "1". A "Selanjutnya" (Next) button is at the bottom.

09:52 AM

Pemilihan E-Wallet

Mohon Berikan Penilaian Anda Mengenai LinkAja

Keterangan : 1 - Tidak Memenuhi Kriteria
2 - Kurang Baik
3 - Cukup/Netral
4 - Baik
5 - Sangat Baik

Biaya Admin : Rp 1.500

Bagaimana Penilaian Anda tentang Kemudahan Top Up pada LinkAja

1

Bagaimana Penilaian Anda tentang Promo pada LinkAja

1

Bagaimana Penilaian Anda tentang Fitur-Fitur pada LinkAja

1

Bagaimana Penilaian Anda tentang Layanan pada LinkAja

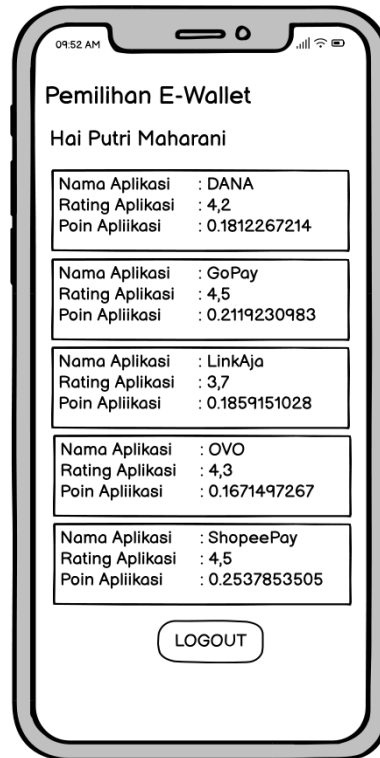
1

Selanjutnya

Gambar 3. 13 Rancangan Halaman Penilaian LinkAja

Gambar diatas merupakan rancangan antarmuka halaman penilaian untuk *E-Wallet* LinkAja. Pada desain antarmuka halaman penilaian ini terdapat komponen view seperti tampilan teks, *combo box* dan tampilan *button*.

3.5.4.5 Rancangan Halaman Perangkingan



Gambar 3. 14 Rancangan Halaman Hasil Perangkingan

Pada gambar di atas merupakan gambar desain antarmuka dalam Aplikasi *Pemilihan E-Wallet*, dimana pengguna dapat melihat hasil penilaian yang telah diisi sebelumnya. Adapun komponen yang terdapat pada aplikasi berupa tampilan *button* dan *textview*.

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Bab ini akan menjelaskan alur uji coba dan evaluasi dari penelitian yang telah dilakukan. Uji coba dilakukan untuk mengetahui keberhasilan penerapan aplikasi yang dibuat dan evaluasi dilakukan untuk menganalisis hasil uji coba serta untuk mendapatkan kesimpulan dan saran untuk pengembangan penelitian selanjutnya.

Seluruh proses perancangan dan penghitungan nilai di terapkan ke dalam sistem yang telah dibuat melalui Android Studio.

Tujuan diterapkannya sistem ini adalah:

1. Memeriksa apakah setiap proses yang telah dibuat berjalan dengan benar dan tidak ada kesalahan.
2. Menguji sistem perhitungan weighted product berdasarkan nilai yang telah didapatkan.
3. Memelihara sistem yang akan dikembangkan di masa mendatang.

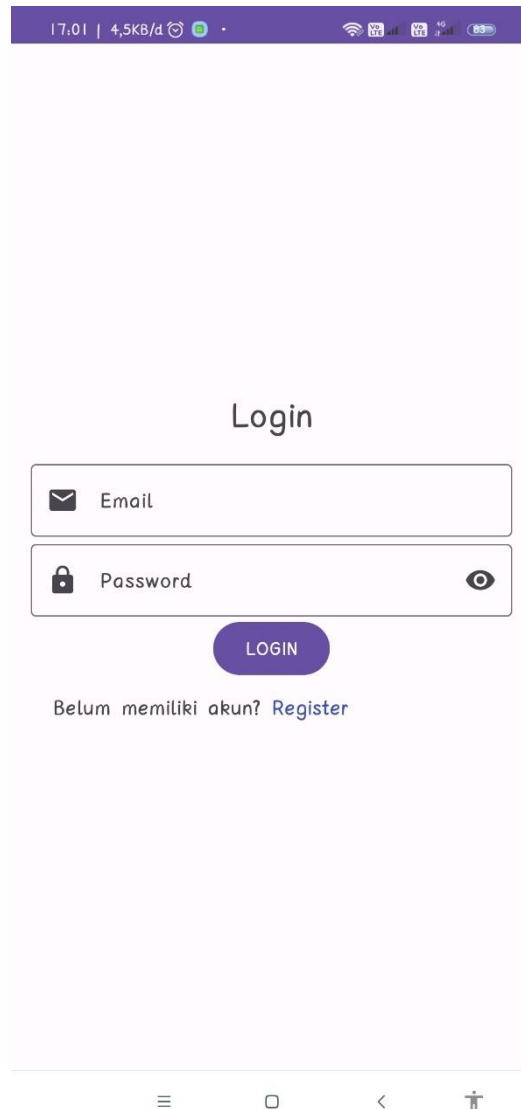
Alat yang digunakan untuk menerapkan sistem ini adalah:

- Perangkat Lunak Android Studio.
- Sistem Operasi Windows 10.

4.1.1 Implementasi Tampilan *Interface*

Implementasi tampilan sistem merupakan tampilan dari sistem yang telah dibangun. Tampilan ini berinteraksi langsung dengan *user*.

4.1.1.1 Tampilan Halaman Login



Gambar 4. 1 Tampilan Halaman Login

Pada gambar di atas merupakan gambar desain antarmuka Aplikasi Pemilihan *E-Wallet* yang menjelaskan suatu kegiatan untuk melakukan login pada desain antarmuka tersebut dapat dilihat dari komponen tampilan Android seperti: *textView*, *edittext* dan *button*. Masing-masing komponen tampilan tersebut mempunyai peranan masing-masing untuk dapat menjalankan dan membantu pengguna dalam menjalankan proses login ke dalam aplikasi.

Pengguna yang belum mempunyai akun dapat melakukan pendaftaran terlebih dahulu dengan klik register, kemudian akan muncul textview yang harus diisi yaitu username, email dan password untuk melakukan proses autentikasi.

4.1.1.2 Tampilan Halaman Register

17:42 | 6,3KB/d

Register

Username

Email

Password 0/20

REGISTER

Sudah memiliki akun? [Login](#)

Gambar 4. 2 Tampilan Halaman Register

Pada gambar di atas merupakan gambar implementasi antarmuka pada Aplikasi Pemilihan E-Wallet, dimana antarmukanya menjelaskan tentang

login ke dalam sistem menggunakan akun Google, desain di atas terdapat komponen-komponen seperti *textview* dan *button*.

4.1.1.3 Tampilan Rancangan Halaman Utama



Gambar 4. 3 Tampilan Halaman Utama

Jika autentikasi berhasil maka pengguna akan diarahkan ke halaman berikutnya yaitu halaman utama. Pada halaman ini terdapat komponen seperti *textview* dan *button*, pengguna dapat melihat nama aplikasi, rating aplikasi dan poin aplikasi untuk 5 e-wallet yang terdiri dari DANA,

GoPay, LinkAja, OVO dan ShopeePay. Pengguna dapat mengklik *button* “Beri Penilaian” untuk menilai setiap aplikasi. Dan klik *button* “logout” jika ingin keluar aplikasi.

4.1.1.4 Tampilan Halaman Penilaian

1) Tampilan Halaman Penilaian ShopeePay

19:40 | 0,1KB/d

Mohon Berikan Penilaian Anda Mengenai ShopeePay

Keterangan :

- 1 - Tidak Memenuhi Kriteria
- 2 - Kurang Baik
- 3 - Cukup / Netral
- 4 - Baik
- 5 - Sangat Baik

Biaya Admin : Rp.2,000

Bagaimana Penilaian Anda tentang Kemudahan TopUp pada ShopeePay

| ▼

Bagaimana Penilaian Anda tentang Promo pada ShopeePay

| ▼

Bagaimana Penilaian Anda tentang Fitur-fitur dari ShopeePay

| ▼

Bagaimana Penilaian Anda tentang Layanan yang diberikan oleh ShopeePay

| ▼

SELANJUTNYA

Gambar 4. 4 Tampilan Halaman Penilaian ShopeePay

Setelah mengklik "Beri Penilaian" pengguna akan diarahkan ke halaman penilaian. Halaman penilaian pertama adalah halaman penilaian ShopeePay, pengguna dapat menjawab 4 pertanyaan dan setiap pertanyaan pengguna bisa memilih nilai 1 sampai 5. Setelah selesai, pengguna dapat

mengklik *button* selanjutnya. Jika pengguna keluar sebelum menjawab kuesioner, maka akan muncul peringatan berupa *AlertDialog* seperti pada gambar.

The image shows a survey form titled "Mohon Berikan Penilaian Anda Mengenai ShopeePay". It includes a rating scale from 1 to 5, with 1 being "Tidak Memenuhi Kriteria" and 5 being "Sangat Baik". Below the scale, it states "Biaya Admin : Rp.2,000" and asks for a rating on the ease of top-up. An "AlertDialog" is overlaid on the form, titled "SPK E-Wallet", asking "Apakah anda yakin ingin keluar dari kuisisioner?" (Are you sure you want to exit the questionnaire?). The dialog has "No" and "Yes" buttons. At the bottom of the form, there is a "SELANJUTNYA" button.

Gambar 4. 5 Tampilan Halaman *AlertDialog*

2) Tampilan Halaman Penilaian OVO

17:45 | 5,0KB/d

Mohon Berikan Penilaian Anda Mengenai OVO

Keterangan :

- 1 ~ Tidak Memenuhi Kriteria
- 2 ~ Kurang Baik
- 3 ~ Cukup / Netral
- 4 ~ Baik
- 5 ~ Sangat Baik

Biaya Admin : Rp.1,200

Bagaimana Penilaian Anda tentang Kemudahan TopUp pada OVO

1

Bagaimana Penilaian Anda tentang Promo pada OVO

1

Bagaimana Penilaian Anda tentang Fitur-fitur dari OVO

1

Bagaimana Penilaian Anda tentang Layanan yang diberikan oleh OVO

1

SELANJUTNYA

Gambar 4. 6Tampilan Halaman Penilaian OVO

Halaman berikutnya adalah halaman penilaian OVO, pengguna dapat menjawab 4 pertanyaan dan setiap pertanyaan pengguna bisa memilih nilai 1 sampai 5. Setelah selesai, pengguna dapat mengklik *button* selanjutnya.

3) Tampilan Halaman Penilaian GoPay

19.41 | 0,0KB/d

Mohon Berikan Penilaian Anda Mengenai GoPay

Keterangan :

- 1 ~ Tidak Memenuhi Kriteria
- 2 ~ Kurang Baik
- 3 ~ Cukup / Netral
- 4 ~ Baik
- 5 ~ Sangat Baik

Biaya Admin : Rp.2,000

Bagaimana Penilaian Anda tentang Kemudahan TopUp pada GoPay

| ▼

Bagaimana Penilaian Anda tentang Promo pada GoPay

| ▼

Bagaimana Penilaian Anda tentang Fitur-fitur dari GoPay

| ▼

Bagaimana Penilaian Anda tentang Layanan yang diberikan oleh GoPay

| ▼

SELANJUTNYA

Gambar 4. 7 Tampilan Halaman Penilaian GoPay

Halaman berikutnya adalah halaman penilaian GoPay, pengguna dapat menjawab 4 pertanyaan dan setiap pertanyaan pengguna bisa memilih nilai 1 sampai 5. Setelah selesai, pengguna dapat mengklik *button* selanjutnya.

4) Tampilan Halaman Penilaian DANA

19.41 | 1,7KB/d

Mohon Berikan Penilaian Anda Mengenai Dana

Keterangan :

- 1 ~ Tidak Memenuhi Kriteria
- 2 ~ Kurang Baik
- 3 ~ Cukup / Netral
- 4 ~ Baik
- 5 ~ Sangat Baik

Biaya Admin : Rp.2,500

Bagaimana Penilaian Anda tentang Kemudahan TopUp pada Dana

| ▼

Bagaimana Penilaian Anda tentang Promo pada Dana

| ▼

Bagaimana Penilaian Anda tentang Fitur-fitur dari Dana

| ▼

Bagaimana Penilaian Anda tentang Layanan yang diberikan oleh Dana

| ▼

SELANJUTNYA

Gambar 4. 8 Tampilan Halaman Penilaian DANA

Halaman berikutnya adalah halaman penilaian DANA, pengguna dapat menjawab 4 pertanyaan dan setiap pertanyaan pengguna bisa memilih nilai 1 sampai 5. Setelah selesai, pengguna dapat mengklik *button* selanjutnya.

5) Tampilan Halaman Penilaian LinkAja

19.41 | 1,4KB/d

Mohon Berikan Penilaian Anda Mengenai LinkAja

Keterangan :
 1 ~ Tidak Memenuhi Kriteria
 2 ~ Kurang Baik
 3 ~ Cukup / Netral
 4 ~ Baik
 5 ~ Sangat Baik

Biaya Admin : Rp.1,500
 Bagaimana Penilaian Anda tentang Kemudahan TopUp pada LinkAja

1

Bagaimana Penilaian Anda tentang Promo pada LinkAja

1

Bagaimana Penilaian Anda tentang Fitur-fitur dari LinkAja

1

Bagaimana Penilaian Anda tentang Layanan yang diberikan oleh LinkAja

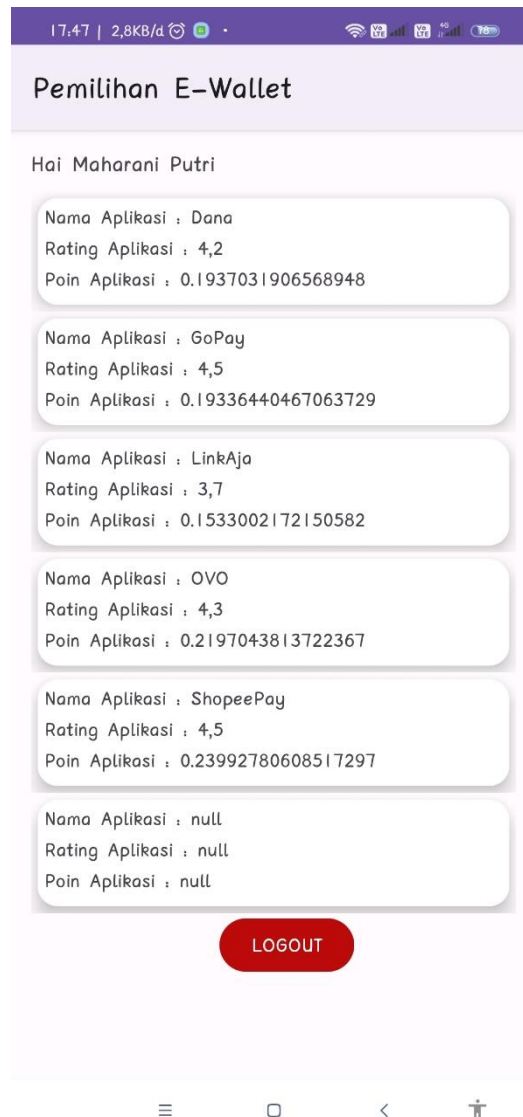
1

SELESAI

Gambar 4. 9Tampilan Halaman Penilaian LinkAja

Halaman berikutnya adalah halaman penilaian LinkAja, pengguna dapat menjawab 4 pertanyaan dan setiap pertanyaan pengguna bisa memilih nilai 1 sampai 5. Setelah selesai, pengguna dapat mengklik *button* selanjutnya.

4.1.1.5 Tampilan Halaman Hasil Perangkingan



Gambar 4. 10 Tampilan Halaman Hasil Perangkingan

Setelah penilaian dilakukan, poin setiap aplikasi akan terisi secara otomatis dan pengguna dapat melihat *e-wallet* mana yang memiliki poin tertinggi dan terendah. Setelah melihat hasilnya, pengguna dapat mengklik *button* "logout" untuk keluar.

4.1.2 Implementasi Metode *Weighted Product*

Penerapan metode *weighted product* pada penelitian ini diterapkan pada sistem pendukung pemilihan *E-Wallet*. Metode *weighted product* dibuat menggunakan bahasa pemrograman java yang kemudian ditautkan menggunakan Android Studio. Berikut adalah source code yang digunakan pada sistem ini:

```

Integer totjumshotopup = jumshotopup + 1;
Integer totjumshopromo = jumshopromo + 1;
Integer totjumshofitur = jumshofitur + 1;
Integer totjumsholayanan = jumsholayanan + 1;
Integer totshotopup = shotopup + shoTopup;
Integer totshopromo = shopromo + shoPromo;
Integer totshofitur = shofitur + shoFitur;
Integer totsholayanan = sholayanan +
shoLayanan;

Integer totalShoTopup = totshotopup /
totjumshotopup;

Integer totalShoPromo = totshopromo /
totjumshopromo;

Integer totalShoFitur = totshofitur /
totjumshofitur;

Integer totalShoLayanan = totsholayanan /
totjumsholayanan;

Double tsadmin =
Double.parseDouble(String.valueOf(shoadmin));
Double tstopup =
Double.parseDouble(String.valueOf(totalShoTopup));
Double tspromo =
Double.parseDouble(String.valueOf(totalShoPromo));
Double tsfitur =
Double.parseDouble(String.valueOf(totalShoFitur));
Double tslayanan =
Double.parseDouble(String.valueOf(totalShoLayanan));
Double tsAdmin = Math.pow(tsadmin,
pangkatc1);

Double tsTopup = Math.pow(tstopup,
pangkatc2);

```

```

        Double tsPromo = Math.pow(tspromo,
pangkatc3);

        Double tsFitur = Math.pow(tsfitur,
pangkatc4);

        Double tsLayanan = Math.pow(tslayanan,
pangkatc5);

        Double shoS = tsAdmin * tsTopup * tsPromo *
tsFitur * tsLayanan;

        Double totals = linkS + shoS + ovoS + goS +
danS;

        Double linkPoin = linkS / totals;
        Double shoPoin = shoS / totals;
        Double ovoPoin = ovoS / totals;
        Double goPoin = goS / totals;
        Double danPoin = danS / totals;

wdatabase.child(shopee).child("poin").setValue(shoPoin);

wdatabase.child(ovo).child("poin").setValue(ovoPoin);

wdatabase.child(gopay).child("poin").setValue(goPoin);

wdatabase.child(dana).child("poin").setValue(danPoin);

wdatabase.child(appName).child("poin").setValue(linkPoin);

        Intent linkmain = new
Intent(getApplicationContext(), MainActivity.class);
        linkmain.putExtra("username", username);
        linkmain.putExtra("finish", finish);
        startActivity(linkmain);
        finish();
    }

    @Override

```

```

        public void onCancelled(@NonNull DatabaseError
error) {}

        });

    }

    });

}

@Override
public void onBackPressed() {
    Intent linda = new Intent(getApplicationContext(),
DanaVote.class);
    linda.putExtra("username", username);
    startActivity(linda);
    finish();
}
}

```

Tabel 14 Implementasi Merode *Weighted Product*

4.1.3 Implementasi *Firebase Realtime database*

Selanjutnya pada sub-bab ini kita akan membahas bagaimana melakukan implementasi *Firebase realtime database* yang berguna untuk *handling* penyimpanan dan manipulasi data dalam bentuk database. Hal pertama yang harus dilakukan untuk mengimplementasikan *realtime database* ini adalah mengkonfigurasi *rules* pada *realtime database*.

Aturannya sendiri merupakan aturan keamanan dalam penggunaan *realtime database* mengenai hak akses siapa saja yang dapat melakukan proses baca dan tulis data. Dalam penelitian, aturan yang digunakan adalah memperbolehkan semua akses akan menulis dan membaca pada *realtime database* Firebase, seperti terlihat pada gambar berikut:



Gambar 4. 11 Rules firebase realtime database Aplikasi Pemilihan E-Wallet

Penjelasan aturan diatas adalah dalam proses manipulasi database, aplikasi yang terintegrasi dengan *realtime database* selalu diperbolehkan untuk diproses menulis dan membaca. Selanjutnya untuk mempermudah pelaksanaannya Dashboard aplikasi Firebase dapat mengisi struktur dokumen setelah diskusi pada sub-bab desain database.

Sama seperti menerapkan *firebase authentication* dalam implementasi *firebase realtime database* juga menggunakan *built in SDK* yang di miliki oleh google melalui perpustakaan yang sudah di integrasikan melalui *gradle*.

4.1.3.1 Add Single Listener Value Event

Pada dasarnya konsep pemanggilan data di Android menggunakan *listener* dan *callback*, secara sederhana konsep *listener* bisa dikatakan sebagai event dalam proses logika yang kita tidak tahu kapan pengguna akan memproses event tersebut, dan *callback* adalah tempat wadah atau tanda kembali dari event yang sudah diminta. Penggunaan *add Single Value Event Listener* hanya untuk mendapatkan atau melakukan permintaan data ke server hanya sekali dan tidak memuat lagi jika ada pembaruan di dalam aplikasi, jika data server berubah dan kemudian jenis tanggapan yang diproduksi ditelepon di dalam *callback* jika pengguna membutuhkan. Implementasi adalah sebagai mengikuti:

```

@Override

    public void onClick(View view) {
        FirebaseAuth.getInstance().signOut();
        Intent mainlog = new
Intent(getApplicationContext(), Login.class);
        startActivity(mainlog);
        finish();
    }
});
udatabase.addListenerForSingleValueEvent(new
 ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot
snapshot) {
        if
(snapshot.child(username).child("finish").exists()){
            votebtn.setVisibility(View.GONE);
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error)
{}
});
votebtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent mainvot = new
Intent(getApplicationContext(), ShopeepayVote.class);
        mainvot.putExtra("username", username);
        startActivity(mainvot);
        finish();
    }
});
}

```

Tabel 15 Add Single Value Event Listener

4.1.3.2 Add Value Event Listener

Berbeda dengan *add Single Value Event Listener*, fungsi ini akan selalu memicu proses perubahan yang terjadi pada database. Akan tetapi memiliki konsep implementasi yang sama dengan *addSingleListenerValueEvent* yaitu menghasilkan dua *callback on Canceled* dan *on Data Change*. Implementasinya adalah sebagai berikut:

```
private void tampilData() {
    database.addValueEventListener(new ValueEventListener()
    {
        @Override
        public void onDataChange(@NonNull DataSnapshot
snapshot) {
            arrayList = new ArrayList<>();
            for (DataSnapshot item :
snapshot.getChildren()) {
                Wallet wallet = new Wallet();
                wallet.setAppname(item.child("appname").getValue(String.class));
                wallet.setRating(item.child("rating").getValue(String.class));
                wallet.setPoin(item.child("poin").getValue(Double.class));
                arrayList.add(wallet);
            }
            adapter = new WalletAdapter(arrayList,
MainActivity.this);
            rvEwallet.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }
        @Override
        public void onCancelled(@NonNull DatabaseError
error) {
        }
    });
}
```

Tabel 16 Add Event Event Listener

4.2 Pengujian Sistem

4.2.1 Pengujian *Black Box*

Pengujian dilakukan untuk mengetahui apakah sistem apa yang dibuat bisa berfungsi dengan baik atau tidak. Tes ini dilakukan dengan menggunakan satu metode pengujian perangkat lunak yaitu pengujian *Black Box*.

Di bawah ini adalah: rincian bagian-bagian yang diuji, soal tes dan juga hasil pengujian menggunakan *Black Box* terhadap Aplikasi Pemilihan *E-Wallet*.

No.	<i>Test Case</i>	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
1.	Pengujian pendaftaran menggunakan email	Pengguna dapat mendaftar menggunakan username, email dan password lalu bisa melakukan <i>login</i> .	Sistem memasukkan data ke dalam <i>authentication</i> dan antarmuka pergi ke halaman utama.	Berhasil
2.	Pengujian <i>Login</i>	Pengguna akan pergi ke halaman atau <i>activity home</i> .	Sistem mengeluarkan <i>toast to login</i> dan pergi ke <i>activity home</i> .	Berhasil
3.	Pengujian <i>button</i> “beri penilaian”	Pengguna menuju ke menu penilaian dari tiap-tiap aplikasi <i>E-Wallet</i> .	Sistem dapat menampilkan pertanyaan-pertanyaan dari aplikasi <i>E-Wallet</i> dan emebri nilai dari 1-5	Berhasil
4.	Pengujian <i>button</i> “Selanjutnya”	Ketika di klik akan pergi ke halaman selanjutnya dan menampilkan	Siste menampilkan pertanyaan-pertanyaan dari tiap <i>E-Wallet</i> .	Berhasil

		pertanyaan dari tiap <i>E-Wallet</i> .		
5.	Apabila pengguna keluar sebelum mengisi kuisioner.	Akan muncul sebuah peringatan berupa <i>alert dialog</i> .	Muncul sebuah <i>alert dialog</i> .	Berhasil
6.	Pengujian Halaman Perangkingan	Sistem akan menampilkan daftar <i>e-wallet</i> berisi poin aplikasi yang sudah terisi.	Sistem akan menampilkan daftar <i>e-wallet</i> berisi poin aplikasi yang sudah terisi.	Berhasil
7.	<i>Button logout</i>	Sistem akan kembali ke halaman login.	Berhasil menampilkan informasi <i>logout</i>	Berhasil

Tabel 17 Pengujian *Black Box*

4.2.2 Pengujian Akurasi

Hasil pengujian pada tabel 18 menunjukkan rekomendasi *E-Wallet* oleh sistem dibandingkan dengan perhitungan manual menggunakan Microsoft Excel.

Vektor	Nilai Vektor		<i>E-Wallet</i>
	Manual (Excel)	Sistem	
V ₁	0,21111	0,21111	OVO
V ₂	0,19302	0,19302	GoPay
V ₃	0,19638	0,19638	DANA
V ₄	0,18180	0,18180	LinkAja
V ₅	0,21769	0,21769	ShopeePay

Tabel 18 Pengujian Akurasi

Nilai terbesar dari hasil perhitungan secara manual menggunakan Microsoft Excel adalah 0,21769 yaitu V_5 dan nilai terkecilnya sebesar 0,18180 yaitu V_4 , sedangkan perhitungan sistem nilai terbesar adalah 0,21769 dan nilai terkecilnya adalah 0,18180, sehingga didapat kesimpulan dari sebagian besar alternatif *E-Wallet* yang direkomendasikan adalah “SHOPEEPAY”. Validasi berdasarkan hasil perhitungan manual menggunakan Microsoft Excel dan perhitungan secara sistematis dapat dikatakan valid karena hasil perhitungan yang didapat pun sama.

BAB V

PENUTUP

5.1 Kesimpulan

Bab ini menjelaskan uji coba dan studi evaluasi yang telah dilakukan. Uji coba dilakukan untuk melihat sejauh mana keberhasilan implementasi perangkat lunak yang telah dibuat dan dievaluasi dengan menganalisis hasil uji coba dan evaluasi untuk mendapatkan kesimpulan dan saran untuk pengembangan selanjutnya. Dari hasil uji coba dan diskusi yang dilakukan, dapat diperoleh kesimpulan sebagai berikut:

1. Penelitian ini menghasilkan aplikasi sistem pendukung keputusan pemilihan *E-Wallet* berbasis android terbaik yang memudahkan calon pengguna mendapatkan rekomendasi *E-Wallet* yang diinginkan.
2. Sistem yang dibuat dapat menerapkan perhitungan dengan menggunakan metode *Weighted product*. Penerapan metode *Weighted Product* pada aplikasi berbasis android menghasilkan perhitungan yang akurat setelah diuji dengan perhitungan sistem dan perhitungan data real memiliki hasil yang seimbang.
3. Pada sistem aplikasi sudah terdapat rekomendasi dari kuisisioner yang telah dilakukan sebelumnya, dan mendapatkan hasil akhir berupa urutan dari beberapa *E-Wallet* yang dibahas. Dari kuisisioner tersebut dapat disimpulkan bahwa GoPay merupakan *E-Wallet* yang memiliki nilai tertinggi dari *E-Wallet* lainnya, hasil tersebut didapat dari penilaian pada setiap kriteria yang ditanyakan pada *E-Wallet* tersebut memiliki rata-rata nilai yang lebih tinggi dari *E-Wallet* lainnya.
4. Dengan didapatkannya rekomendasi *E-Wallet* dari beberapa peserta kuisisioner bukan berarti *E-Wallet* lainnya tidak memiliki kualitas yang baik, semua *E-Wallet* yang dibahas dalam penelitian ini memiliki penilaian tinggi pada beberapa kriteria. Seperti ShopeePay yang memiliki nilai tertinggi

yaitu 0,217 serta OVO yang mendapat nilai sederhana membuat aplikasi ini menjadi *E-Wallet* paling stabil dari segala kriteria.

5.2 Saran

Berdasarkan pengujian yang telah dilakukan oleh peneliti terhadap sistem Pemilihan E-wallet dengan metode *Weighted product* diharapkan pada penelitian selanjutnya kriteria dan subkriteria dapat diperbanyak sehingga menjadi sistem pemilihan yang lebih baik daripada penelitian ini, atau bisa juga dibandingkan dengan metode lain dalam hal penilaian untuk mendapatkan hasil yang lebih memuaskan.

Dalam penelitian ini peneliti hanya memasukkan beberapa *E-Wallet* ternama di Indonesia, diharapkan penelitian selanjutnya dapat menambah jumlah *E-Wallet* yang akan diteliti.

DAFTAR PUSTAKA

- Artanti Prahastuti, R., Hutabarat, J., & Indra Laksmmana, D. (n.d.). Penentuan Bobot Kriteria Pemilihan Alternatif Online-Shop dalam Memasarkan Produk Kosmetik dengan Menggunakan Analytical Hierarchy Process (AHP). In *Jurnal Teknologi dan Manajemen Industri* (Vol. 6, Issue 2). <https://iprice.co.id/insight.co>
- Aulia, S. (2020). Pola Perilaku Konsumen Digital Dalam Memanfaatkan Aplikasi Dompet Digital. *Jurnal Komunikasi*, 12(2), 311. <https://doi.org/10.24912/jk.v12i2.9829>
- Homepage, J., Amelia Putri, W., Rachmawati, D., Sarah Silalahi, W., Studi Sistem Informasi, P., Sains dan Teknologi, F., Sultan Syarif Kasim Riau, U., Studi Matematika, P., & Matematika dan Ilmu Pengetahuan Alam, F. (2022). *MALCOM: Indonesian Journal of Machine Learning and Computer Science E-Wallet Selection Decision Support System Using Analytic Hierarchy Process-TOPSIS Method Sistem Pendukung Keputusan Pemilihan E-Wallet Menggunakan Metode Analytic Hierarchy Process-TOPSIS*. 2, 18–27.
- Karsen, M., Chandra, Y. U., & Juwitasary, H. (2019). Technological factors of mobile payment: A systematic literature review. *Procedia Computer Science*, 157, 489–498. <https://doi.org/10.1016/j.procs.2019.09.004>
- Muqorobin, M., & Ma'ruf, M. H. (2022). SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN OBYEK WISATA TERBAIK DI KABUPATEN SRAGEN DENGAN METODE WEIGHTED PRODUCT. *Jurnal Teknik Informasi Dan Komputer (Tekinkom)*, 5(2), 364. <https://doi.org/10.37600/tekinkom.v5i2.536>
- Nadila, T. A., Andryana, S., & Sholihati, I. D. (2021). Analisa Perbandingan Metode MOORA, Promethee, dan Weighted product dalam Penentuan Lokasi Usaha. *Jurnal Teknologi Informasi dan Komunikasi*, 5(3), 2021. <https://doi.org/10.35870/jti>
- Noviansyah, M. R., Suharso, W., Azmi, M. S., Hermawan, M., Mustikaningtyas, D. R., Ulya, F. S., & Chandranegara, R. (t.t.). *Seminar Nasional Teknologi dan Rekayasa (SENTRA) 2019 ISSN (Cetak) 2527-6042 eISSN (Online). www.cybermallmalang.com*.

- Nurdiawan, O. (2018). *PENERAPAN SISTEM PENDUKUNG KEPUTUSAN DENGAN METODE PROFILE MATCHING DALAM PENILAIAN KINERJA KARYAWAN* (Vol. 8, Issue 2).
- Nurjannah, M. (2021). *IMPLEMENTASI METODE WEIGHT PRODUCT DAN ANALYTICAL HIERARCHY PROCESS DALAM MENENTUKAN BENIH KOPI BERKUALITAS PADA IP2TP GAYO BERBASIS MOBILE SKRIPSI*.
- Sitinjak, R., & Nurlela, S. (2022). Pemilihan *E-Wallet* Pada Kerry Parcel Outlet Menggunakan Metode Analytical Hierarchy Process (AHP). Dalam *Sains Teknik Elektro* (Vol. 3, Nomor 2). <http://jurnal.bsi.ac.id/index.php/insantek49>
- Syach Nuralam, H., Raya Jl Yos Sudarso, P., Raya, J., Raya, P., & Tengah, K. (t.t.). *PEMBUATAN APLIKASI KIDS APPLICATION DENGAN MENGGUNAKAN PROGRAM ANDROID STUDIO*.
- Sya'id Maftuh. (2022). *SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN TEMPAT WISATA DI KOTA BATU MENGGUNAKAN METODE WEIGHTED PRODUCT*. Skripsi, 2.
- Siregar, V. M. M., Sonang, S., & Damanik, E. (2021). *SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PELANGGAN TERBAIK MENGGUNAKAN METODE WEIGHTED PRODUCT*. *Jurnal Teknik Informasi Dan Komputer (Tekinkom)*, 4(2), 239. <https://doi.org/10.37600/tekinkom.v4i2.392>
- Teguh Kurniawan. (2021). *IMPLEMENTASI FIREBASE DALAM PENGEMBANGAN PLATFORM SEWA SARANA OLAHRAGA BERBASIS ANDROID*. <http://repository.uinsu.ac.id/13103/1/%5BFINAL%20PDF%5D%20-%20SKRIPSI%20TEGUH%20SOFTCOPY.pdf>
- Yanti, F., & Sutresna, J. (2020). *Sistem Pendukung Keputusan Pemilihan Customer Terbaik Menggunakan Metode WP*. 1(2). <http://openjournal.unpam.ac.id/index.php/JOAIIA/index90>
- Wahyudi, M. A. (n.d.). *RANCANG BANGUN SISTEM REKOMENDASI PEMILIHAN LOKASI KULINER MENGGUNAKAN METODE WEIGHTED PRODUCT*.

Lampiran 1

Source Code Program

Login.java

```
package com.putri.spk;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputEditText;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

public class Login extends AppCompatActivity {
    TextInputEditText etEmail,
    etPassword;
    Button btnLog;
    FirebaseAuth mAuth;
    ProgressBar progressBar;
    TextView tvRegister;
    DatabaseReference
    database;

    @Override
    protected void
    onCreate(Bundle
    savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_login);

        etEmail =
        findViewById(R.id.email);
        etPassword =
        findViewById(R.id.password);
        btnLog =
        findViewById(R.id.btnLogin);
        progressBar =
        findViewById(R.id.progressBar);
        ;
        tvRegister =
        findViewById(R.id.registerNow);
        ;
        mAuth =
        FirebaseAuth.getInstance();
        database =
        FirebaseDatabase.getInstance().getReference("user");

        tvRegister.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void
            onClick(View view) {
```

```

        Intent
logtoreg = new
Intent(getApplicationContext()
, Register.class);

startActivity(logtoreg);
        finish();
    }
});

btnLog.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void
onClick(View view) {

progressBar.setVisibility(View
.VISIBLE);

        String email,
password;
        email =
String.valueOf(etEmail.getText
());
        password =
String.valueOf(etPassword.getT
ext());

        if
(TextUtils.isEmpty(email)){

Toast.makeText(getApplicationContext
ontext(), "Mohon Mengisi Email
Terlebih Dahulu",
Toast.LENGTH_SHORT).show();
        return;
    }

    if
(TextUtils.isEmpty(password)){

Toast.makeText(getApplicationContext
ontext(), "Mohon Mengisi
Password Terlebih Dahulu",
Toast.LENGTH_SHORT).show();
        return;
    }

 mAuth.signInWithEmailAndPasswo
rd(email,
password).addOnCompleteListener
r(new
OnCompleteListener<AuthResult>
() {

        @Override

```

```

        public
void onComplete(@NonNull
Task<AuthResult> task) {

progressBar.setVisibility(View
.GONE);

        if
(task.isSuccessful()){

database.addListenerForSingleV
alueEvent(new
ValueEventListener() {

    @Override

    public void
onDataChange(@NonNull
DataSnapshot snapshot) {

        if
(snapshot.child(password).exis
ts()){

String username =
snapshot.child(password).child
("username").getValue(String.C
lass);

Toast.makeText(getApplicationContext
ontext(), "Login Berhasil",
Toast.LENGTH_SHORT).show();

Intent main = new
Intent(getApplicationContext()
, MainActivity.class);

main.putExtra("username",
username);

startActivity(main);

finish();

    }

    }

    @Override

    public void
onCancelled(@NonNull
DatabaseError error) {

    }

}

```

```
});
                                } else
                                });
{
                                }
});
Toast.makeText(getApplicationContext()
ontext(), "Login Gagal, Mohon
Periksa Kembali",
Toast.LENGTH_SHORT).show();
```

Register.java

```
package com.putri.spk;

import
androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import
android.widget.ProgressBar;
import
android.widget.TextView;
import android.widget.Toast;

import
com.google.android.gms.tasks.OnCompleteListener;
import
com.google.android.gms.tasks.Task;
import
com.google.android.material.textfield.TextInputEditText;
import
com.google.firebase.auth.AuthResult;
import
com.google.firebase.auth.FirebaseAuth;
import
com.google.firebase.auth.FirebaseUser;
import
com.google.firebase.database.DataSnapshot;
import
com.google.firebase.database.DatabaseError;
```

```
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;
import
com.google.firebase.database.ValueEventListener;

public class Register extends
AppCompatActivity {
    TextInputEditText
etUsername, etEmail,
etPassword;
    Button btnReg;
    FirebaseAuth mAuth;
    ProgressBar progressBar;
    TextView tvLogin;
    DatabaseReference
database;

    @Override
    protected void
onCreate(Bundle
savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_register);

        etUsername =
findViewById(R.id.username);
        etEmail =
findViewById(R.id.email);
        etPassword =
findViewById(R.id.password);
        btnReg =
findViewById(R.id.btnRegister)
;
    }
}
```

```

        progressBar =
findViewById(R.id.progressBar)
;
        tvLogin =
findViewById(R.id.loginNow);
        mAuth =
FirebaseAuth.getInstance();

        database =
FirebaseDatabase.getInstance()
.getReferenceFromUrl("https://
spk-ewallet-firebase-default-
rtadb.firebaseio.com");

tvLogin.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void
onClick(View view) {
        Intent
regtolog = new
Intent(getApplicationContext()
, Login.class);

startActivity(regtolog);
        finish();
    }
});

btnReg.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void
onClick(View view) {
        database =
FirebaseDatabase.getInstance()
.getReference("user");

progressBar.setVisibility(View
.VISIBLE);

        String
username, email, password;
        username =
String.valueOf(etUsername.getT
ext());
        email =
String.valueOf(etEmail.getText
());
        password =
String.valueOf(etPassword.getT
ext());

        if
(TextUtils.isEmpty(username)){

```

```

Toast.makeText(getApplicationContext
ontext(), "Mohon Mengisi
Username Terlebih Dahulu",
Toast.LENGTH_SHORT).show();
        return;
    } else {

database.addListenerForSingleV
alueEvent(new
ValueEventListener() {

    @Override
        public
void onDataChange(@NonNull
DataSnapshot snapshot) {
        if
(snapshot.child(username).exis
ts()){

Toast.makeText(getApplicationContext
ontext(), "Username Telah
digunakan",
Toast.LENGTH_SHORT).show();
        }
    }

    @Override
        public
void onCancelled(@NonNull
DatabaseError error) {}
        });
        if
(TextUtils.isEmpty(email)){

Toast.makeText(getApplicationContext
ontext(), "Mohon Mengisi Email
Terlebih Dahulu",
Toast.LENGTH_SHORT).show();
        return;
    }
        if
(TextUtils.isEmpty(password)){

Toast.makeText(getApplicationContext
ontext(), "Mohon Mengisi
Password Terlebih Dahulu",
Toast.LENGTH_SHORT).show();
        return;
    } else {

database.addListenerForSingleV
alueEvent(new
ValueEventListener() {

```

```

@Override
    public
    void onDataChange(@NonNull
    DataSnapshot snapshot) {
        if
        (snapshot.child(password).exists()) {

            Toast.makeText(getApplicationContext()
            onText(), "Password Telah
            digunakan",
            Toast.LENGTH_SHORT).show();
        }

    }

@Override
    public
    void onCancelled(@NonNull
    DatabaseError error) {
        }
    });
}

 mAuth.createUserWithEmailAndPassword(email,
 password).addOnCompleteListener(new
 OnCompleteListener<AuthResult>
 () {

        @Override
        public
        void onComplete(@NonNull
        Task<AuthResult> task) {

            progressBar.setVisibility(View
            .GONE);

```

Main.activity

```

package com.putri.spk;

import
androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.DefaultItemAnimator;
import
androidx.recyclerview.widget.LinearLayoutManager;

```

```

        if
        (task.isSuccessful()){
            database.child(password).child
            ("username").setValue(username
            );
            database.child(password).child
            ("password").setValue(password
            );

            Toast.makeText(getApplicationContext()
            onText(), "Autentikasi
            Berhasil",
            Toast.LENGTH_SHORT).show();

            Intent regtolog = new
            Intent(getApplicationContext()
            , Login.class);

            startActivity(regtolog);

            finish();

        } else
        {
            Toast.makeText(getApplicationContext()
            onText(), "Autentikasi Gagal",
            Toast.LENGTH_SHORT).show();
        }
    });
}
}
}
}
}

```

```

import
androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import
android.widget.TextView;
import android.widget.Toast;

import
com.google.firebase.auth.FirebaseAuth;

```

```

import
com.google.firebase.auth.FirebaseAuth;
import
com.google.firebase.database.DatabaseSnapshot;
import
com.google.firebase.database.DatabaseError;
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;
import
com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class MainActivity
extends AppCompatActivity {
    private long backpress;
    private Toast backtoast;
    FirebaseAuth auth;

    Button logoutbtn, votebtn;
    TextView tvUser;
    FirebaseUser user;
    RecyclerView rvEwallet;
    DatabaseReference
database, udatabase;
    WalletAdapter adapter;
    String username, finish;
    ArrayList<Wallet>
arrayList;

    @Override
    protected void
onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        auth =
FirebaseAuth.getInstance();
        logoutbtn =
findViewById(R.id.btnLogout);
        votebtn =
findViewById(R.id.btnVote);

        tvUser =
findViewById(R.id.userDetail);
        rvEwallet =
findViewById(R.id.rvEwallet);
        user =
auth.getCurrentUser();
        database =
FirebaseDatabase.getInstance().getReference("wallet");
        udatabase =
FirebaseDatabase.getInstance().getReference("user");

        rvEwallet.setHasFixedSize(true);

        RecyclerView.LayoutManager
layoutManager = new
LinearLayoutManager(this);

        rvEwallet.setLayoutManager(layoutManager);

        rvEwallet.setItemAnimator(new
DefaultItemAnimator());

        tampilData();

        if
(getIntent().hasExtra("username")){
            username =
getIntent().getStringExtra("username");

            tvUser.setText("Hai " +
username);
        } else {

            tvUser.setText("Selamat Datang
" + user.getEmail());
        }

        if
(getIntent().hasExtra("finish")){
            votebtn.setVisibility(View.GONE);
        }

        logoutbtn.setOnClickListener(new
View.OnClickListener() {

```

```

        @Override
        public void
onClick(View view) {

    FirebaseAuth.getInstance().signOut();

        Intent mainlog
= new
Intent(getApplicationContext()
, Login.class);

    startActivity(mainlog);
        finish();
    }
});

udatabase.addListenerForSingle
ValueEvent(new
ValueEventListener() {
        @Override
        public void
onDataChange(@NonNull
DataSnapshot snapshot) {
            if
(snapshot.child(username).child("finish").exists()){

    votebtn.setVisibility(View.GONE);

            }
        @Override
        public void
onCancelled(@NonNull
DatabaseError error) {}
    });

    votebtn.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void
onClick(View view) {
            Intent mainvot
= new
Intent(getApplicationContext()
, ShopeepayVote.class);

    mainvot.putExtra("username",
username);

    startActivity(mainvot);
        finish();
    }
});

```

```

    }

    @Override
    public void
onBackPressed() {
        if (backpress + 2000 >
System.currentTimeMillis()){

    backtoast.cancel();

    super.onBackPressed();
        return;
    } else {
        backtoast =
Toast.makeText(getApplicationContext(), "Tekan dua kali Untuk
Keluar", Toast.LENGTH_LONG);
        backtoast.show();
    }

    backpress =
System.currentTimeMillis();

    FirebaseAuth.getInstance().signOut();
    }

    private void tampilData()
{

    database.addValueEventListener
(new ValueEventListener() {
        @Override
        public void
onDataChange(@NonNull
DataSnapshot snapshot) {
            arrayList =
new ArrayList<>();
            for
(DataSnapshot item :
snapshot.getChildren()){
                Wallet
wallet = new Wallet();

                wallet.setAppname(item.child("appname").getValue(String.class));

                wallet.setRating(item.child("rating").getValue(String.class));

                wallet.setPoin(item.child("poin").getValue(Double.class));

                arrayList.add(wallet);
            }

```

```

        adapter = new
WalletAdapter(arrayList,
MainActivity.this);

rvEwallet.setAdapter(adapter);

adapter.notifyDataSetChanged();
;

        }

        @Override
        public void
onCancelled(@NonNull
DatabaseError error) {

            }

        });
    }
}

```

LinkAjaVote.java

```

package com.putri.spk;

import
androidx.annotation.NonNull;
import
androidx.appcompat.app.AlertDi
alog;
import
androidx.appcompat.app.AppComp
atActivity;

import
android.content.DialogInterfac
e;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import
android.widget.AdapterView;
import android.widget.Button;
import android.widget.Spinner;

import
com.google.firebase.database.D
ataSnapshot;
import
com.google.firebase.database.D
atabaseError;
import
com.google.firebase.database.D
atabaseReference;

```

```

import
com.google.firebase.database.F
irebaseDatabase;
import
com.google.firebase.database.V
alueEventListener;

public class LinkajaVote
extends AppCompatActivity {
    Spinner spinTopup,
spinPromo, spinFitur,
spinLayanan;
    Button btnNext;
    String username;
    AlertDialog.Builder
builder;
    DatabaseReference
wdatabase, udatabase;

    @Override
    protected void
onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceSt
ate);

        setContentView(R.layout.activi
ty_linkaja_vote);

        spinTopup =
findViewById(R.id.spinTopup);
        spinPromo =
findViewById(R.id.spinPromo);
        spinFitur =
findViewById(R.id.spinFitur);
        spinLayanan =
findViewById(R.id.spinLayanan)
;

        btnNext =
findViewById(R.id.btnNext);

        if
(getIntent().hasExtra("usernam
e")){

            username =
getIntent().getStringExtra("us
ername");
        }

        wdatabase =
FirebaseDatabase.getInstance()
.getReferenceFromUrl("https://
spk-ewallet-firebase-default-
rtdb.firebaseio.com");

```



```

        udatabase =
        FirebaseDatabase.getInstance()
        .getReferenceFromUrl("https://
        spk-ewallet-firebase-default-
        rtadb.firebaseio.com");

        ArrayAdapter<CharSequence>
        arrayAdapter =
        ArrayAdapter.createFromResource
        (getApplicationContext(),
        R.array.nilai,
        android.R.layout.simple_spinner_
        item);

        arrayAdapter.setDropDownViewRe
        source(android.R.layout.simple
        _spinner_dropdown_item);

        spinTopup.setAdapter(arrayAdap
        ter);

        spinPromo.setAdapter(arrayAdap
        ter);

        spinFitur.setAdapter(arrayAdap
        ter);

        spinLayanan.setAdapter(arrayAd
        apter);

        btnNext.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void
            onClick(View view) {
                wdatabase =
                FirebaseDatabase.getInstance()
                .getReference("wallet");
                udatabase =
                FirebaseDatabase.getInstance()
                .getReference("user");
                String appName
                = "LinkAja";

                String
                temporary = "Temporary";
                String shopee
                = "ShopeePay";
                String ovo =
                "OVO";
                String gopay =
                "GoPay";
                String dana =
                "Dana";

                String finish
                = "finish";

                String topup =
                spinTopup.getSelectedItem().to
                String();

                Integer tpp =
                Integer.parseInt(topup);

                String promo =
                spinPromo.getSelectedItem().to
                String();

                Integer prm =
                Integer.parseInt(promo);

                String fitur =
                spinFitur.getSelectedItem().to
                String();

                Integer ftr =
                Integer.parseInt(fitur);

                String layanan
                =
                spinLayanan.getSelectedItem().
                toString();

                Integer lyn =
                Integer.parseInt(layanan);

                udatabase.addListenerForSingle
                ValueEvent(new
                ValueEventListener() {
                    @Override
                    public
                    void onDataChange(@NonNull
                    DataSnapshot snapshot) {

                        Integer shoTopup =
                        snapshot.child(temporary).chil
                        d("shoTopup").getValue(Integer
                        .class);

                        Integer shoPromo =
                        snapshot.child(temporary).chil
                        d("shoPromo").getValue(Integer
                        .class);

                        Integer shoFitur =
                        snapshot.child(temporary).chil
                        d("shoFitur").getValue(Integer
                        .class);

                        Integer shoLayanan =
                        snapshot.child(temporary).chil
                        d("shoLayanan").getValue(Integ
                        er.class);

                        Integer ovoTopup =
                        snapshot.child(temporary).chil
                        d("ovoTopup").getValue(Integer
                        .class);

```

```
Integer ovoPromo =  
snapshot.child(temporary).child  
("ovoPromo").getValue(Integer  
.class);
```

```
Integer ovoFitur =  
snapshot.child(temporary).child  
("ovoFitur").getValue(Integer  
.class);
```

```
Integer ovoLayanan =  
snapshot.child(temporary).child  
("ovoLayanan").getValue(Integer.  
class);
```

```
Integer goTopup =  
snapshot.child(temporary).child  
("goTopup").getValue(Integer.  
class);
```

```
Integer goPromo =  
snapshot.child(temporary).child  
("goPromo").getValue(Integer.  
class);
```

```
Integer goFitur =  
snapshot.child(temporary).child  
("goFitur").getValue(Integer.  
class);
```

```
Integer goLayanan =  
snapshot.child(temporary).child  
("goLayanan").getValue(Integer.  
class);
```

```
Integer daTopup =  
snapshot.child(temporary).child  
("daTopup").getValue(Integer.  
class);
```

```
Integer daPromo =  
snapshot.child(temporary).child  
("daPromo").getValue(Integer.  
class);
```

```
Integer daFitur =  
snapshot.child(temporary).child  
("daFitur").getValue(Integer.  
class);
```

```
Integer daLayanan =  
snapshot.child(temporary).child  
("daLayanan").getValue(Integer.  
class);
```

```
udatabase.child(username).child  
("shoTopup").setValue(shoTopu  
p);
```

```
udatabase.child(username).child  
("shoPromo").setValue(shoProm  
o);
```

```
udatabase.child(username).child  
("shoFitur").setValue(shoFitu  
r);
```

```
udatabase.child(username).child  
("shoLayanan").setValue(shoLa  
yanan);
```

```
udatabase.child(username).child  
("ovoTopup").setValue(ovoTopu  
p);
```

```
udatabase.child(username).child  
("ovoPromo").setValue(ovoProm  
o);
```

```
udatabase.child(username).child  
("ovoFitur").setValue(ovoFitu  
r);
```

```
udatabase.child(username).child  
("ovoLayanan").setValue(ovoLa  
yanan);
```

```
udatabase.child(username).child  
("goTopup").setValue(goTopup)  
;
```

```
udatabase.child(username).child  
("goPromo").setValue(goPromo)  
;
```

```
udatabase.child(username).child  
("goFitur").setValue(goFitur)  
;
```

```
udatabase.child(username).child  
("goLayanan").setValue(goLaya  
nan);
```

```
udatabase.child(username).child  
("daTopup").setValue(daTopup)  
;
```

```
udatabase.child(username).child  
("daPromo").setValue(daPromo)  
;
```

```

udatabase.child(username).child("daFitur").setValue(daFitur);

udatabase.child(username).child("daLayanan").setValue(daLayanan);

udatabase.child(username).child("linkTopup").setValue(tpp);

udatabase.child(username).child("linkPromo").setValue(prm);

udatabase.child(username).child("linkFitur").setValue(ftr);

udatabase.child(username).child("linkLayanan").setValue(lyn);

udatabase.child(username).child("finish").setValue(finish);
    }

    @Override
    public
void onCancelled(@NonNull DatabaseError error) {}
    });

wdatabase.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public
void onDataChange(@NonNull DataSnapshot snapshot) {

Integer shoTopup =
snapshot.child(temporary).child("shoTopup").getValue(Integer.class);

Integer shoPromo =
snapshot.child(temporary).child("shoPromo").getValue(Integer.class);

Integer shoFitur =
snapshot.child(temporary).child("shoFitur").getValue(Integer.class);

Integer shoLayanan =

```

```

snapshot.child(temporary).child("shoLayanan").getValue(Integer.class);

Integer ovoTopup =
snapshot.child(temporary).child("ovoTopup").getValue(Integer.class);

Integer ovoPromo =
snapshot.child(temporary).child("ovoPromo").getValue(Integer.class);

Integer ovoFitur =
snapshot.child(temporary).child("ovoFitur").getValue(Integer.class);

Integer ovoLayanan =
snapshot.child(temporary).child("ovoLayanan").getValue(Integer.class);

Integer goTopup =
snapshot.child(temporary).child("goTopup").getValue(Integer.class);

Integer goPromo =
snapshot.child(temporary).child("goPromo").getValue(Integer.class);

Integer goFitur =
snapshot.child(temporary).child("goFitur").getValue(Integer.class);

Integer goLayanan =
snapshot.child(temporary).child("goLayanan").getValue(Integer.class);

Integer daTopup =
snapshot.child(temporary).child("daTopup").getValue(Integer.class);

Integer daPromo =
snapshot.child(temporary).child("daPromo").getValue(Integer.class);

Integer daFitur =
snapshot.child(temporary).child

```

```
d("daFitur").getValue(Integer.  
class);
```

```
Integer daLayanan =  
snapshot.child(temporary).child(  
"daLayanan").getValue(Integer.  
class);
```

```
Double  
pangkatc1 = -0.25;
```

```
Double  
pangkatc2 = 0.20;
```

```
Double  
pangkatc3 = 0.20;
```

```
Double  
pangkatc4 = 0.20;
```

```
Double  
pangkatc5 = 0.15;
```

```
Integer linkadmin =  
snapshot.child(appName).child(  
"admin").getValue(Integer.class);
```

```
Integer linktopup =  
snapshot.child(appName).child(  
"topup").getValue(Integer.class);
```

```
Integer linkpromo =  
snapshot.child(appName).child(  
"promo").getValue(Integer.class);
```

```
Integer linkfitur =  
snapshot.child(appName).child(  
"fitur").getValue(Integer.class);
```

```
Integer linklayanan =  
snapshot.child(appName).child(  
"layanan").getValue(Integer.class);
```

```
Integer jumlinktopup =  
snapshot.child(appName).child(  
"jtopup").getValue(Integer.class);
```

```
Integer jumlinkpromo =  
snapshot.child(appName).child(  
"jpromo").getValue(Integer.class);
```

```
Integer jumlinkfitur =
```

```
snapshot.child(appName).child(  
"jfitur").getValue(Integer.class);
```

```
Integer jumlinklayanan =  
snapshot.child(appName).child(  
"jlayanan").getValue(Integer.class);
```

```
Integer totjumlinktopup =  
jumlinktopup + 1;
```

```
Integer totjumlinkpromo =  
jumlinkpromo + 1;
```

```
Integer totjumlinkfitur =  
jumlinkfitur + 1;
```

```
Integer totjumlinklayanan =  
jumlinklayanan + 1;
```

```
Integer totlinktopup =  
linktopup + tpp;
```

```
Integer totlinkpromo =  
linkpromo + prm;
```

```
Integer totlinkfitur =  
linkfitur + ftr;
```

```
Integer totlinklayanan =  
linklayanan + lyn;
```

```
Integer totalLinkTopup =  
totlinktopup /  
totjumlinktopup;
```

```
Integer totalLinkPromo =  
totlinkpromo /  
totjumlinkpromo;
```

```
Integer totalLinkFitur =  
totlinkfitur /  
totjumlinkfitur;
```

```
Integer totalLinkLayanan =  
totlinklayanan /  
totjumlinklayanan;
```

```
Double  
tladmin =  
Double.parseDouble(String.valueOf(linkadmin));
```

```
Double  
tltopup =  
Double.parseDouble(String.valueOf(totalLinkTopup));
```

```

                                Double
    tlpromo =
    Double.parseDouble(String.valu
eOf (totalLinkPromo));

                                Double
    tlfitur =
    Double.parseDouble(String.valu
eOf (totalLinkFitur));

                                Double
    tllayanan =
    Double.parseDouble(String.valu
eOf (totalLinkLayanan));

                                Double
    tlAdmin = Math.pow(tladmin,
pangkatc1);

                                Double
    tlTopup = Math.pow(tltopup,
pangkatc2);

                                Double
    tlPromo = Math.pow(tlpromo,
pangkatc3);

                                Double
    tlFitur = Math.pow(tlfitur,
pangkatc4);

                                Double
    tllayanan =
    Math.pow(tllayanan,
pangkatc5);

                                Double
    linkS = tlAdmin * tlTopup *
    tlPromo * tlFitur * tllayanan;

    wdatabase.child(appName).child
    ("topup").setValue(totlinktopu
p);

    wdatabase.child(appName).child
    ("promo").setValue(totlinkprom
o);

    wdatabase.child(appName).child
    ("fitur").setValue(totlinkfitu
r);

    wdatabase.child(appName).child
    ("layanan").setValue(totlinkla
yanan);

    wdatabase.child(appName).child
    ("jtopup").setValue(totjumlink
topup);

    wdatabase.child(appName).child
    ("jpromo").setValue(totjumlink
promo);

```

```

    wdatabase.child(appName).child
    ("jfitur").setValue(totjumlink
fitur);

    wdatabase.child(appName).child
    ("jlayanan").setValue(totjumli
nklayananan);

    Integer shoadmin =
    snapshot.child(shopee).child("
admin").getValue(Integer.class
);

    Integer shotopup =
    snapshot.child(shopee).child("
topup").getValue(Integer.class
);

    Integer shopromo =
    snapshot.child(shopee).child("
promo").getValue(Integer.class
);

    Integer shofitur =
    snapshot.child(shopee).child("
fitur").getValue(Integer.class
);

    Integer sholayanan =
    snapshot.child(shopee).child("
layanan").getValue(Integer.cla
ss);

    Integer jumshotopup =
    snapshot.child(shopee).child("
jtopup").getValue(Integer.clas
s);

    Integer jumshopromo =
    snapshot.child(shopee).child("
jpromo").getValue(Integer.clas
s);

    Integer jumshofitur =
    snapshot.child(shopee).child("
jfitur").getValue(Integer.clas
s);

    Integer jumsholayanan =
    snapshot.child(shopee).child("
jlayanan").getValue(Integer.cl
ass);

```

```

Integer totjumshotopup =
jumshotopup + 1;

Integer totjumshopromo =
jumshopromo + 1;

Integer totjumshofitur =
jumshofitur + 1;

Integer totjumsholayanan =
jumsholayanan + 1;

Integer totshotopup = shotopup
+ shoTopup;

Integer totshopromo = shopromo
+ shoPromo;

Integer totshofitur = shofitur
+ shoFitur;

Integer totsholayanan =
sholayanan + shoLayanan;

Integer totalShoTopup =
totshotopup / totjumshotopup;

Integer totalShoPromo =
totshopromo / totjumshopromo;

Integer totalShoFitur =
totshofitur / totjumshofitur;

Integer totalShoLayanan =
totsholayanan /
totjumsholayanan;

Double
tsadmin =
Double.parseDouble(String.valu
eOf(shoadmin));

Double
tstopup =
Double.parseDouble(String.valu
eOf(totalShoTopup));

Double
tspromo =
Double.parseDouble(String.valu
eOf(totalShoPromo));

Double
tsfitur =
Double.parseDouble(String.valu
eOf(totalShoFitur));

Double
tslayanan =

```

```

Double.parseDouble(String.valu
eOf(totalShoLayanan));

Double
tsAdmin = Math.pow(tsadmin,
pangkatc1);

Double
tsTopup = Math.pow(tstopup,
pangkatc2);

Double
tsPromo = Math.pow(tspromo,
pangkatc3);

Double
tsFitur = Math.pow(tsfitur,
pangkatc4);

Double
tsLayanan =
Math.pow(tslayanan,
pangkatc5);

Double
shoS = tsAdmin * tsTopup *
tsPromo * tsFitur * tsLayanan;

wdatabase.child(shopee).child(
"topup").setValue(totshotopup)
;

wdatabase.child(shopee).child(
"promo").setValue(totshopromo)
;

wdatabase.child(shopee).child(
"fitur").setValue(totshofitur)
;

wdatabase.child(shopee).child(
"layanan").setValue(totsholaya
nan);

wdatabase.child(shopee).child(
"jtopup").setValue(totjumshoto
pup);

wdatabase.child(shopee).child(
"jpromo").setValue(totjumshopr
omo);

wdatabase.child(shopee).child(
"jfitur").setValue(totjumshofi
tur);

wdatabase.child(shopee).child(
"jlayanan").setValue(totjumsho
layanan);

Integer ovoidadmin =

```

```
snapshot.child(ovo).child("admin").getValue(Integer.class);
```

```
Integer ovotopup =  
snapshot.child(ovo).child("topup").getValue(Integer.class);
```

```
Integer ovopromo =  
snapshot.child(ovo).child("promo").getValue(Integer.class);
```

```
Integer ovofitur =  
snapshot.child(ovo).child("fitur").getValue(Integer.class);
```

```
Integer ovolayanan =  
snapshot.child(ovo).child("layananan").getValue(Integer.class);  
;
```

```
Integer jumovotopup =  
snapshot.child(ovo).child("jtopup").getValue(Integer.class);
```

```
Integer jumovopromo =  
snapshot.child(ovo).child("jpromo").getValue(Integer.class);
```

```
Integer jumovofitur =  
snapshot.child(ovo).child("jfitur").getValue(Integer.class);
```

```
Integer jumovolayanan =  
snapshot.child(ovo).child("jlayananan").getValue(Integer.class);  
;
```

```
Integer totjumovotopup =  
jumovotopup + 1;
```

```
Integer totjumovopromo =  
jumovopromo + 1;
```

```
Integer totjumovofitur =  
jumovofitur + 1;
```

```
Integer totjumovolayanan =  
jumovolayanan + 1;
```

```
Integer totovotopup = ovotopup  
+ ovoTopup;
```

```
Integer totovopromo = ovopromo  
+ ovoPromo;
```

```
Integer totovofitur = ovofitur  
+ ovoFitur;
```

```
Integer totovolayanan =  
ovolayanan + ovoLayanan;
```

```
Integer totalOvoTopup =  
totovotopup / totjumovotopup;
```

```
Integer totalOvoPromo =  
totovopromo / totjumovopromo;
```

```
Integer totalOvoFitur =  
totovofitur / totjumovofitur;
```

```
Integer totalOvoLayanan =  
totovolayanan /  
totjumovolayanan;
```

```
Double  
toadmin =  
Double.parseDouble(String.valueOf(ovoadmin));
```

```
Double  
totopup =  
Double.parseDouble(String.valueOf(totalOvoTopup));
```

```
Double  
topromo =  
Double.parseDouble(String.valueOf(totalOvoPromo));
```

```
Double  
tofitur =  
Double.parseDouble(String.valueOf(totalOvoFitur));
```

```
Double  
tolayanan =  
Double.parseDouble(String.valueOf(totalOvoLayanan));
```

```
Double  
toAdmin = Math.pow(toadmin,  
pangkatc1);
```

```
Double  
toTopup = Math.pow(totopup,  
pangkatc2);
```

```
Double  
toPromo = Math.pow(topromo,  
pangkatc3);
```

```
Double  
toFitur = Math.pow(tofitur,  
pangkatc4);
```

```
Double  
toLayanan =  
Math.pow(tolayanan,  
pangkatc5);
```

```

                                Double
ovoS = toAdmin * toTopup *
toPromo * toFitur * toLayanan;

wdatabase.child(ovo).child("to
pup").setValue(totovotopup);

wdatabase.child(ovo).child("pr
omo").setValue(totovopromo);

wdatabase.child(ovo).child("fi
tur").setValue(totovofitur);

wdatabase.child(ovo).child("la
yanan").setValue(totovolayanan
);

wdatabase.child(ovo).child("jt
opup").setValue(totjumovotopup
);

wdatabase.child(ovo).child("jp
romo").setValue(totjumovopromo
);

wdatabase.child(ovo).child("jf
itur").setValue(totjumovofitur
);

wdatabase.child(ovo).child("jl
ayanan").setValue(totjumovolay
anan);

Integer goadmin =
snapshot.child(gopay).child("a
dmin").getValue(Integer.class)
;

Integer gotopup =
snapshot.child(gopay).child("t
opup").getValue(Integer.class)
;

Integer gopromo =
snapshot.child(gopay).child("p
romo").getValue(Integer.class)
;

Integer gofitur =
snapshot.child(gopay).child("f
itur").getValue(Integer.class)
;

Integer golayanan =
snapshot.child(gopay).child("l

```

```

ayanan").getValue(Integer.clas
s);

Integer jumgotopup =
snapshot.child(gopay).child("j
topup").getValue(Integer.class
);

Integer jumgopromo =
snapshot.child(gopay).child("j
promo").getValue(Integer.class
);

Integer jumgofitur =
snapshot.child(gopay).child("j
fitur").getValue(Integer.class
);

Integer jumgolayanan =
snapshot.child(gopay).child("j
layanan").getValue(Integer.cla
ss);

Integer totjumgotopup =
jumgotopup + 1;

Integer totjumgopromo =
jumgopromo + 1;

Integer totjumgofitur =
jumgofitur + 1;

Integer totjumgolayanan =
jumgolayanan + 1;

Integer totgotopup = gotopup +
goTopup;

Integer totgopromo = gopromo +
goPromo;

Integer totgofitur = gofitur +
goFitur;

Integer totgolayanan =
golayanan + goLayanan;

Integer totalGoTopup =
totgotopup / totjumgotopup;

Integer totalGoPromo =
totgopromo / totjumgopromo;

Integer totalGoFitur =
totgofitur / totjumgofitur;

```



```

Integer totalGoLayanan =
totgolayanan /
totjumgolayanan;

                                Double
tgadmin =
Double.parseDouble(String.valu
eOf(goadmin));

                                Double
tgtopup =
Double.parseDouble(String.valu
eOf(totalGoTopup));

                                Double
tgpromo =
Double.parseDouble(String.valu
eOf(totalGoPromo));

                                Double
tgfitur =
Double.parseDouble(String.valu
eOf(totalGoFitur));

                                Double
tglayananan =
Double.parseDouble(String.valu
eOf(totalGoLayanan));

                                Double
tgAdmin = Math.pow(tgadmin,
pangkatc1);

                                Double
tgTopup = Math.pow(tgtopup,
pangkatc2);

                                Double
tgPromo = Math.pow(tgpromo,
pangkatc3);

                                Double
tgFitur = Math.pow(tgfitur,
pangkatc4);

                                Double
tgLayanan =
Math.pow(tglayananan,
pangkatc5);

                                Double
goS = tgAdmin * tgTopup *
tgPromo * tgFitur * tgLayanan;

wdatabase.child(gopay).child("
topup").setValue(totgotopup);

wdatabase.child(gopay).child("
promo").setValue(totgopromo);

wdatabase.child(gopay).child("
fitur").setValue(totgofitur);

wdatabase.child(gopay).child("
layananan").setValue(totgolayana
n);

wdatabase.child(gopay).child("
jtopup").setValue(totjumgotopu
p);

wdatabase.child(gopay).child("
jpromo").setValue(totjumgoprom
o);

wdatabase.child(gopay).child("
jfitur").setValue(totjumgofitu
r);

wdatabase.child(gopay).child("
jlayananan").setValue(totjumgola
yanan);

Integer danadmin =
snapshot.child(dana).child("ad
min").getValue(Integer.class);

Integer dantopup =
snapshot.child(dana).child("to
pup").getValue(Integer.class);

Integer danpromo =
snapshot.child(dana).child("pr
omo").getValue(Integer.class);

Integer danfitur =
snapshot.child(dana).child("fi
tur").getValue(Integer.class);

Integer danlayananan =
snapshot.child(dana).child("la
yanan").getValue(Integer.class
);

Integer jumdantopup =
snapshot.child(dana).child("jt
opup").getValue(Integer.class
);

Integer jumdanpromo =
snapshot.child(dana).child("jp
romo").getValue(Integer.class
);

Integer jumdanfitur =
snapshot.child(dana).child("jf
itur").getValue(Integer.class
);

Integer jumdanlayananan =
snapshot.child(dana).child("jl

```

```

ayanan").getValue(Integer.class);

Integer totjumdantopup =
jumdantopup + 1;

Integer totjumdanpromo =
jumdanpromo + 1;

Integer totjumdanfitur =
jumdanfitur + 1;

Integer totjumdanlayananan =
jumdanlayananan + 1;

Integer totdantopup = dantopup
+ daTopup;

Integer totdanpromo = danpromo
+ daPromo;

Integer totdanfitur = danfitur
+ daFitur;

Integer totdanlayananan =
danlayananan + daLayanan;

Integer totalDaTopup =
totdantopup / totjumdantopup;

Integer totalDaPromo =
totdanpromo / totjumdanpromo;

Integer totalDaFitur =
totdanfitur / totjumdanfitur;

Integer totalDaLayanan =
totdanlayananan /
totjumdanlayananan;

Double
tdadmin =
Double.parseDouble(String.value
Of(danadmin));

Double
tdtopup =
Double.parseDouble(String.value
Of(totalDaTopup));

Double
tdpromo =
Double.parseDouble(String.value
Of(totalDaPromo));

Double
tdfitur =
Double.parseDouble(String.value
Of(totalDaFitur));

Double
tdlayananan =
Double.parseDouble(String.value
Of(totalDaLayanan));

Double
tdAdmin = Math.pow(tdadmin,
pangkatc1);

Double
tdTopup = Math.pow(tdtopup,
pangkatc2);

Double
tdPromo = Math.pow(tdpromo,
pangkatc3);

Double
tdFitur = Math.pow(tdfitur,
pangkatc4);

Double
tdLayanan =
Math.pow(tdlayananan,
pangkatc5);

Double
danS = tdAdmin * tdTopup *
tdPromo * tdfitur * tdLayanan;

wdatabase.child(dana).child("t
opup").setValue(totdantopup);

wdatabase.child(dana).child("p
romo").setValue(totdanpromo);

wdatabase.child(dana).child("f
itur").setValue(totdanfitur);

wdatabase.child(dana).child("l
ayanan").setValue(totdanlayana
n);

wdatabase.child(dana).child("j
topup").setValue(totjumdantopu
p);

wdatabase.child(dana).child("j
promo").setValue(totjumdanprom
o);

wdatabase.child(dana).child("j
fitur").setValue(totjumdanfitu
r);

wdatabase.child(dana).child("j
layananan").setValue(totjumdanla
yanan);

Double
totalS = linkS + shoS + ovoS +
goS + danS;

```

```

                                Double
linkPoin = linkS / totals;
                                Double
shoPoin = shoS / totals;
                                Double
ovoPoin = ovoS / totals;
                                Double
goPoin = goS / totals;
                                Double
danPoin = danS / totals;

wdatabase.child(shopee).child(
"poin").setValue(shoPoin);

wdatabase.child(ovo).child("po
in").setValue(ovoPoin);

wdatabase.child(gopay).child("
poin").setValue(goPoin);

wdatabase.child(dana).child("p
oin").setValue(danPoin);

wdatabase.child(appName).child(
"poin").setValue(linkPoin);

                                Intent
linkmain = new
Intent(getApplicationContext()
, MainActivity.class);

linkmain.putExtra("username",
username);

linkmain.putExtra("finish",
finish);

startActivity(linkmain);

finish();

                                }

                                @Override
                                public
void onCancelled(@NonNull
DatabaseError error) {}

                                });

                                }

                                @Override
                                public void
onBackPressed() {

```

```

                                Intent linda = new
Intent(getApplicationContext()
, DanaVote.class);

linda.putExtra("username",
username);
                                startActivity(linda);
                                finish();
                                }
                                }

```

WalletAdapter.java

```

package com.putri.spk;

import
android.content.Context;
import
android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import
android.widget.TextView;

import
androidx.annotation.NonNull;
import
androidx.recyclerview.widget.R
ecyclerView;

import java.util.List;

public class WalletAdapter
extends
RecyclerView.Adapter<WalletAda
pter.MyViewHolder> {
    List<Wallet> wlist;
    Context context;
    public
WalletAdapter(List<Wallet>
wlist, Context context){
        this.wlist = wlist;
        this.context =
context;
    }

    @NonNull
    @Override
    public
WalletAdapter.MyViewHolder
onCreateViewHolder(@NonNull
ViewGroup parent, int
viewType) {
        View view =
LayoutInflater.from(parent.get

```

```

Context()).inflate(R.layout.li
st_ewallet, parent, false);
        return new
MyViewHolder(view);
    }

    @Override
    public void
onBindViewHolder(@NonNull
WalletAdapter.MyViewHolder
holder, int position) {
        final Wallet item =
wlist.get(position);

holder.tvAppname.setText("Nama
Aplikasi : " +
item.getAppname());

holder.tvRating.setText("Ratin
g Aplikasi : " +
item.getRating());

holder.tvPoin.setText("Poin
Aplikasi : " +
item.getPoin());
    }

    @Override
    public int getItemCount()
{
        return wlist.size();
    }

    public class MyViewHolder
extends
RecyclerView.ViewHolder {
        private TextView
tvAppname, tvRating, tvPoin;
        public
MyViewHolder(@NonNull View
itemView) {
            super(itemView);

            tvAppname =
itemView.findViewById(R.id.tvA
ppname);

            tvRating =
itemView.findViewById(R.id.tvR
ating);

            tvPoin =
itemView.findViewById(R.id.tvP
oin);
        }
    }
}

```

Lampiran 2

Tabel Alternatif

Alternatif	Kode
OVO	R1
GoPay	R2
DANA	R3
LinkAja	R4
ShopeePay	R5

Tabel Kriteria

Kriteria	Bobot	Cost/Benefit	Kode
Biaya Admin	5	Cost	C1
Kemudahan Top Up	4	Benefit	C2
Promo	4	Benefit	C3
Fitur	4	Benefit	C4
Layanan	3	Benefit	C5
Jumlah	20		

Bobot	Kriteria				
	C1	C2	C3	C4	C5
Bobot Kepentingan	0,25	0,2	0,2	0,2	0,15

$\sum w_j$

1

Alternatif	Kriteria				
	C1	C2	C3	C4	C5
R1	1200	3,82	3,32	3,45	3,27
R2	2000	4,05	3,55	3,64	3,32
R3	2500	4,59	3,82	4,09	3,55
R4	1500	3,36	2,91	2,86	3,18
R5	2000	4,41	4,32	4,18	4,23

Pangkat	-0,25	0,2	0,2	0,2	0,15
Alternatif	S				
R1	0,43211				
R2	0,39507				
R3	0,40195				

R4	0,37210
R5	0,44557
Jumlah	2,0468

Alternatif	V
R1	0,21111
R2	0,19302
R3	0,19638
R4	0,18180
R5	0,21769
Jumlah	1

Alternatif	V	Rangking
R1	0,21111	2
R2	0,19302	4
R3	0,19638	3
R4	0,18180	5
R5	0,21769	1
Jumlah	1	

Lampiran 3

No.	User	DaFitur	DaLayanan	DaPromo	DaTopup	finish	GoFitur	GoLayanan	GoPromo	GoTopup	LinkFitur	LinkLayanan	LinkPromo	LinkTopup	OvoFitur	OvoLayanan	OvoPromo	OvoTopup	ShoFitur	ShoLayanan	ShoPromo	ShoTopup
1	Aa Mahdhor	5	5	5	5	finish	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3
2	Aan anggraeni	3	3	3	4	finish	3	4	4	3	2	1	3	3	2	3	2	3	3	3	3	3
3	Amelia Maulinda	4	4	3	4	finish	3	3	3	4	4	3	3	3	3	3	3	3	4	3	3	4
4	Chandra M	3	2	3	5	finish	5	2	3	5	2	4	3	5	2	3	2	4	4	5	5	4
5	Heni Sonia	3	1	3	3	finish	4	2	3	2	5	5	5	5	5	2	4	5	3	2	3	2
6	Laela Latifah	4	3	3	4	finish	3	3	4	4	3	3	3	4	3	3	3	4	4	4	4	4
7	Maharani Putri	5	1	3	5	finish	4	2	2	4	2	3	1	2	4	4	4	3	3	3	3	4
8	Neneng	2	2	3	3	finish	1	4	2	5	3	3	3	3	3	2	3	3	4	3	2	4
9	Nurhela	3	3	3	3	finish	3	3	4	4	2	2	1	2	4	3	3	5	3	4	3	5
10	Putri Maharani	3	3	3	5	finish	4	4	3	5	3	3	3	3	1	3	3	4	5	5	5	5
11	Siti Rusnanti	5	5	5	5	finish	3	3	3	4	3	3	3	4	4	4	4	5	4	4	5	5
12	Susilawati	3	2	3	5	finish	3	2	3	5	1	2	2	5	3	3	4	5	5	5	5	5
13	Temporary	3	3	3	4		3	4	4	3					2	3	2	3	3	3	3	3
14	adi saprudin	4	3	4	5	finish	4	3	3	2	2	3	3	3	4	3	3	2	4	3	5	3
15	alfarizi	4	4	4	4	finish	3	3	4	4	2	3	2	3	3	3	3	3	4	4	5	3
16	dara	3	3	3	3	finish	4	4	4	4	3	3	3	3	3	3	3	3	3	4	4	4
17	dewi	5	5	5	5	finish	5	4	5	5	3	3	3	3	5	5	5	5	4	4	4	4
18	diva	4	3	2	4	finish	4	4	4	4	2	3	2	3	3	3	1	3	4	4	4	4
19	jimmy	3	3	2	4	finish	3	2	3	4	2	2	2	3	3	1	2	3	5	5	4	5
20	mujahidin	5	5	5	5	finish	3	4	3	4	4	5	3	2	3	3	3	3	3	4	3	4
21	puput	3	3	3	3	finish	1	1	1	1	1	1	1	1	1	1	1	1	3	3	3	3
22	riniaryani910	3	4	3	4	finish	3	3	3	3	3	3	3	3	3	3	3	3	3	4	4	4
23	winda ningsih	4	4	4	4	finish	4	4	4	4	3	3	3	3	4	4	4	4	5	5	5	5

No.	Wallet	Biaya Admin	Tfitur	Jfitur	Jlayanan	Jpromo	Jtopup	Tlayanan	Poin	Tpromo	Rating	Ttopup
1.	OVO	1200	76	22	22	22	22	72	0,211172	73	4,3	84
2.	GoPay	2000	80	22	22	22	22	73	0,192887	78	4,5	89
3.	DANA	2500	90	22	22	22	22	78	0,196357	84	4,2	101
4.	LinkAja	1500	63	22	22	22	22	70	0,181903	64	3,7	74
5.	ShopeePay	2000	92	22	22	22	22	93	0,217681	95	4,5	97

