

SKRIPSI

**APLIKASI MANAJEMEN KEUANGAN PRIBADI
BERBASIS *MOBILE* MENGGUNAKAN *FRAMEWORK*
*FLUTTER***



SR-063

21040039

ILHAMMULLAH

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS AL-KHAIRIYAH

2024 – 2025

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : ILHAMMULLAH
NIM : 21040039
Program studi : Teknik Informatika
Fakultas : Ilmu Komputer
Jenjang Pendidikan : Strata 1

Menyatakan bahwa skripsi yang saya buat dengan judul :

**“APLIKASI MANAJEMEN KEUANGAN PRIBADI
BERBASIS *MOBILE* MENGGUNAKAN *FRAMEWORK*
FLUTTER”**

Adalah benar hasil karya tulis ilmiah sendiri, bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.

Pernyataan ini saya buat dengan penuh kesadaran dan tanggung jawab. Dan saya bersedia menerima konsekuensi apapun sesuai aturan yang berlaku apabila dikemudian hari pernyataan ini tidak benar.

Cilegon, 18 Juni 2025

Ilhammullah

PERSETUJUAN

APLIKASI MANAJEMEN KEUANGAN PRIBADI BERBASIS *MOBILE* MENGGUNAKAN *FRAMEWORK* *FLUTTER*

NAMA : ILHAMMULLAH
NIM : 21040039
PROGRAM STUDI : TEKNIK INFORMATIKA

Cilegon, 5 Juni 2025

Telah Diperiksa dan Disetujui

Pada Tanggal : 5 Juni 2025

Pembimbing I

Pembimbing II

Tubagus Mochamad Isnaeni, M.TI

NIDN. 0403099003

Ainin Najmi, M.Kom

NIDN. 0404048903

PENGESAHAN

APLIKASI MANAJEMEN KEUANGAN PRIBADI BERBASIS *MOBILE* MENGGUNAKAN *FRAMEWORK* *FLUTTER*

NAMA : ILHAMMULLAH
NIM : 21040039
PROGRAM STUDI : TEKNIK INFORMATIKA

Cilegon, 18 Juni 2025

**Telah Diperiksa dan Disetujui
Pada Tanggal : 18 Juni 2025**

Penguji I

Penguji II

Rulin Swastika, M.Kom

NIDN. 0431058401

Teuku Fadjar Shadek, S.Kom., M.Kom

NIDN. 0421048404

PENGESAHAN

APLIKASI MANAJEMEN KEUANGAN PRIBADI BERBASIS *MOBILE* MENGGUNAKAN *FRAMEWORK* *FLUTTER*

NAMA : ILHAMMULLAH

NIM : 21040039

PROGRAM STUDI : TEKNIK INFORMATIKA

Cilegon, 18 Juni 2025

Telah Diperiksa dan Disetujui

Pada Tanggal : 18 Juni 2025

Dekan

Fakultas Ilmu Komputer

Ketua Program Studi

Teknik Informatika

Roy Amrullah Ritonga, M.Kom

NIDN. 0423038304

Didda Rahayu Yuliana, M.Kom

NIDN. 0426078601

MOTTO

*Tak selalu dia yang pintar yang menyelesaikan,
Tapi dia yang bertahan dalam keterbatasan.
Tak selalu dia yang cepat yang sampai tujuan,
Tapi dia yang terus melangkah meski perlahan.*

*“Sukses bukan milik orang pintar, tapi milik mereka yang mau berusaha.”
– B.J. Habibie*

*Dalam lelah, aku belajar setia,
Dalam gagal, aku belajar makna usaha.
Dalam skripsi ini, tak ada jalan mulus tanpa luka,
Tapi justru di situ kutemukan maknanya.*

*"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya."
(QS. Al-Baqarah: 286)*

*Terima kasih untuk setiap kesalahan yang mengajarku,
Untuk setiap revisi yang menguatkan,
Untuk setiap error yang membuatku berpikir ulang,
Dan untuk setiap ‘simpan’ yang membuat semuanya tenang.*

*Karena pada akhirnya,
Yang menentukan bukan seberapa bisa,
Tapi seberapa terbiasa.*

*Dan seberapa besar keinginan untuk terus melangkah meski tak sempurna.
Kerja keras mungkin melelahkan,
tapi penyesalan karena menyerah jauh lebih menyakitkan.*

PERSEMBAHAN

ALHAMDULILLAH RABBIL ‘ALAMIN,
BERKAT RAHMAT DAN RIDHO ALLAH SWT, AKHIRNYA SAYA DAPAT
MENYELESAIKAN KARYA INI.

DENGAN SEGALA KERENDAHAN HATI, KARYA SEDERHANA INI SAYA
PERSEMBAHKAN KEPADA:

ALLAH SWT,
SUMBER KEKUATAN, KETENANGAN, DAN HARAPAN DALAM SETIAP
LANGKAH. TIADA DAYA DAN UPAYA SELAIN ATAS IZIN DAN
PERTOLONGAN-MU. SEGALA PROSES INI BERJALAN KARENA
KEMURAHAN-MU.

ORANG TUA TERCINTA,
TERIMA KASIH ATAS DOA YANG TAK PERNAH PUTUS, CINTA YANG
TAK PERNAH BERSYARAT, DAN DUKUNGAN YANG MENJADI PIJAKAN
SAYA UNTUK TERUS MELANGKAH. TANPA RESTU DAN
PENGORBANAN KALIAN, SAYA TIDAK AKAN SAMPAI SEJAUH INI.

SAUDARA-SAUDARAKU,
YANG SELALU MENJADI TEMAN CERITA, PENGINGAT SEMANGAT,
DAN SUMBER TAWA DI TENGAH LEHAH.

DOSEN PEMBIMBING DAN PARA PENGAJAR,
TERIMA KASIH ATAS BIMBINGAN, ILMU, DAN KESABARAN YANG
TELAH MENGANTARKAN SAYA HINGGA TITIK INI. SETIAP KOREKSI
DAN ARAHAN MENJADI PELITA DALAM GELAPNYA PROSES INI.

TEMAN-TEMAN SEPERJUANGAN,
YANG TURUT BERJUANG DALAM SENYAP, SALING MENGUATKAN
MESKI SAMA-SAMA LETIH. TERIMA KASIH TELAH HADIR DALAM
PERJALANAN AKADEMIK INI.

DIRIKU SENDIRI,
TERIMA KASIH TELAH BERTAHAN, MESKI KADANG INGIN
MENYERAH. TELAH MEMILIH UNTUK BANGKIT, BELAJAR, DAN
MENYELESAIKAN APA YANG SUDAH DIMULAI.

SEMOGA ALLAH SWT MEMBALAS SEMUA KEBAIKAN DENGAN
BALASAN YANG TERBAIK DAN INDAH.

AAMIIN.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul ***“Aplikasi Manajemen Keuangan Pribadi Berbasis Mobile Menggunakan Framework Flutter”***. Skripsi ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Al-Khairiyah.

Dalam proses penyusunan skripsi ini, penulis menyadari bahwa keberhasilan ini tidak terlepas dari bantuan, dukungan, dan bimbingan berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Roy Amrullah Ritonga, M.Kom, selaku Dekan Fakultas Ilmu Komputer, yang telah memberikan arahan dan dukungan selama masa studi penulis di universitas ini.
2. Didda Rahayu Yuliana, M.Kom, selaku Kepala Program Studi Teknik Informatika, atas bimbingan dan kebijaksanaan yang diberikan selama proses perkuliahan hingga penyusunan skripsi ini.
3. Tubagus Mochamad Isnaeni, M.TI, selaku Dosen Pembimbing I, yang telah memberikan bimbingan, arahan, dan masukan berharga selama proses penyusunan skripsi ini.
4. Ainin Najmi, M.Kom, selaku Dosen Pembimbing II, atas dukungan dan saran yang sangat membantu dalam memperbaiki dan menyempurnakan karya ini.
5. Kedua orang tua, yang telah memberikan doa, kasih sayang, dan dukungan moral maupun material yang tiada henti. Keberhasilan ini adalah wujud dari pengorbanan dan cinta kalian.
6. Teman-teman seperjuangan di Teknik Informatika, yang telah memberikan dukungan, semangat, dan kebersamaan yang sangat berarti selama masa perkuliahan dan penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis terbuka terhadap kritik dan saran yang membangun untuk menyempurnakan karya ini di masa depan.

Akhir kata, penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang teknologi informasi, dan memberikan kontribusi nyata bagi masyarakat.

Cilegon, 18 Juni 2025

Ilhammullah



ABSTRAKSI

Ilhammullah “Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* Menggunakan *Framework Flutter*”. Dosen Pembimbing I Tubagus Mochamad Isnaeni, M.TI. Dosen Pembimbing II Ainin Najmi, M.Kom. Tahun 2024-2025 Skripsi 105 Halaman. Pengelolaan keuangan pribadi merupakan aspek penting dalam kehidupan sehari-hari. Banyak individu mengalami kesulitan dalam mencatat pemasukan, pengeluaran, dan alokasi dana secara terstruktur. Penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi mobile berbasis Flutter yang dapat membantu pengguna dalam mengelola keuangan pribadi secara efisien. Aplikasi ini memiliki fitur pencatatan transaksi harian, kategori transaksi, rekap bulanan dan tahunan, serta ekspor data ke dalam bentuk PDF. Metode pengembangan perangkat lunak yang digunakan adalah metode Waterfall. Pengujian aplikasi dilakukan menggunakan metode black box dan melibatkan beberapa pengguna untuk uji coba penggunaan. Dengan pendekatan minimalis, aplikasi ini tidak memerlukan konektivitas *online*, sehingga dapat digunakan secara *offline* dan memastikan privasi data pengguna. Sistem database yang digunakan adalah *SQLite*, yang ringan dan cocok untuk aplikasi *Mobile*. Hasil pengujian menunjukkan bahwa aplikasi dapat berjalan dengan baik dan fitur-fitur yang disediakan dapat membantu pengguna dalam mengatur keuangannya secara mandiri. Aplikasi ini diharapkan dapat memberikan solusi praktis bagi individu yang membutuhkan alat manajemen keuangan sederhana, serta menjadi landasan untuk pengembangan fitur yang lebih kompleks di masa mendatang.

Kata Kunci: Manajemen Keuangan, Flutter, Aplikasi Mobile, Transaksi, PDF

ABSTRACT

Ilhammullah "Mobile-Based Personal Financial Management Application Using Flutter Framework". Supervisor I Tubagus Mochamad Isnaeni, M.TI. Supervisor II Ainin Najmi, M.Kom. 2024-2025 Thesis 105 Pages. Personal financial management is an important aspect of everyday life. Many individuals have difficulty recording income, expenses, and fund allocations in a structured manner. This study aims to design and build a Flutter-based mobile application that can help users manage their personal finances efficiently. This application has features for recording daily transactions, transaction categories, monthly and annual recaps, and exporting data to PDF. The software development method used is the Waterfall method. Application testing is carried out using the black box method and involves several users for trial use. With a minimalist approach, this application does not require online connectivity, so it can be used offline and ensures the privacy of user data. The database system used is SQLite, which is lightweight and suitable for mobile applications. The test results show that the application can run well and the features provided can help users manage their finances independently. This application is expected to provide a practical solution for individuals who need a simple financial management tool, as well as being a foundation for the development of more complex features in the future.

Keywords: Financial Management, Flutter, Mobile Application, Transaction, PDF

DAFTAR ISI

Halaman

LEMBAR JUDUL	i
LEMBAR PERNYATAAN	ii
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PENGESAHAN	iv
MOTTO	v
PERSEMBAHAN.....	vi
KATA PENGANTAR.....	vii
ABSTRAKSI.....	ix
<i>ABSTRACT</i>	x
DAFTAR ISI.....	x
DAFTAR GAMBAR	xv
DAFTAR TABEL	xviii
DAFTAR SIMBOL	xix
DAFTAR LAMPIRAN	xxii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Identifikasi Masalah	2
1.3 Rumusan Masalah	2
1.4 Batasan Masalah.....	3
1.5 Tujuan Penelitian.....	4
1.6 Manfaat Penelitian	4
1.7 Metode Penelitian.....	5

1.8 Jadwal Kegiatan Penelitian	11
1.9 Sistematika Penulisan.....	11
BAB II TINJAUAN PUSTAKAN DAN LANDASAN TEORI	13
2.1 Tinjauan Pustaka	13
2.2 Landasan Teori	16
2.2.1 Manajemen Keuangan.....	16
2.2.2 Aplikasi	20
2.2.3 Android.....	20
2.2.4 Android Studio	21
2.2.5 Framework Flutter.....	22
2.2.6 Figma.....	24
2.2.7 Tentang Aplikasi.....	25
2.2.8 Model Pengembangan Sistem.....	26
BAB III ANALISA DAN PERANCANGAN	32
3.1 Analisa Sistem.....	32
3.1.1 Identifikasi Kebutuhan Sistem	32
3.1.2 Analisis Proses	34
3.1.3 Analisis Struktur Data	36
3.1.4 Analisis Antarmuka Pengguna	36
3.1.5 Analisis Kinerja Sistem	37
3.1.6 Sistem yang Berjalan.....	37
3.1.7 Sistem yang diusulkan.....	38
3.2 Perancangan Sistem	40
3.2.1 Use Case Diagram.....	40
3.2.2 Class Diagram	40

3.2.3 Activity Diagram.....	41
3.2.4 Sequence Diagram	46
3.3 Perancangan <i>Database</i>	51
3.4 Desain <i>Database</i>	51
3.5 Perancangan Antarmuka.....	52
3.5.1 HIPO	52
3.5.2 UI (User Interface)	53
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	60
4.1 Implementasi Sistem	61
4.2 Implementasi Basis Data.....	61
4.3 Implementasi Program	63
4.3.1 Halaman <i>Splashscreen</i>	63
4.3.2 Halaman <i>Login</i>	63
4.3.3 Halaman Registrasi	64
4.3.4 Halaman Utama.....	65
4.3.5 Halaman Kategori	67
4.3.6 Halaman Tambah Transaksi	70
4.3.7 Halaman Rekap Bulanan.....	73
4.3.8 Halaman Profil	77
4.4 Hasil Pengujian (Testing).....	78
4.4.1 Metode Pengujian.....	78
4.4.2 Hasil Pengujian	78
4.4.3 Hasil Pengujian Detail.....	79
4.4.4 Kesimpulan Dari Hasil Pengujian.....	85
BAB V PENUTUP	85

5.1 Kesimpulan	86
5.2 Saran.....	86
DAFTAR PUSTAKA.....	87



DAFTAR GAMBAR

	Halaman
Gambar 1. 1 Metode <i>Waterfall</i>	7
Gambar 2. 1 Logo <i>Android</i>	21
Gambar 2. 2 Logo <i>Android Studio</i>	22
Gambar 2. 3 Logo <i>Flutter</i>	22
Gambar 2. 4 Logo <i>Dart</i>	24
Gambar 2. 5 Logo <i>Figma</i>	25
Gambar 2. 6 Logo AMANDI	26
Gambar 2. 7 Logo <i>SQLite</i>	29
Gambar 3. 1 <i>Flowchart</i> Sistem yang Berjalan	38
Gambar 3. 2 <i>Flowmap</i> Sistem yang diusulkan	39
Gambar 3. 3 <i>Use Case Diagram</i>	40
Gambar 3. 4 <i>Class Diagram</i>	41
Gambar 3. 5 <i>Activity Diagram Login</i>	41
Gambar 3. 6 <i>Activity Diagram Register</i>	42
Gambar 3. 7 <i>Activity Diagram Create</i>	42
Gambar 3. 8 <i>Activity Diagram Read</i>	43
Gambar 3. 9 <i>Activity Diagram Update</i>	43
Gambar 3. 10 <i>Activity Diagram Delete</i>	44
Gambar 3. 11 <i>Activity Diagram</i> Tambah Kategori	44
Gambar 3. 12 <i>Activity Diagram</i> Tambah Kategori	45
Gambar 3. 13 <i>Activity Diagram</i> Hapus Kategori	45
Gambar 3. 14 <i>Activity Diagram</i> Rekap Bulanan.....	46
Gambar 3. 15 <i>Activity Diagram Logout</i>	46
Gambar 3. 16 <i>Sequence Diagram Login</i>	47
Gambar 3. 17 <i>Sequence Diagram Register</i>	47
Gambar 3. 18 <i>Sequence Diagram Create</i>	48
Gambar 3. 19 <i>Sequence Diagram Read</i>	48
Gambar 3. 20 <i>Sequence Diagram Update</i>	48

Gambar 3. 21 <i>Sequence Diagram Delete</i>	49
Gambar 3. 22 <i>Sequence Diagram Tambah Kategori</i>	49
Gambar 3. 23 <i>Sequence Diagram Ubah Kategori</i>	49
Gambar 3. 24 <i>Sequence Diagram Hapus Kategori</i>	50
Gambar 3. 25 <i>Sequence Diagram Rekap Bulanan</i>	50
Gambar 3. 26 <i>Sequence Diagram Logout</i>	50
Gambar 3. 27 HIPO.....	52
Gambar 3. 28 UI (<i>User Interface</i>) <i>Splashscreen</i>	53
Gambar 3. 29 UI (<i>User Interface</i>) <i>Login</i>	53
Gambar 3. 30 UI (<i>User Interface</i>) <i>Register</i>	54
Gambar 3. 31 UI (<i>User Interface</i>) <i>Halaman Utama</i>	54
Gambar 3. 32 UI (<i>User Interface</i>) <i>Tambah Pemasukan</i>	55
Gambar 3. 33 UI (<i>User Interface</i>) <i>Tambah Pengeluaran</i>	55
Gambar 3. 34 UI (<i>User Interface</i>) <i>Dropdown Kategori</i>	56
Gambar 3. 35 UI (<i>User Interface</i>) <i>Tanggal</i>	56
Gambar 3. 36 UI (<i>User Interface</i>) <i>Login</i>	57
Gambar 3. 37 UI (<i>User Interface</i>) <i>Kategori Pemasukan</i>	57
Gambar 3. 38 UI (<i>User Interface</i>) <i>Kategori Pengeluaran</i>	58
Gambar 3. 39 UI (<i>User Interface</i>) <i>Tambah Kategori</i>	58
Gambar 3. 40 UI (<i>User Interface</i>) <i>Ubah Kategori</i>	59
Gambar 3. 41 UI (<i>User Interface</i>) <i>Rekap Bulanan</i>	59
Gambar 3. 42 UI (<i>User Interface</i>) <i>Review Rekap Bulanan</i>	60
Gambar 3. 43 UI (<i>User Interface</i>) <i>Halaman Profil</i>	60
Gambar 4. 1 <i>User</i>	61
Gambar 4. 2 <i>Kategori</i>	62
Gambar 4. 3 <i>Transaksi</i>	62
Gambar 4. 4 <i>Halaman Splashscreen</i>	63
Gambar 4. 5 <i>Halaman Login</i>	64
Gambar 4. 6 <i>Halaman Registrasi</i>	65
Gambar 4. 7 <i>Halaman Utama (Login Berhasil)</i>	66
Gambar 4. 8 <i>Halaman Utama (Transaksi kosong)</i>	66

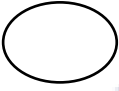
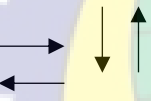
Gambar 4. 9 Halaman Utama (Transaksi ada)	67
Gambar 4. 10 Halaman Utama (<i>Calendar appbar</i>)	67
Gambar 4. 11 Halaman Kategori (Pemasukan).....	68
Gambar 4. 12 Halaman Kategori (Pengeluaran).....	68
Gambar 4. 13 Halaman Kategori (Tambah Kategori)	69
Gambar 4. 14 Halaman Kategori (Ubah Kategori)	69
Gambar 4. 15 Halaman Kategori (Hapus Kategori).....	70
Gambar 4. 16 Halaman Tambah Transaksi (Pemasukan)	71
Gambar 4. 17 Halaman Tambah Transaksi (Pengeluaran).....	71
Gambar 4. 18 Halaman Tambah Transaksi (<i>Dropdown</i> pilih kategori)	72
Gambar 4. 19 Halaman Tambah Transaksi (pilih tanggal).....	72
Gambar 4. 20 Halaman Tambah Transaksi (Keterangan)	73
Gambar 4. 21 Halaman Rekap Bulanan.....	73
Gambar 4. 22 Halaman Rekap Bulanan (<i>Dropdown</i> bulan)	74
Gambar 4. 23 Halaman Rekap Bulanan (<i>Review PDF</i>)	74
Gambar 4. 24 Halaman Rekap Bulanan (Simpan PDF ke Perangkat).....	75
Gambar 4. 25 Halaman Rekap Tahunan.....	75
Gambar 4. 26 Halaman Rekap Tahunan (<i>Dropdown</i> tahun).....	76
Gambar 4. 27 Halaman Rekap Tahunan (<i>Review PDF</i>).....	76
Gambar 4. 28 Halaman Profil	77
Gambar 4. 29 Halaman Profil (<i>Upload Foto</i>)	77
Gambar 4. 30 Halaman Profil (<i>Logout</i>)	78

DAFTAR TABEL

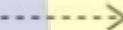
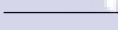
	Halaman
Tabel 1. 1 Jadwal Kegiatan Penelitian	11
Tabel 3. 1 Perangkat Keras Pembuatan.....	33
Tabel 3. 2 Perangkat Keras Implementasi.....	33
Tabel 3. 3 Tabel <i>User</i>	51
Tabel 3. 4 Tabel Kategori	52
Tabel 3. 5 Tabel Transaksi.....	52
Tabel 4. 1 <i>User</i>	61
Tabel 4. 2 Kategori.....	62
Tabel 4. 3 Transaksi.....	62
Tabel 4. 4 Pengujian fitur Utama	78
Tabel 4. 5 Skenario Pengujian pada Halaman Login	79
Tabel 4. 6 Skenario Pengujian pada Halaman Registrasi.....	80
Tabel 4. 7 Skenario Pengujian pada Halaman Utama	82
Tabel 4. 8 Skenario Pengujian pada Halaman Kategori.....	83
Tabel 4. 9 Skenario Pengujian pada Halaman Tambah Transaksi.....	83
Tabel 4. 10 Skenario Pengujian pada Halaman Rekap	84
Tabel 4. 11 Skenario Pengujian pada Halaman Profil.....	85

DAFTAR SIMBOL

A. Flowchart dan Flowmap

No	Simbol	Nama	Keterangan
1.		Terminal	Berfungsi untuk mengidentifikasi titik awal atau akhir suatu proses.
2.		Proses Manual	Menunjukkan proses yang dilakukan secara manual.
3.		Penyimpanan Magnetik	Menunjukkan penyimpanan data atau informasi <i>file</i> pada proses berbasis komputer. <i>File</i> dapat disimpan pada <i>harddisk</i> , disket, CD.
4.		Arah Alir Dokumen	Menunjukkan arah aliran dokumen.
5.		Input/Output	Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
6.		Proses Komputer	Menunjukkan proses yang terkomputerisasi.
7.		Decision	Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak
8.		Input <i>Keyboard</i>	Menunjukkan input yang dimasukkan oleh <i>keyboard</i> .

B. UML (Unified Modelling Language)

No	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen tidak mandiri (<i>independent</i>).
3.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		Sistem	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan oleh sistem yang menghasilkan hasil yang terukur bagi suatu <i>actor</i> .

No	Simbol	Nama	Keterangan
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.



DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Halaman Biografi	88
Lampiran 2 Nota Dinas	89
Lampiran 3 Form Bimbingan.....	90
Lampiran 4 Form Revisi Penguji	92
Lampiran 5 Daftar Hadir.....	94
Lampiran 6 Berita Acara	95
Lampiran 7 Listing Program.....	96



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Manajemen keuangan pribadi merupakan aspek penting dalam kehidupan sehari-hari yang membantu untuk mengelola pemasukan dan pengeluaran secara lebih bijaksana. Dengan perkembangan teknologi yang pesat, kebutuhan akan solusi manajemen keuangan yang praktis dan efisien semakin meningkat. Salah satu permasalahan yang sering dihadapi oleh masyarakat adalah kurangnya pencatatan yang rapi terhadap transaksi keuangan pribadi, yang pada akhirnya dapat menyebabkan ketidakmampuan untuk memantau arus keuangan secara akurat.

Saat ini, ada banyak aplikasi manajemen keuangan yang tersedia di pasaran (*Play Store*), seperti *Mint*, *Money Lover*, *Monefy*, dan *YNAB (You Need A Budget)*, yang menawarkan berbagai fitur canggih seperti sinkronisasi *cloud*, analisis grafis, dan integrasi dengan bank. Meskipun fitur-fitur tersebut memberikan kemudahan, sebagian pengguna merasa bahwa kompleksitas dan banyaknya fitur yang ditawarkan sering kali tidak diperlukan untuk keperluan pencatatan sederhana. Pengguna yang hanya ingin melakukan pencatatan transaksi keuangan dasar sering kali merasa kebingungan atau kesulitan menggunakan aplikasi yang terlalu kompleks.

Aplikasi yang akan dirancang dalam penelitian ini menawarkan pendekatan yang berbeda. Fokus dari aplikasi ini adalah kesederhanaan dan kemudahan penggunaan. Aplikasi ini hanya menyediakan komponen inti yang dibutuhkan pengguna untuk mencatat pemasukan dan pengeluaran, dengan antarmuka yang minimalis dan tidak rumit. Pengguna dapat dengan mudah menambahkan, memperbarui, dan menghapus transaksi tanpa harus melalui langkah-langkah yang kompleks atau berurusan dengan fitur tambahan yang mungkin tidak dibutuhkan.

Berdasarkan uraian di atas, penulis berinisiatif untuk mengembangkan "**Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* Menggunakan *Framework Flutter***". Aplikasi ini diharapkan dapat membantu pengguna dalam mencatat dan memantau pemasukan serta pengeluaran secara efisien, memberikan visualisasi data yang mudah dipahami, serta menyajikan informasi keuangan pribadi secara terstruktur.

1.2 Identifikasi Masalah

Berdasarkan latar belakang tersebut, penulis menemukan beberapa permasalahan Manajemen Keuangan yaitu :

- a. Kesulitan dalam Mencatat dan Memantau Transaksi Keuangan Pribadi : Banyak individu mengalami kesulitan dalam mencatat transaksi keuangan mereka secara manual, baik pemasukan maupun pengeluaran. Hal ini sering menyebabkan pencatatan yang tidak tersistem, sehingga kondisi keuangan sulit dipantau secara menyeluruh.
- b. Kurangnya Aplikasi yang Memadai : Aplikasi manajemen keuangan yang tersedia saat ini banyak menggunakan fitur-fitur yang berlebihan. Hal ini mengakibatkan kebingungan bagi pengguna awam.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah tersebut, penulis menemukan beberapa rumusan masalah Manajemen Keuangan yaitu :

- a. Bagaimana merancang dan mengembangkan aplikasi manajemen keuangan pribadi berbasis *Mobile* yang dapat membantu pengguna dalam mencatat dan memantau transaksi keuangan secara mudah dan efisien ?
- b. Bagaimana merancang visualisasi data keuangan yang informatif dan mudah dipahami untuk membantu pengguna dalam mengevaluasi kondisi finansial mereka secara *real-time* ?

1.4 Batasan Masalah

Agar penelitian ini dapat berjalan dengan fokus dan terarah, serta untuk membatasi ruang lingkup pengembangan aplikasi, maka penulis menetapkan batasan masalah sebagai berikut :

1. Platform Pengembangan : Aplikasi manajemen keuangan pribadi ini dibuat dengan *Android Studio* menggunakan *Framework Flutter* dengan bahasa *Dart*. *Database* yang digunakan adalah *SQLite*. Aplikasi ini menggunakan penyimpanan lokal melalui *SQLite*, sehingga pengguna dapat tetap mencatat transaksi tanpa perlu terhubung ke internet atau akun *Online*. Pengembangan dan pengujian aplikasi hanya dilakukan pada perangkat *Mobile (smartphone) Android* dan tidak mencakup platform *IOS*, desktop atau *web*.
2. Pengguna Aplikasi : Aplikasi ini dirancang untuk digunakan oleh individu yang ingin mengelola keuangan pribadinya secara mandiri. Aplikasi tidak mencakup fitur untuk pengelolaan keuangan kelompok atau organisasi, seperti pengelolaan keuangan keluarga atau bisnis kecil. Aplikasi ini hanya memiliki komponen inti *CRUD (Create, Read, Update, Delete)*, kategori transaksi, rekap bulanan dan sisa saldo pengguna. Fokus utama aplikasi ini adalah kesederhanaan, di mana pengguna dapat mencatat transaksi keuangan secara langsung tanpa harus melalui langkah-langkah atau fitur yang rumit.
3. Platform Aplikasi : Aplikasi ini dirancang untuk perangkat *Mobile* dan hanya tersedia pada *platform Android*. Pengguna yang ingin menggunakan aplikasi ini harus memiliki perangkat berbasis *Android* (minimal versi 12 *Snow Cone*). Aplikasi tidak tersedia untuk platform *iOS* dan desktop (*Windows, macOS, Linux*) atau *web*.

1.5 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

1. Untuk menghasilkan sebuah aplikasi manajemen keuangan pribadi berbasis *Mobile* yang dapat membantu pengguna mencatat dan memantau pemasukan serta pengeluaran keuangan secara terstruktur dengan komponen inti CRUD (*Create, Read, Update, Delete*), kategori transaksi, rekap bulanan dan sisa saldo pengguna .
2. Untuk menghasilkan antarmuka pengguna yang sederhana, mudah digunakan, dan responsif.

1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan beberapa manfaat baik secara praktis maupun teoritis, yaitu sebagai berikut :

1. Manfaat Bagi Pengguna :
 - a. Aplikasi manajemen keuangan pribadi dapat membantu pengguna dalam mencatat dan memantau transaksi keuangan secara lebih efisien, sehingga dapat mengelola pemasukan dan pengeluaran dengan lebih baik.
 - b. Aplikasi ini juga memberikan kemudahan bagi pengguna dalam memahami kondisi keuangan mereka melalui visualisasi data yang sederhana, membantu dalam membuat keputusan keuangan yang lebih tepat dan terarah.
2. Manfaat Bagi Penulis :
 - a. Aplikasi ini dapat memperdalam pengetahuan peneliti mengenai pengembangan sistem menggunakan *Framework Flutter*.
 - b. Aplikasi ini juga memberikan pengalaman bagi peneliti dalam merancang dan mengembangkan aplikasi.

3. Manfaat Bagi Dunia Pendidikan :

- a. Aplikasi ini dapat menjadi referensi bagi mahasiswa atau peneliti lain yang tertarik pada topik pengembangan aplikasi berbasis *Mobile*, terutama dalam penerapan *Framework Flutter*.
- b. Aplikasi ini juga dapat digunakan sebagai studi kasus atau bahan ajar pengembangan aplikasi *Mobile*.

4. Manfaat Bagi Pengembang Aplikasi :

- a. Aplikasi ini dapat menjadi contoh penerapan *Framework Flutter* dalam pengembangan aplikasi yang efisien dan efektif.
- b. Aplikasi ini juga dapat menjadi model acuan dalam merancang aplikasi manajemen keuangan pribadi yang mengutamakan kemudahan penggunaan.

5. Manfaat Bagi Masyarakat Umum :

- a. Aplikasi yang dikembangkan dalam penelitian ini diharapkan dapat meningkatkan kesadaran masyarakat akan pentingnya manajemen keuangan pribadi dan membantu individu untuk lebih disiplin dalam mengelola keuangan. Khususnya dalam pencatatan pemasukan dan pengeluaran.
- b. Dengan adanya aplikasi ini, masyarakat juga dapat lebih mudah memantau kondisi keuangan mereka.

1.7 Metode Penelitian

Metode penelitian yang digunakan dalam pengembangan “**Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* Menggunakan *Framework Flutter***” ini meliputi beberapa tahapan, yaitu sebagai berikut :

- 1. Metode Pengembangan Aplikasi : Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah metode *Waterfall*, yang terdiri dari tahapan-tahapan berikut :

a. Analisis Kebutuhan :

Pada tahap ini, peneliti melakukan analisis terhadap kebutuhan pengguna dan merumuskan spesifikasi fungsional serta non-fungsional yang harus dipenuhi oleh aplikasi.

b. Perancangan Sistem (*System Design*) :

Tahap ini mencakup perancangan arsitektur sistem, antarmuka pengguna UI (*User Interface*), alur kerja (*workflow*), serta pemodelan basis data yang akan digunakan dalam aplikasi. Perancangan dilakukan dengan menggunakan diagram-diagram seperti *Usecase Diagram*, *Class Diagram*, *Activity Diagram*, *Sequence Diagram* dan *wireframe* untuk visualisasi antarmuka.

c. Pengembangan (*Coding & Implementation*):

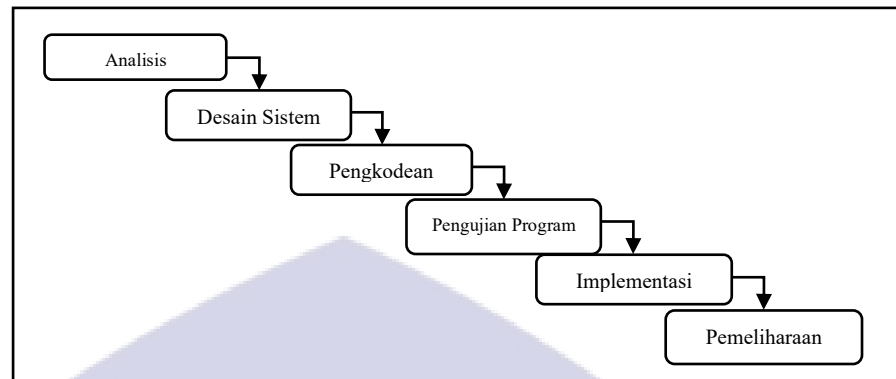
Implementasi kode program dilakukan dengan menggunakan *Framework Flutter* untuk pengembangan aplikasi. Pada tahap ini, komponen-komponen aplikasi seperti halaman *input* transaksi dan visualisasi data keuangan.

d. Pengujian (*Testing*):

Pengujian aplikasi dilakukan dengan metode *Black Box Testing* untuk memeriksa fungsionalitas aplikasi secara keseluruhan. Selain itu, pengujian dilakukan pada *platform Android* untuk memastikan konsistensi fitur dan stabilitas performa. Hasil pengujian akan dievaluasi untuk memastikan bahwa aplikasi telah berjalan sesuai dengan spesifikasi yang ditetapkan.

e. Pemeliharaan (*Maintenance*) :

Setelah pengujian, aplikasi yang telah selesai dikembangkan akan dipelihara dan diperbaiki jika ditemukan kesalahan (*bug*) atau jika ada penambahan fitur di masa mendatang. Pemeliharaan ini juga mencakup optimasi aplikasi agar lebih responsif dan ringan.



Gambar 1. 1 Metode *Waterfall*

2. Metode Pengumpulan Data :

a. Studi Literatur :

Peneliti melakukan kajian literatur dari berbagai sumber, seperti buku, jurnal, artikel ilmiah, dan *internet* yang terkait dengan topik manajemen keuangan, pengembangan aplikasi berbasis *Mobile*, serta penerapan *Framework Flutter*. Tujuan dari studi literatur ini adalah untuk memahami konsep dasar dan mendapatkan wawasan tentang solusi yang telah ada serta tren terkini di bidang yang relevan.

b. Observasi :

Peneliti melakukan observasi terhadap aplikasi manajemen keuangan pribadi yang ada di pasaran untuk mempelajari fitur-fitur yang ditawarkan, kelebihan, kekurangan, serta antarmuka pengguna yang diterapkan. Hasil observasi ini digunakan sebagai acuan dalam merancang fitur dan desain aplikasi yang dikembangkan.

3. Metode Analisis Data : Analisis data dilakukan dengan mengevaluasi hasil pengujian aplikasi untuk menilai apakah aplikasi telah memenuhi kebutuhan pengguna. Data dari pengujian akan diolah dan disajikan dalam bentuk laporan yang mencakup temuan, kendala yang dihadapi, serta solusi yang diterapkan.

4. Metode Perancangan

a. Analisis Kebutuhan Pada tahap ini, dilakukan analisis kebutuhan sistem dengan berfokus pada kebutuhan fungsional dan non-fungsional. Tujuannya adalah untuk memahami fitur apa saja yang

diperlukan oleh pengguna dan bagaimana aplikasi harus berfungsi untuk memenuhi kebutuhan pencatatan keuangan pribadi. Kebutuhan sistem meliputi operasi CRUD (*Create, Read, Update, Delete*), kategori transaksi, rekap bulanan dan sisa saldo pengguna serta tampilan antarmuka yang sederhana.

- b. Perancangan Sistem Setelah analisis kebutuhan dilakukan, tahap selanjutnya adalah merancang sistem. Proses perancangan meliputi beberapa aspek penting:
 1. Perancangan Arsitektur Sistem: Sistem dirancang menggunakan arsitektur *client-side*, di mana seluruh pengolahan data dilakukan pada perangkat pengguna (*Mobile*). Aplikasi ini dikembangkan menggunakan *Framework Flutter* untuk mendukung platform *Android*, dengan basis data *SQLite* untuk penyimpanan lokal.
 2. Perancangan Basis Data: Pada tahap ini, dibuat perancangan basis data yang akan digunakan dalam aplikasi. Basis data menggunakan *SQLite* dengan satu tabel utama, yaitu tabel transaksi, yang berisi informasi transaksi keuangan (tanggal, kategori, jumlah dan keterangan).
 3. Perancangan Antarmuka Pengguna (*User Interface*): Merancang antarmuka pengguna yang mudah digunakan dengan navigasi yang jelas. Antarmuka dirancang menggunakan pendekatan *Material Design* untuk tampilan yang konsisten di *Android*, dengan fokus pada kemudahan akses ke fitur-fitur utama seperti pencatatan transaksi dan melihat daftar transaksi.
- c. Implementasi sistem setelah perancangan selesai, dilakukan tahap implementasi. Pada tahap ini, kode program aplikasi dikembangkan menggunakan *Flutter* sebagai *Framework* pengembangan aplikasi *Mobile*. Bahasa pemrograman yang digunakan adalah *Dart*, dan basis data *SQLite* diintegrasikan untuk penyimpanan data transaksi.
- d. Pengujian sistem setelah implementasi selesai, dilakukan pengujian untuk memastikan bahwa semua fitur berfungsi dengan baik sesuai

dengan kebutuhan yang telah dianalisis sebelumnya. Pengujian yang dilakukan meliputi:

1. Pengujian Fungsional: Menguji operasi CRUD untuk memastikan pengguna dapat menambah, melihat, memperbarui, dan menghapus transaksi dengan benar.
2. Pengujian Kinerja: Memastikan aplikasi berjalan dengan performa yang baik di berbagai perangkat *Android*.
3. Pengujian Antarmuka: Menguji kenyamanan dan kemudahan penggunaan antarmuka aplikasi oleh pengguna.
- e. Penyempurnaan dan Pemeliharaan Setelah pengujian, dilakukan perbaikan jika ditemukan *bug* atau masalah yang mengganggu kinerja aplikasi. Penyempurnaan terus dilakukan berdasarkan umpan balik dari pengguna, dan aplikasi direncanakan untuk dipelihara secara berkala.

5. Metode Pengembangan

a. Pembuatan Basis Data (*Database*)

Pada tahap ini, basis data dirancang untuk mendukung kebutuhan aplikasi manajemen keuangan. Basis data dibuat menggunakan *SQLite* yang bersifat lokal dan mudah diintegrasikan dengan aplikasi *Mobile*.

b. Pembuatan UI (*User Interface*)

Antarmuka pengguna dirancang menggunakan *Flutter* dengan pendekatan Material Design. Fokus utamanya adalah memastikan antarmuka mudah digunakan,

c. Koneksi Antar *Database* dan UI (*User Interface*)

Setelah basis data (*Database*) dan UI (*User Interface*) selesai dirancang, koneksi antara aplikasi dan basis data diimplementasikan menggunakan *SQLite*. Fungsi CRUD (*Create, Read, Update, Delete*) dikembangkan agar aplikasi dapat berinteraksi langsung dengan basis data, memungkinkan pengguna untuk menambah, melihat, mengubah, dan menghapus data transaksi keuangan.

d. Pembuatan *Activity Diagram*

Activity Diagram yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas.

e. Pembuatan *Sequence Diagram*

Sequence Diagram adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan.

f. Pembuatan HIPO (*Hierarchy plus Input Process Output*)

HIPO dibuat untuk memodelkan proses sistem secara hierarkis, dimulai dari level atas hingga detail *input*-proses-output. Diagram HIPO menggambarkan struktur logis dari aliran data di dalam aplikasi. Setiap fungsi dalam aplikasi, seperti pencatatan transaksi dan tampilan riwayat transaksi, diuraikan dalam diagram ini untuk memetakan alur data yang terjadi di antara komponen.

6. Metode Pengujian : Pengujian dilakukan dengan dua pendekatan :

a. Fungsionalitas (*Functional Testing*) :

Pengujian dilakukan untuk memastikan bahwa setiap fitur yang dikembangkan berfungsi sesuai dengan baik.

b. Pengalaman Pengguna (*User Experience Testing*) :

Pengujian ini dilakukan untuk mengukur seberapa mudah aplikasi digunakan oleh pengguna, kenyamanan antarmuka, serta kepuasan pengguna dalam menggunakan aplikasi. Pengujian dapat dilakukan dengan meminta beberapa responden untuk mencoba aplikasi dan memberikan *feedback* terkait desain dan fungsionalitas.

1.8 Jadwal Kegiatan Penelitian

Berikut adalah jadwal penelitian yang direncanakan dalam pengembangan "**Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile Menggunakan Framework Flutter***". Jadwal ini dirinci ke dalam beberapa tahapan kegiatan berdasarkan metode penelitian yang telah dijelaskan :

Tabel 1. 1 Jadwal Kegiatan Penelitian

No	Kegiatan	Waktu (Bulan) 2024 - 2025										
		Sep	Okt	Nov	Des	Jan	Feb	Mar	Apr	Mei	Jun	Jul
1	Permohonan Judul											
2	Pengumpulan Data											
3	Penyusunan Proposal											
4	Bimbingan Proposal											
5	Seminar Proposal											
6	Revisi Proposal											
7	Skripsi											
8	Perbaikan											
9	Jadwal Meja Hijau											

1.9 Sistematika Penulisan

Dalam menyusun karya tulis ilmiah ini, agar dalam pembahasan terfokus pada pokok permasalahan dan tidak melebar ke masalah yang lain, maka penulis membuat sistematika penulisan karya tulis ilmiah sebagai berikut :

BAB I PENDAHULUAN

Dalam bab ini penulis membahas tentang Latar Belakang Masalah, Rumusan Masalah, Identifikasi Masalah, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Metode Penelitian, Jadwal Kegiatan Penelitian dan Sistematika Penulisan.

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Dalam bab ini penulis membahas tentang Tinjauan Pustaka dan Landasan Teori.

BAB III ANALISA DAN PERANCANGAN

Dalam bab ini penulis membahas tentang Analisa dan Perancangan Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* Menggunakan *Framework Flutter*.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Dalam bab ini penulis membahas tentang Metodologi Penelitian, pembahasan, Pengoperasian Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile*.

BAB V PENUTUP

Dalam bab ini berisi tentang Kesimpulan dan Saran



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Setelah peneliti melakukan telaah terhadap beberapa penelitian, ada beberapa yang memiliki keterkaitan dengan penelitian yang peneliti lakukan, yaitu sebagai berikut :

1. **Penelitian oleh Munawir Sajali Harahap, Richi Andrianto, Dermilan, Dewi Elok, Citra Khoiriah dan Babang Andika (2023) “Pencatat Keuangan: Pengembangan Aplikasi *Flutter* untuk Mencatat dan Mengelola Transaksi Keuangan”.** Penelitian ini membahas tentang pengembangan aplikasi pencatat keuangan menggunakan *Flutter*, sebuah *Framework* pengembangan yang terkenal dalam pembuatan aplikasi lintas platform. Tujuan dari aplikasi ini adalah untuk memberikan dukungan bagi pengguna dalam mencatat dan mengelola keuangan pribadi mereka dengan cara yang mudah, efisien, interaktif dan memiliki fitur pengelompokkan kategori utama (pemasukan dan pengeluaran). Sedangkan penelitian yang akan dilakukan adalah bertujuan untuk memudahkan pengguna dengan fitur minimalis, aplikasi ini akan lebih ringan dan cepat dijalankan, sehingga cocok untuk pengguna yang hanya memerlukan pencatatan dasar. Penelitian sebelumnya menggunakan MySQL berbasis server, sedangkan penelitian ini menggunakan *SQLite* sebagai penyimpanannya.
2. **Penelitian oleh Rakhel Cakra Kusumadinata Sera (2021) “Rancangan Bangun Aplikasi Money Changer Sederhana Berbasis *Android*”.** Penelitian ini menggunakan metode *Waterfall*. Aplikasi *Mobile* pada platform *Android* yang dibuat menggunakan aplikasi *Android Studio* dengan bahasa *dart* dan menggunakan *Framework Flutter*. Aplikasi ini mempunyai fitur utama untuk membagi keuangan secara spesifik. Dari

kebutuhan investasi, simpanan, keperluan sehari-hari, dan pengeluaran bebas. Sedangkan penelitian yang akan dilakukan adalah berfokus pada kesederhanaan, di mana pengguna dapat mencatat transaksi keuangan secara langsung tanpa harus melalui langkah-langkah atau fitur yang rumit.

3. **Penelitian oleh Suhatati Tjandra, Grace Levina Dewi, Surya Putrasia Santoso dan Judi Prajetno (2022) “Pengembangan Aplikasi Pengaturan Keuangan Pribadi Berbasis *Android*”.** Fitur dari aplikasi manajemen keuangan yang dibuat adalah pengaturan budget bulanan, pencatatan transaksi, tambah kategori transaksi, pencatatan hutang piutang, push notifikasi tanggal jadi pembayaran hutang, melihat tabel mutasi saldo, laporan pengeluaran dan pemasukan dalam bentuk grafik *pie*, laporan perbandingan pengeluaran dan pemasukan dalam bentuk grafik bar, laporan bulanan pengeluaran dalam bentuk *line*, detail laporan harian, mingguan, bulanan, dan tahunan, *backup* dan *restore* data, dan *reminder* pencatatan transaksi. Dengan dibuatnya aplikasi ini pengguna dapat dipermudah dalam mengatur transaksi pengeluaran dan pemasukan keuangan. Pengguna juga dapat memonitor keuangannya yang tersedia dalam laporan harian, mingguan, bulanan, dan tahunan. Sedangkan penelitian yang akan dilakukan hanya mempunyai operasi dasar CRUD (*Create, Read, Update, Delete*), kategori transaksi, rekap bulanan dan sisa saldo pengguna, aplikasi yang akan dibuat berfokus pada pengguna awam agar lebih mudah digunakan dan tidak membingungkan.
4. **Penelitian oleh Erga Trivaika dan Mamok Andri Senubekti (2022) “Perancangan Aplikasi Pengelola Keuangan Pribadi Berbasis *Android*”.** Penelitian ini menjelaskan perilaku konsumtif yang sering terjadi di tengah masyarakat dapat menjadi masalah utama dalam pengelolaan keuangan. Adapun yang menjadi fokus dalam penelitian ini adalah bagaimana penulis dapat membuat suatu aplikasi untuk memberikan informasi dan referensi kepada pengguna tentang cara pengelolaan keuangan yang baik dan benar, dengan menampilkan

informasi keuangan di smartphone dengan Platform *Android*. Sedangkan penelitian yang akan dilakukan adalah hanya berfungsi sebagai alat pencatat keuangan sederhana tanpa fitur edukasi, analisis perilaku keuangan, atau referensi untuk pengelolaan keuangan.

5. **Penelitian oleh Ujang Juhardi dan Khairullah (2020) “Sistem Pencatatan dan Pengolahan Keuangan Pada Aplikasi Manajemen Keuangan E-Dompet Berbasis *Android*”.** Tujuan dari penelitian ini adalah untuk mempermudah dan membantu seseorang dalam mencatat, menghitung dan mengelola pemasukan serta pengeluaran keuangannya menggunakan sebuah perangkat bergerak yang berbasis *Android*. Sedangkan penelitian yang akan dilakukan adalah berfokus pada pencatatan transaksi keuangan yang sederhana tanpa fitur-fitur tambahan seperti pengingat, laporan otomatis, atau integrasi bank.

6. **Penelitian oleh Alwin Fau (2024) “Pelatihan Pengenalan Dasar *Framework Flutter* dalam Pembangunan Aplikasi *Mobile*”.** Penelitian ini menjelaskan bahwa Pemanfaatan media *Mobile* dalam transaksi bisnis saat ini sudah sangat berkembang pesat. Hal inilah yang menjadi salah satu pendorong perkembangan di bidang *Framework* dan bahasa pemrograman berkembang khususnya bahasa pemrograman yang dapat digunakan untuk membangun dan mengembangkan aplikasi *Mobile*. Keahlian dalam pengembangan dan pembuatan aplikasi *Mobile* merupakan salah satu keahlian yang memberikan peluang besar bagi siapa saat ini terutama kepada para peserta didik yang sedang menggeluti bidang rekayasa perangkat lunak. Hal ini didasari oleh perkembangan teknologi yang semakin memudahkan seluruh proses transaksi secara *Online* terutama dengan memanfaatkan media *Mobile*. *Flutter* merupakan salah satu *Framework* yang telah dikembangkan menggunakan bahasa dart dan yang diperuntukkan untuk mengembangkan dan membangun aplikasi *Mobile*. Kemudahan dan fleksibilitas dari *Framework* ini membuatnya telah banyak digunakan mengembangkan dan membangun aplikasi *Mobile*. Pelatihan ini bertujuan untuk memberikan pengenalan dan pelatihan dasar

bagi peserta didik jurusan rekayasa perangkat lunak tentang pemanfaatan *Framework Flutter* dalam membangun dan mengembangkan aplikasi *Mobile*, dengan harapan para peserta didik memiliki referensi dalam pengembangan keahliannya khususnya dalam pembangunan aplikasi *Mobile*. Sedangkan penelitian yang akan dilakukan adalah untuk mempermudah pencatatan keuangan, tanpa elemen edukatif atau pelatihan. Aplikasi ini berfokus pada kemudahan penggunaan dan penyediaan fitur dasar bagi pengguna umum, bukan pada pengembangan keterampilan teknis.

2.2 Landasan Teori

Landasan teori ini mencakup konsep-konsep dasar serta penjelasan tentang teknologi dan metode yang digunakan dalam pengembangan aplikasi. Berikut adalah beberapa landasan teori yang terkait :

2.2.1 Manajemen Keuangan

Manajemen keuangan adalah proses perencanaan, pengorganisasian, pengendalian, dan pemantauan sumber daya keuangan untuk mencapai tujuan keuangan individu, keluarga, atau organisasi (Dian Sudiantini, 2023). Dalam praktiknya, manajemen keuangan mencakup aktivitas pengelolaan pemasukan dan pengeluaran, investasi, serta pengelolaan risiko untuk menjaga stabilitas dan keberlanjutan keuangan.

2.2.1.1 Komponen Manajemen Keuangan

Manajemen keuangan terdiri dari beberapa komponen utama yang saling berkaitan untuk memastikan pengelolaan keuangan yang optimal. Berikut adalah penjelasan dari masing-masing komponen :

1. Perencanaan keuangan

Perencanaan keuangan adalah langkah awal dalam manajemen keuangan yang bertujuan untuk menetapkan tujuan keuangan jangka pendek, menengah, dan panjang.

a. Aktivitas Utama:

1. Membuat anggaran (budgeting) untuk mengatur pemasukan dan pengeluaran.
2. Menentukan alokasi dana untuk kebutuhan rutin, investasi, dan tabungan.
3. Mengidentifikasi prioritas keuangan dan strategi pencapaiannya.

b. Manfaat :

1. Membantu individu atau organisasi memiliki pandangan yang jelas tentang tujuan keuangan.
2. Menghindari pengeluaran yang tidak perlu dan memastikan ketersediaan dana untuk kebutuhan penting.

2. Pengelolaan likuiditas

Pengelolaan likuiditas adalah proses memastikan bahwa terdapat dana tunai yang cukup untuk memenuhi kebutuhan mendesak atau operasional sehari-hari.

a. Aktivitas Utama:

1. Memantau saldo kas atau rekening bank secara rutin.
2. Mengelola arus kas masuk dan keluar agar tetap seimbang.
3. Menyediakan dana darurat untuk kebutuhan mendesak.

b. Manfaat:

1. Memastikan kemampuan untuk membayar kewajiban dalam jangka pendek, seperti tagihan atau biaya operasional.
2. Menghindari risiko kebangkrutan akibat kekurangan dana tunai.

3. Pengendalian pengeluaran

Pengendalian pengeluaran adalah proses pengawasan dan pengendalian penggunaan dana agar sesuai dengan anggaran atau rencana yang telah ditetapkan.

a. Aktivitas Utama:

1. Memantau pengeluaran rutin untuk memastikan kesesuaiannya dengan anggaran.
2. Mengidentifikasi dan mengurangi pemborosan atau pengeluaran tidak efisien.
3. Membandingkan realisasi pengeluaran dengan target yang telah ditentukan.

b. Manfaat:

1. Mencegah pengeluaran berlebihan yang dapat menyebabkan defisit keuangan.
2. Meningkatkan efisiensi penggunaan dana untuk mencapai tujuan keuangan.

2.2.1.2 Prinsip Dasar Manajemen Keuangan

Prinsip dasar manajemen keuangan adalah pedoman utama yang membantu individu atau organisasi mengelola keuangan secara efektif (Erga Trivaika dan Mamok Andri Sanubekti, 2022).

Berikut adalah penjelasan dari masing-masing prinsip :

1. Akurasi dalam pencatatan

Akurasi dalam pencatatan mengacu pada pentingnya mencatat semua transaksi keuangan dengan benar dan rinci untuk mencerminkan kondisi keuangan yang sebenarnya.

a. Penerapan :

1. Mencatat semua pemasukan, pengeluaran, dan transaksi lainnya secara terperinci.
2. Menggunakan alat bantu seperti buku kas, spreadsheet, atau perangkat lunak akuntansi.

3. Memastikan setiap catatan transaksi memiliki bukti pendukung, seperti struk atau faktur.

b. Manfaat :

1. Membantu mengidentifikasi sumber kebocoran keuangan atau ketidaksesuaian anggaran.
2. Mempermudah proses pelaporan keuangan, analisis, dan pengambilan keputusan.
3. Menjamin transparansi dan akuntabilitas keuangan.

2. Efisiensi penggunaan sumber daya

Efisiensi dalam penggunaan sumber daya berarti memastikan bahwa setiap dana yang tersedia digunakan secara optimal untuk mencapai tujuan yang telah ditetapkan.

a. Penerapan :

1. Menyusun anggaran yang realistis dan berbasis prioritas.
2. Menghindari pengeluaran untuk kebutuhan yang kurang penting atau tidak mendesak.
3. Memaksimalkan hasil atau manfaat dari dana yang dikeluarkan, seperti memilih vendor atau produk dengan kualitas terbaik dalam anggaran.

b. Manfaat :

1. Menghindari pemborosan sumber daya keuangan.
2. Memastikan bahwa setiap dana yang dikeluarkan memberikan nilai atau manfaat maksimal.
3. Membantu mencapai tujuan keuangan tanpa menimbulkan beban finansial tambahan.

3. Pengelolaan risiko keuangan

Pengelolaan risiko keuangan adalah proses mengidentifikasi, menilai, dan mengambil langkah untuk mengurangi atau mengelola risiko yang dapat mempengaruhi stabilitas keuangan.

a. Penerapan:

1. Menyediakan dana darurat untuk menghadapi situasi tak terduga, seperti kehilangan pekerjaan atau kebutuhan mendadak.
2. Diversifikasi investasi untuk mengurangi risiko kerugian besar dari satu instrumen keuangan.
3. Menggunakan asuransi untuk melindungi dari risiko seperti kecelakaan, kesehatan, atau kerugian bisnis.

b. Manfaat:

1. Mengurangi dampak negatif dari kejadian tak terduga terhadap kondisi keuangan.
2. Memastikan keberlanjutan dan stabilitas keuangan dalam jangka panjang.
3. Membantu menciptakan rasa aman dalam pengelolaan keuangan.

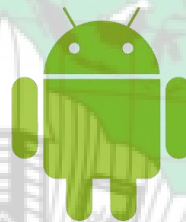
2.2.2 Aplikasi

Aplikasi adalah suatu perangkat lunak atau program komputer untuk melakukan perintah tertentu. Secara harfiah, aplikasi adalah suatu penerapan perangkat lunak atau software yang dikembangkan untuk tujuan melakukan tugas-tugas tertentu. Aplikasi juga dapat diartikan sebagai sebuah perangkat lunak atau program yang diciptakan dan dikembangkan untuk melakukan tugas-tugas tertentu pada perangkat komputer, laptop, atau ponsel. Aplikasi adalah suatu perangkat lunak yang dibuat khusus untuk memenuhi kebutuhan berbagai aktivitas dan pekerjaan, seperti pelayanan masyarakat, periklanan, game, dan sebagainya. (Hengky W. Pramana 2022).

2.2.3 Android

Android adalah sebuah sistem operasi platform open source yang dirancang untuk perangkat *Mobile*. *Android* dikuasai dan dikembangkan

oleh *Google* dan dimiliki oleh *Open Handset Alliance*. *Android* menyediakan kit *development* perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna *Android*. *Android* juga menyediakan pasar untuk mendistribusikan aplikasi. Secara keseluruhan, *Android* menyatakan ekosistem untuk aplikasi seluler. Sebagai platform perangkat seluler paling populer di dunia, *Android* mendukung ratusan juta perangkat seluler di lebih dari 190 negara di seluruh dunia. *Android* memiliki basis pemasangan terbesar dari platform seluler apapun dan masih tumbuh dengan cepat. Setiap hari, jutaan pengguna lain menyalakan perangkat *Android* mereka untuk pertama kalinya dan mulai mencari aplikasi, game, dan materi digital lainnya .



Gambar 2. 1 Logo *Android*

2.2.4 Android Studio

Android Studio adalah Integrated Development Enviroment (IDE) resmi untuk sistem operasi Android, yang dibangun di atas perangkat lunak JetBrains IntelliJ IDEA dan didesain khusus untuk pengembangan Android. IDE ini merupakan pengganti dari Eclipse Android Development Tools (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi android. Android studio sendiri pertama kali diumumkan di konferensi Google I/O pada tanggal 16 Mei 2013. Ini merupakan tahap preview dari versi 0.1 pada Mei 2013, dan memasuki tahap beta sejak versi 0.8 dan mulai diliris pada Juni 2014. Versi rilis stabil yang pertama diliris pada Desember 2014, dimulai sejak versi 1.0. Sedangkan versi stabil yang sekarang adalah versi 4.0 yang diliris pada

Mei 2020. Pada 7 Mei 2019, Kotlin menggantikan Java sebagai bahasa pilihan Google untuk pengembangan aplikasi Android. Java masih didukung, begitu pula C++. (Sumber : wikipedia.org)



Gambar 2. 2 Logo *Android Studio*

2.2.5 Framework **Flutter**

Flutter adalah *Framework* open-source yang dikembangkan oleh *Google* untuk membangun aplikasi *Mobile*, *web*, dan *desktop* dengan satu basis kode. (Sumber: *Flutter Documentation*, 2024).



Gambar 2. 3 Logo *Flutter*

2.2.5.1 Keunggulan **Flutter**

1. Pemrograman lintas platform

Flutter memungkinkan pengembang membuat aplikasi untuk berbagai *platform* (*Android*, *iOS*, *web*, dan *desktop*) dengan satu basis kode (pada penelitian ini hanya berfokus pada *platform Android* saja).

a. Keuntungan:

1. Menghemat waktu dan biaya pengembangan karena tidak perlu menulis kode terpisah untuk setiap platform.

2. Memastikan konsistensi antarmuka dan pengalaman pengguna di berbagai perangkat.

b. Contoh Penggunaan:

Aplikasi yang ingin menjangkau banyak platform tanpa memerlukan tim pengembang terpisah untuk masing-masing platform.

2. Penggunaan *Widget* untuk antarmuka pengguna

Flutter menggunakan sistem *Widget* yang fleksibel untuk membangun antarmuka pengguna.

a. Keunggulan *Widget*:

1. Setiap elemen antarmuka (seperti tombol, teks, atau tata letak) adalah *Widget* yang dapat disesuaikan.
2. Mendukung desain yang responsif dan dinamis, memudahkan pengembangan antarmuka yang menarik.

b. Manfaat:

1. Pengembang memiliki kontrol penuh atas tampilan dan interaksi aplikasi.

2.2.5.2 Komponen Utama *Flutter*

1. *Dart*: Bahasa pemrograman yang digunakan

Flutter menggunakan *Dart*, bahasa pemrograman yang dirancang oleh *Google*.

a. Keunggulan *Dart*:

1. Mendukung pemrograman sinkron dan asinkron untuk mengelola proses yang berjalan secara bersamaan.
2. Kompilasi langsung ke kode mesin (native code) untuk meningkatkan performa aplikasi.
3. Mudah dipelajari, terutama bagi pengembang yang sudah familiar dengan bahasa seperti JavaScript atau Java.

Gambar 2. 4 Logo *Dart*

2. *Widget*: Elemen antarmuka pengguna

Widget adalah blok bangunan utama dalam *Flutter* untuk membangun antarmuka pengguna.

a. Jenis *Widget*:

1. *Stateless Widget*: *Widget* yang tidak berubah (statis) setelah dibuat.
2. *Stateful Widget*: *Widget* yang dapat berubah berdasarkan interaksi atau data dinamis.

b. Fleksibilitas:

Widget dapat disusun, dikombinasikan, atau dimodifikasi untuk menciptakan antarmuka yang kompleks

3. *Hot Reload*: Fitur untuk mempercepat proses pengembangan

Flutter menyediakan fitur *Hot Reload* yang memungkinkan pengembang melihat perubahan kode secara instan tanpa harus memulai ulang aplikasi.

a. Manfaat *Hot Reload*:

1. Mengurangi waktu *debugging* dan pengujian.
2. Memudahkan pengembang bereksperimen dengan desain atau logika aplikasi secara langsung.

2.2.6 Figma

Figma adalah sebuah aplikasi web kolaboratif untuk user interface (UI), dengan fitur-fitur luring tambahan yang tersedia pada aplikasi desktop untuk Windows dan macOS. Kumpulan fitur Figma berfokus pada pengalaman pengguna dan antarmuka pengguna (UX/UI) dengan

penekanan pada kolaborasi real-time, menggunakan berbagai editor grafik vektor dan peralatan prototyping (pembuatan purwarupa). Versi aplikasi mobile untuk Android dan iOS memungkinkan untuk melihat dan berinteraksi dengan prototipe secara real-time pada perangkat smartphone dan tablet. (Sumber : wikipedia.org).



Gambar 2. 5 Logo *Figma*

2.2.7 Tentang Aplikasi

Aplikasi yang akan dibuat memiliki nama “AMANDI” yang berarti “Aplikasi Manajemen Keuangan Pribadi”. Filosofi logo aplikasi ini adalah sebagai berikut :

1. Bentuk Dua Segitiga yang Bertemu

Warna Biru dan Merah melambangkan dua sisi keuangan:

- a. Biru: stabilitas, kepercayaan, dan kontrol (melambangkan pemasukan).
- b. Merah: energi, aksi, dan kewaspadaan (melambangkan pengeluaran).

Perpaduan dua bentuk ini menciptakan kesan seimbang, mencerminkan manajemen keuangan yang seimbang dan terorganisir.

2. Roket yang Meluncur dari Tengah Logo

Melambangkan pertumbuhan, percepatan, dan kemajuan finansial. Roket yang muncul dari titik pertemuan biru dan merah menunjukkan bahwa: "Dengan mengelola pemasukan dan pengeluaran secara bijak, pengguna dapat mencapai kemajuan finansial dengan cepat."

3. Arah Diagonal dan Dinamis

Elemen desain yang miring memberikan kesan dinamis dan modern, cocok untuk aplikasi yang ingin mempermudah pengelolaan keuangan pribadi secara praktis dan efisien.

4. Tipografi “AMANDI”

Font tegas dan rapi menekankan kepercayaan dan profesionalisme. Nama AMANDI bisa diartikan sebagai akronim dari “*Aplikasi Manajemen Keuangan Pribadi*”, memperjelas identitas dan fungsi aplikasi.

Logo ini menggambarkan **semangat pengguna untuk mengelola keuangan secara bijak**, menjaga keseimbangan antara pemasukan dan pengeluaran, serta mendorong mereka menuju **kemajuan finansial yang pesat dan terarah**, seperti roket yang meluncur menuju tujuan.



Gambar 2. 6 Logo AMANDI

2.2.8 Model Pengembangan Sistem

2.2.8.1 Metode Waterfall

Metode Waterfall adalah model pengembangan perangkat lunak yang sistematis dan berurutan, dengan tahapan yang harus diselesaikan sebelum beralih ke tahap berikutnya. Model ini sering digunakan untuk proyek dengan persyaratan yang sudah jelas sejak awal. Berikut adalah tahapan dalam metode Waterfall:

1. Analisis kebutuhan sistem

Tahap ini berfokus pada pengumpulan dan analisis kebutuhan sistem.

a. Tujuan:

1. Mendefinisikan fitur dan fungsi yang harus dimiliki sistem.

b. Contoh:

1. Pengguna membutuhkan fitur pencatatan pemasukan dan pengeluaran (CRUD).
2. Sistem harus berjalan secara *offline* tanpa koneksi internet.

2. Desain sistem

Tahap ini adalah proses merancang bagaimana sistem akan dibangun, mencakup desain antarmuka dan struktur database.

a. Tujuan:

1. Membuat *blueprint* sistem, termasuk *Entity Relationship Diagram* (ERD) untuk menggambarkan hubungan data.
2. Merancang antarmuka pengguna (UI) yang sesuai dengan kebutuhan dan mudah digunakan.

b. Hasil:

1. Desain diagram alur.
2. Prototipe awal dari tampilan aplikasi.

3. Implementasi

Pada tahap ini, sistem yang telah dirancang mulai dikembangkan sesuai dengan desain yang telah dibuat.

a. Tujuan:

1. Menulis kode program berdasarkan desain sistem.
2. Mengintegrasikan komponen sistem seperti antarmuka pengguna dan database.

b. Contoh:

1. Membangun aplikasi menggunakan *Framework Flutter*.
2. Mengintegrasikan database *SQLite* untuk menyimpan data keuangan.

4. Pengujian dan evaluasi

Tahap ini bertujuan untuk memastikan sistem berfungsi sesuai dengan kebutuhan yang telah ditentukan.

a. Tujuan:

1. Mengidentifikasi dan memperbaiki bug.
2. Melakukan pengujian fungsionalitas untuk memastikan semua fitur bekerja dengan baik.

b. Contoh:

1. Menguji operasi CRUD untuk pencatatan keuangan.
2. Memastikan aplikasi tetap berjalan dengan baik meskipun dalam kondisi *offline*.

5. Pemeliharaan

Tahap akhir dalam model *Waterfall*, di mana sistem yang telah dirilis dirawat dan diperbarui jika ditemukan masalah atau kebutuhan baru.

a. Tujuan:

1. Menyediakan dukungan teknis untuk pengguna.
2. Melakukan pembaruan sistem jika diperlukan.

b. Contoh:

1. Menambahkan fitur baru berdasarkan umpan balik pengguna.
2. Memperbaiki *bug* yang ditemukan setelah aplikasi digunakan

2.2.8.2 Penggunaan Database *SQLite*

SQLite adalah sistem manajemen database relasional yang ringan dan cocok untuk aplikasi Mobile. (Sumber: *SQLite Documentation*, 2024).

1. Keunggulan *SQLite*:

- a. Tidak memerlukan server, sehingga cocok untuk aplikasi offline.
- b. Cepat dan efisien karena menggunakan format file tunggal.

- c. Mudah diintegrasikan dengan *Framework* seperti *Flutter*.
- 2. Penerapan pada Aplikasi:
 - a. Menyimpan data pemasukan dan pengeluaran.
 - b. Memastikan data tetap tersedia meskipun aplikasi tidak terhubung ke internet



Gambar 2. 7 Logo *SQLite*

2.2.8.3 Perancangan ERD (Entity Relationship Diagram)

ERD adalah diagram yang digunakan untuk memodelkan hubungan antar entitas dalam sistem.

1. Komponen Utama:

- a. Entitas: Representasi dari objek nyata dalam sistem, seperti pengguna, pemasukan, dan pengeluaran.
- b. Atribut: Properti atau informasi yang dimiliki oleh setiap entitas, misalnya nama pengguna, jumlah pemasukan, dan tanggal transaksi.
- c. Hubungan: Menjelaskan keterkaitan antara entitas, misalnya hubungan antara pengguna dengan data keuangan mereka.

2. Contoh Hubungan:

- a. Pengguna memiliki banyak Pemasukan.
- b. Pengguna memiliki banyak Pengeluaran.

2.2.8.4 HIPO (Hierarchical Input Process Output)

HIPO adalah pendekatan sistematis dalam merancang sistem yang menggambarkan hubungan *input*, proses, dan output.

A. Penerapan pada Sistem yang akan dibuat

Sistem manajemen keuangan berbasis *Mobile* yang dirancang menggunakan *Flutter* dengan database *SQLite* memiliki alur kerja yang sederhana namun fungsional. Berikut adalah penjelasan penerapan berdasarkan *input*, proses, dan output

1. *Input*: Data pemasukan dan pengeluaran

Input adalah data yang dimasukkan oleh pengguna ke dalam sistem untuk dicatat dan diolah.

a. Jenis Data:

1. Pemasukan:

- a. Jumlah pemasukan.
- b. Kategori pemasukan.
- c. Tanggal pemasukan.
- d. Keterangan

2. Pengeluaran:

- a. Jumlah pengeluaran.
- b. Kategori pengeluaran.
- c. Tanggal pengeluaran.
- d. Keterangan

b. Proses *Input*:

1. Pengguna mengisi form yang telah disediakan di aplikasi.
2. Data yang dimasukkan diverifikasi agar valid (contoh: jumlah tidak boleh negatif).

2. Proses: Penyimpanan dan manipulasi data

Data yang dimasukkan akan diproses dan disimpan di database *SQLite*.

a. Penyimpanan Data:

1. Data pemasukan dan pengeluaran disimpan dalam tabel terpisah di database *SQLite*.

2. Setiap entri data memiliki ID unik untuk mempermudah identifikasi.

b. Manipulasi Data:

1. *Create* (C): Pengguna dapat menambahkan data pemasukan atau pengeluaran baru.
2. *Read* (R): Data yang tersimpan dapat dilihat kembali dalam bentuk daftar transaksi.
3. *Update* (U): Pengguna dapat mengedit data jika terjadi kesalahan pencatatan.
4. *Delete* (D): Data yang tidak lagi relevan dapat dihapus.

3. *Output* : Daftar transaksi

Output adalah hasil yang ditampilkan kepada pengguna berdasarkan data yang telah diolah.

a. Tampilan *Output*:

1. Daftar pemasukan dan pengeluaran dalam format tabel atau daftar.
2. Setiap entri menampilkan:
 - a. Jumlah transaksi.
 - b. Kategori transaksi
 - c. Tanggal transaksi.
 - d. Keterangan.

b. Pengelompokan dan Penyajian Data:

1. Data dapat diurutkan berdasarkan tanggal untuk mempermudah pencarian.
2. Total pemasukan dan pengeluaran ditampilkan secara ringkas di bagian atas.
3. Perbedaan antara total pemasukan dan pengeluaran (saldo) juga disediakan.

BAB III

ANALISA DAN PERANCANGAN

3.1 Analisa Sistem

Analisis sistem pada penelitian “**Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile Menggunakan Framework Flutter***” dilakukan untuk memahami kebutuhan fungsional dan non-fungsional yang akan diterapkan pada aplikasi, serta alur proses yang akan diimplementasikan. Aplikasi ini hanya memiliki komponen inti CRUD (*Create, Read, Update, Delete*) untuk pencatatan transaksi keuangannya.

3.1.1 Identifikasi Kebutuhan Sistem

Pada tahap ini dilakukan identifikasi kebutuhan sistem yang mencakup kebutuhan fungsional dan non-fungsional. Kebutuhan ini menjadi acuan dalam perancangan dan pengembangan aplikasi.

1. Kebutuhan Fungsional

- a. Pencatatan Transaksi Keuangan (*Create*): Aplikasi harus dapat mencatat transaksi keuangan pengguna, baik berupa pemasukan maupun pengeluaran. Pengguna harus dapat memasukkan informasi transaksi seperti tanggal, kategori, jumlah uang, dan keterangan.
- b. Melihat Daftar Transaksi (*Read*): Aplikasi harus menampilkan daftar transaksi yang telah dicatat dalam urutan waktu tertentu. Pengguna dapat melihat total pemasukan, pengeluaran, dan saldo akhir berdasarkan transaksi yang tercatat.
- c. Memperbarui Transaksi (*Update*): Aplikasi harus memungkinkan pengguna untuk mengubah informasi pada transaksi yang telah dicatat sebelumnya, seperti mengedit jumlah atau keterangan transaksi.

- d. Menghapus Transaksi (*Delete*): Aplikasi harus memiliki fitur untuk menghapus transaksi yang tidak diperlukan atau salah *input*.

2. Kebutuhan Non-Fungsional

a. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam pembuatan dan implementasi Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* Menggunakan *Framework Flutter* adalah sebagai berikut:

Tabel 3. 1 Perangkat Keras Pembuatan

No	Tipe	Keterangan
1	Merk Laptop	HP
2	Type	HP Laptop 14s-fq1xxx
3	Processor	AMD Ryzen 7 5700U with Radeon Graphics
4	Memory	8 GB
5	Operating System	Windows 11

Tabel 3. 2 Perangkat Keras Implementasi

No	Tipe	Keterangan
1	Merk HP	Oppo
2	Type	Reno 3
3	Processor	Delapan-Inti
4	RAM	8 GB
5	Versi colorOS	V12.1

b. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam pembuatan dan implementasi Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* Menggunakan *Framework Flutter* adalah sebagai berikut:

1. Pembuatan program aplikasi menggunakan *Android Studio* dan *Framework Flutter*.
2. Bahasa pemrograman aplikasi menggunakan *Dart*.

3. Pembuatan database aplikasi menggunakan *database* lokal (*SQLite*).
4. Pembuatan desain aplikasi menggunakan *Figma*.

3.1.2 Analisis Proses

Pada tahap ini, dilakukan analisis alur proses yang mencakup interaksi pengguna dengan sistem operasi CRUD (*Create, Read, Update, Delete*).

1. *Use Case Diagram* :

1. Aktor Utama: Pengguna
2. Use Case Utama:
 - a. *Login*
 - b. *Registrasi*
 - c. Halaman Utama : Rekap harian dan daftar transaksi harian.
 - d. Halaman Kategori : Jenis kategori (Pemasukan/Pengeluaran) dan Tambah Kategori.
 - e. Halaman Tambah Transaksi : Jenis Transaksi (Pemasukan/Pengeluaran), Jumlah, Jenis kategori, Tanggal dan Keterangan.
 - f. Halaman Rekap : Pilihan bulan atah tahun, PDF (simpan rekap bulanan) dan daftar transaksi bulanan.
 - g. Halaman Profil : Foto profil, Username, Sisa saldo (keseluruhan) dan *Logout*.

2. Diagram Alur Proses

- a. Proses Pencatatan Transaksi (*Create*)
 1. Pengguna memilih opsi “Tambah Transaksi”.
 2. Pengguna mengisi detail transaksi seperti jenis transaksi (pemasukan/pengeluaran), jumlah, kategori, tanggal, dan keterangan.
 3. Pengguna menekan tombol “Simpan”.
 4. Sistem menyimpan transaksi ke dalam *database*.
 5. Sistem menampilkan daftar transaksi yang telah diperbarui.

b. Proses Melihat Daftar Transaksi (*Read*)

1. Pengguna dapat melihat daftar transaksi harian pada halaman utama sesuai dengan tanggal yang dipilih.
2. Untuk melihat daftar transaksi bulanan, pengguna dapat memilih halaman rekap bulanan, maka seluruh daftar transaksi pada bulan yang dipilih dapat dilihat. Pengguna juga bisa melihat total pemasukan dan pengeluaran bulanan pada halaman ini.

c. Proses Memperbarui Transaksi (*Update*)

1. Pengguna memilih transaksi yang ingin diperbaharui pada halaman utama.
2. Pengguna mengubah informasi yang ingin diperbarui (misal : jumlah, kategori, tanggal atau keterangan).
3. Pengguna menekan tombol “Simpan”.
4. Sistem menyimpan perubahan pada transaksi.
5. Sistem menampilkan daftar transaksi yang telah diperbarui pada halaman utama.

d. Proses Menghapus Transaksi (*Delete*)

1. Pengguna memilih transaksi yang ingin dihapus pada halaman utama.
2. Pengguna menekan tombol “Hapus”.
3. Sistem menampilkan konfirmasi hapus transaksi. Jika pengguna memilih batal maka sistem akan kembali ke halaman utama (membatalkan perintah hapus transaksi). Jika pengguna memilih hapus maka sistem akan menghapus transaksi yang dipilih dan menampilkan dialog “Transaksi berhasil dihapus”.
4. Sistem menampilkan daftar transaksi yang telah diperbarui pada halaman utama.

3.1.3 Analisis Struktur Data

Struktur data dalam aplikasi ini hanya melibatkan tabel transaksi, yang digunakan untuk menyimpan informasi terkait pencatatan keuangan pengguna. Desain tabel dalam basis data dirancang agar sederhana dan mudah dikelola.

1. Tabel Pengguna

- a. *Id_user*
- b. *Username*
- c. *Password*

2. Tabel Kategori

- a. *Id_Kategori*
- b. *Id_user*
- c. *Nama_Kategori*
- d. *Type_Kategori* : Pemasukan atau Pengeluaran

3. Tabel Transaksi

- a. *Id_Transaksi*
- b. *Id_user*
- c. *Keterangan*
- d. *Id_Kategori*
- e. *Tanggal_Transaksi*
- f. *Jumlah*

3.1.4 Analisis Antarmuka Pengguna

Antarmuka pengguna (*User Interface*) dirancang agar sederhana dan mudah dipahami. Struktur antarmuka meliputi beberapa halaman utama sebagai berikut :

1. Halaman Utama : Menampilkan rekap saldo harian, daftar transaksi harian serta navigasi ke halaman kategori, halaman tambah transaksi, halaman rekap bulanan dan halaman profil.

2. Halaman Kategori : Menampilkan jenis kategori (pemasukan/pengeluaran), tambah kategori dan daftar kategori yang telah disimpan.
3. Halaman Tambah Transaksi : Menampilkan formulir *input* untuk mencatat transaksi baru. Formulir ini mencakup *input* jumlah, kategori, tanggal dan keterangan.
4. Halaman Rekap Bulanan : Menampilkan rekap bulanan yang dipilih.
5. Halaman Profil : Menampilkan foto profil, nama *user* dan tombol *Logout*.

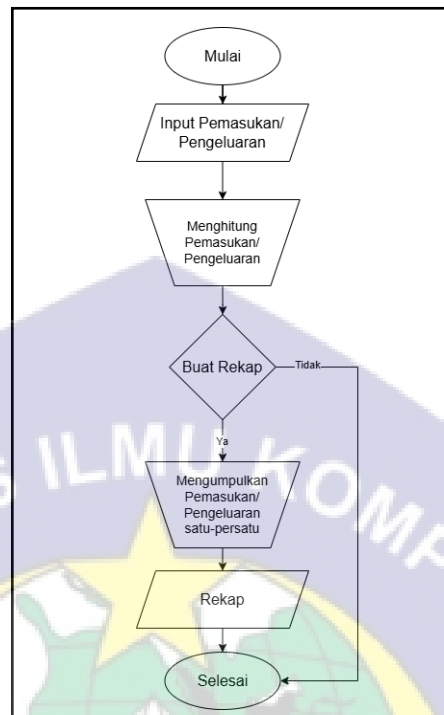
3.1.5 Analisis Kinerja Sistem

Kinerja aplikasi ini dinilai dari seberapa cepat aplikasi dapat menambah, memperbarui, menampilkan, dan menghapus data transaksi. Karena hanya menggunakan operasi CRUD (*Create, Read, Update, Delete*) dan tidak memiliki proses yang kompleks, aplikasi ini diharapkan memiliki kinerja yang sangat optimal dan mudah digunakan.

3.1.6 Sistem yang Berjalan

Sistem yang berjalan menggambarkan bagaimana pengguna saat ini mengelola keuangan pribadinya sebelum adanya aplikasi yang diusulkan, yaitu :

1. Pengguna mencatat pemasukan dan pengeluaran secara manual
2. Pengguna menghitung saldo akhir secara manual
3. Jika ingin melihat laporan, pengguna harus mengumpulkan data satu per satu.

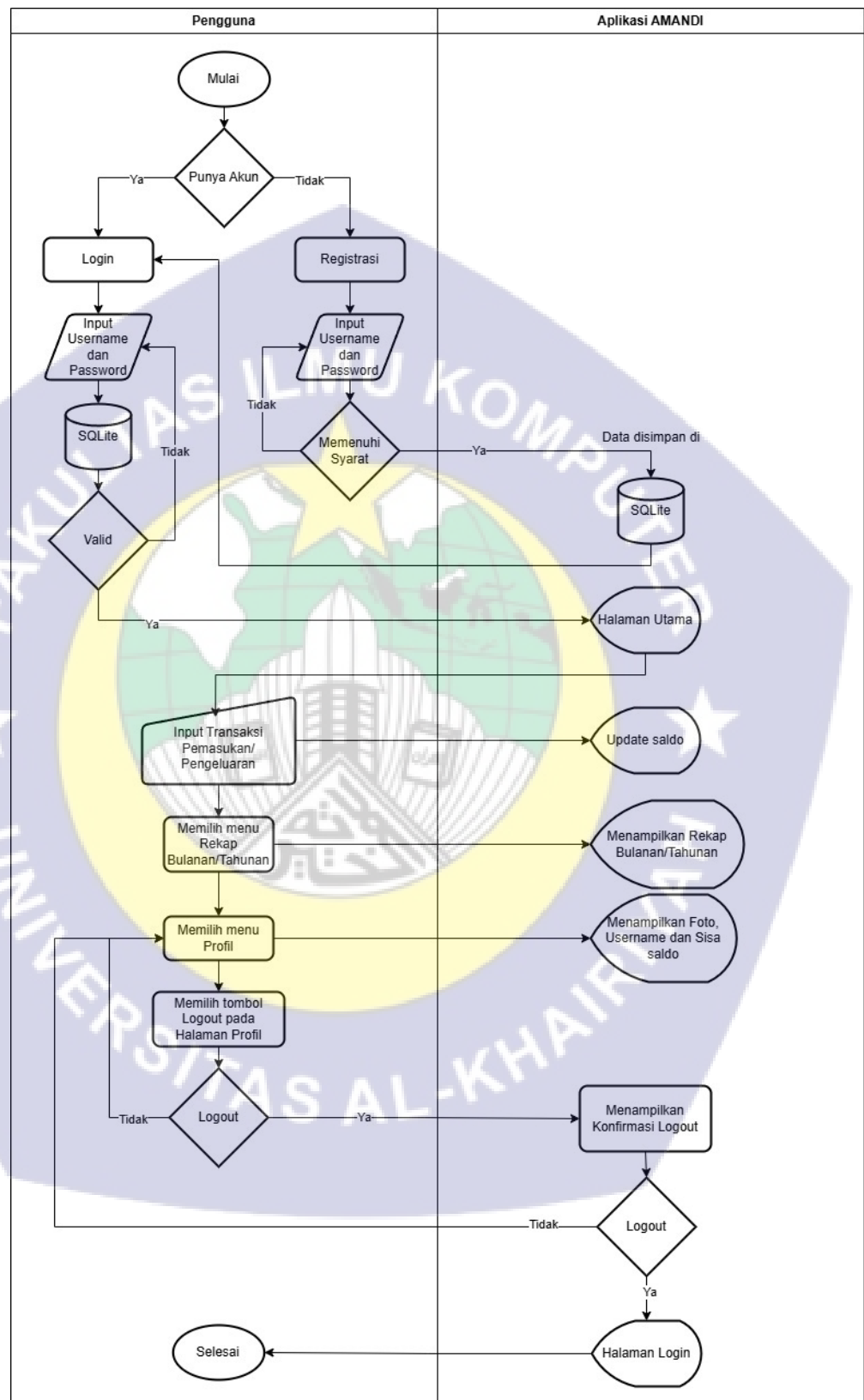


Gambar 3. 1 *Flowchart* Sistem yang Berjalan

3.1.7 Sistem yang diusulkan

Sistem yang diusulkan adalah aplikasi manajemen keuangan pribadi berbasis mobile menggunakan *Flutter* dan *SQLite* sebagai database. Keunggulan sistem ini dibandingkan sistem yang berjalan :

1. Pengguna harus Registrasi terlebih dahulu. Jika sudah mempunyai akun, pengguna bisa login ke aplikasi.
2. Aplikasi menampilkan halaman utama.
3. Pengguna mencatat pemasukan atau pengeluaran.
4. Sistem secara otomatis memperbarui saldo harian.
5. Pengguna dapat menambahkan kategori pemasukan dan pengeluaran.
6. Pengguna bisa melihat rekap pada halaman rekap bulanan.
7. Terdapat juga halaman profil yang menampilkan nama *user* dan sisa saldo akhir.



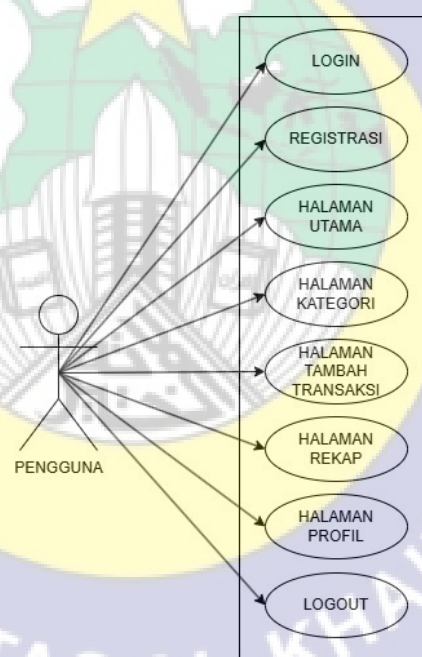
Gambar 3. 2 Flowmap Sistem yang diusulkan

3.2 Perancangan Sistem

Perancangan sistem pada aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* ini dirancang agar dapat memenuhi kebutuhan dasar pengguna untuk melakukan pencatatan transaksi dengan operasi CRUD (*Create, Read, Update, Delete*). Perancangan ini mencakup perancangan *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, *Sequence Diagram* dan perancangan antarmuka pengguna.

3.2.1 Use Case Diagram

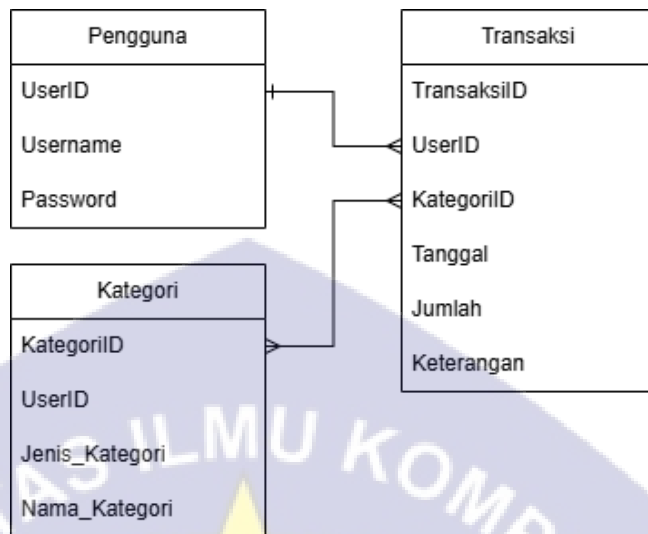
Berikut adalah perancangan sistem dalam bentuk *Use Case Diagram* pada sistem yang akan dibuat.



Gambar 3. 3 *Use Case Diagram*

3.2.2 Class Diagram

Berikut adalah tabel yang perlu dibuat untuk *Class Diagram* pada sistem yang diusulkan dengan operasi CRUD (*Create, Read, Update, Delete*):

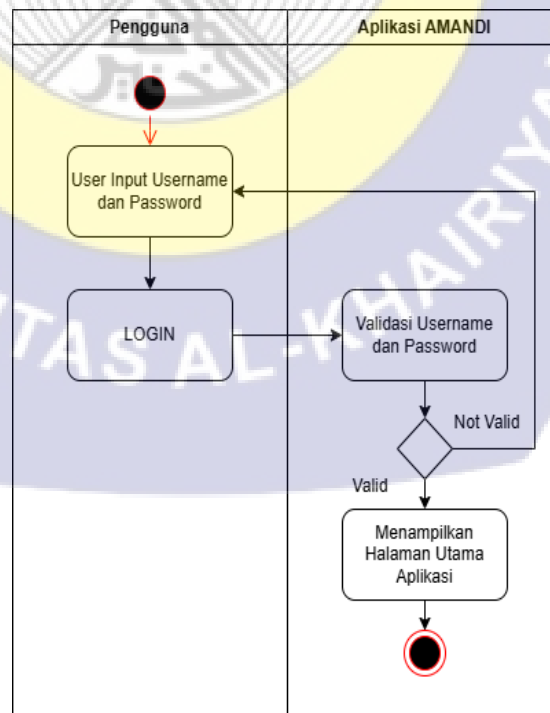


Gambar 3. 4 Class Diagram

3.2.3 Activity Diagram

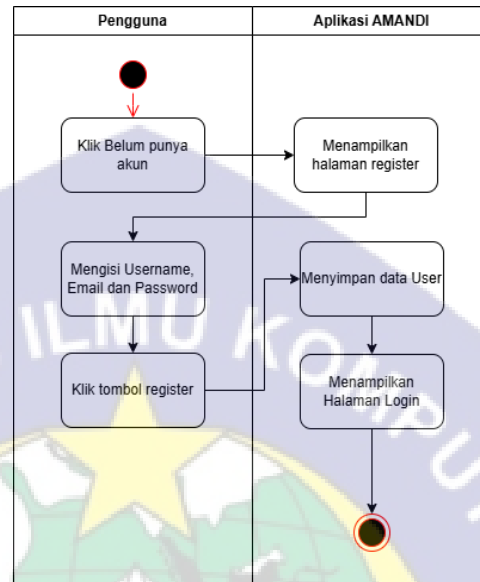
Activity Diagram ini dibuat untuk memberikan gambaran tentang alur kerja dan proses dalam aplikasi yang diusulkan. Berikut *Activity diagram* yang diusulkan :

1. Activity Diagram Login



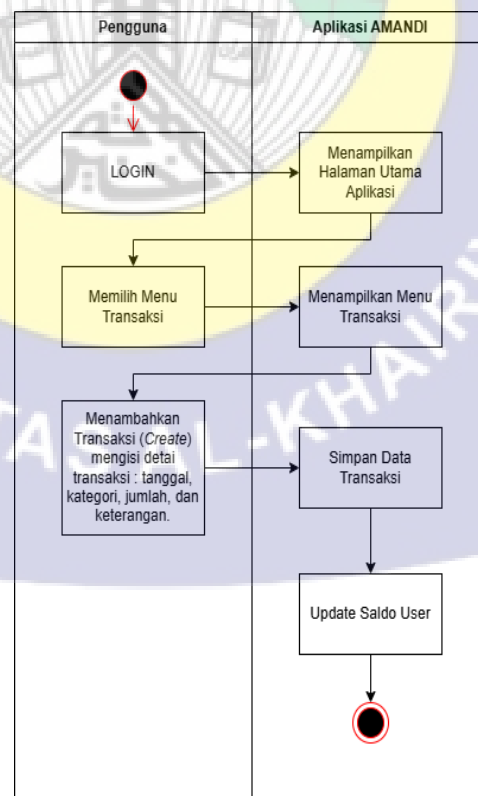
Gambar 3. 5 Activity Diagram Login

2. Activity Diagram Register



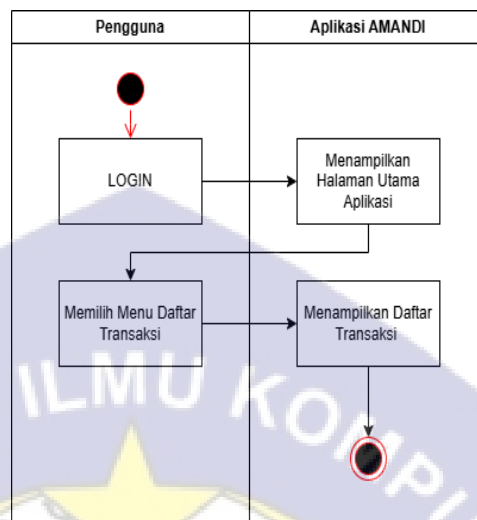
Gambar 3. 6 Activity Diagram Register

3. Activity Diagram Create



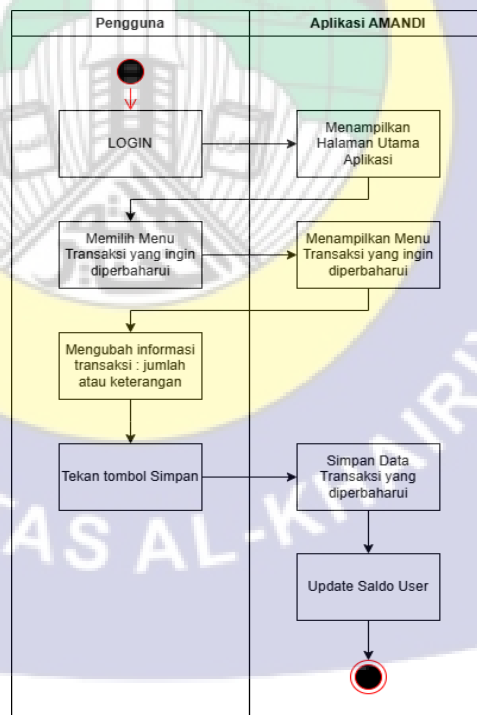
Gambar 3. 7 Activity Diagram Create

4. Activity Diagram Read



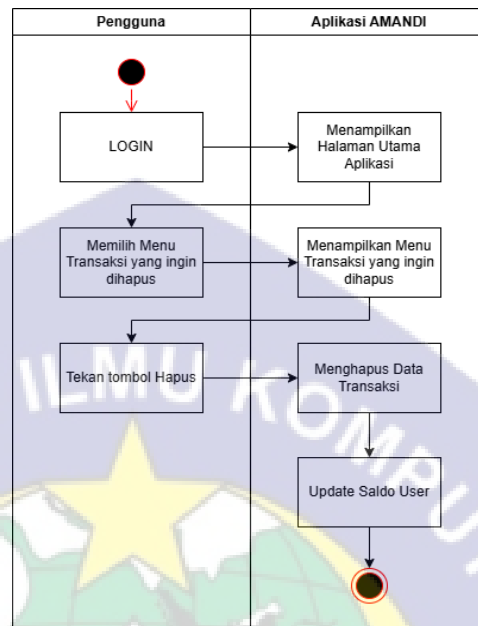
Gambar 3. 8 Activity Diagram Read

5. Activity Diagram Update



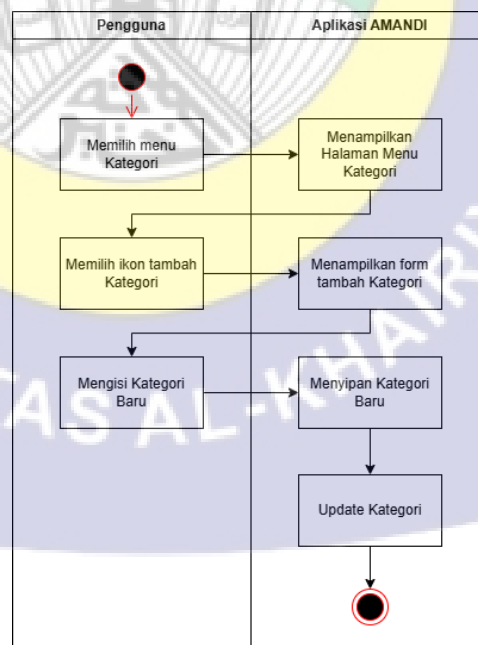
Gambar 3. 9 Activity Diagram Update

6. Activity Diagram Delete



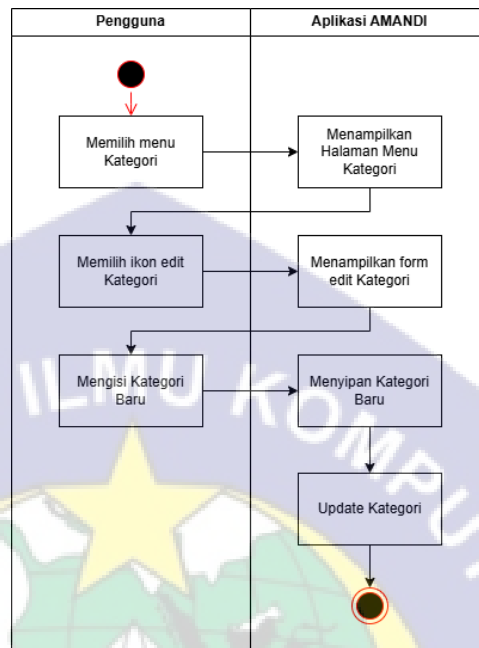
Gambar 3. 10 Activity Diagram Delete

7. Activity Diagram Tambah Kategori



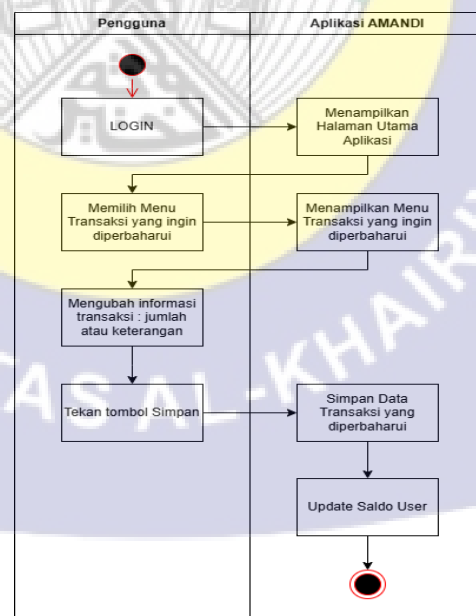
Gambar 3. 11 Activity Diagram Tambah Kategori

8. Activity Diagram Ubah Kategori



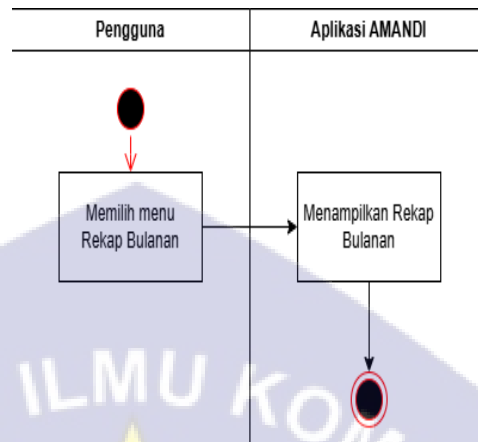
Gambar 3. 12 Activity Diagram Tambah Kategori

9. Activity Diagram Hapus Kategori



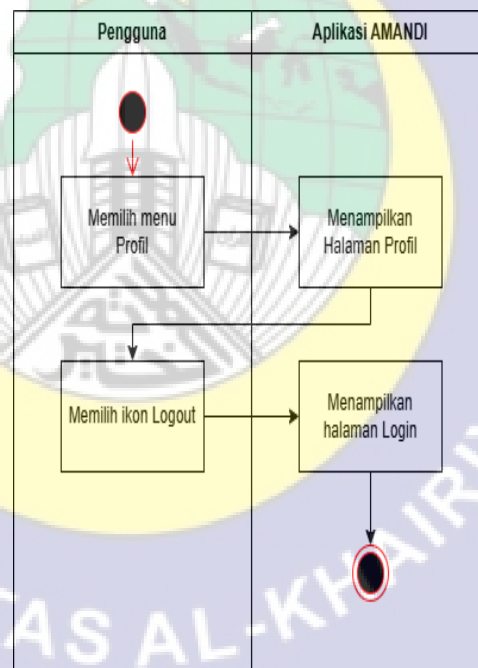
Gambar 3. 13 Activity Diagram Hapus Kategori

10. Activity Diagram Rekap Bulanan



Gambar 3. 14 Activity Diagram Rekap Bulanan

11. Activity Diagram Logout

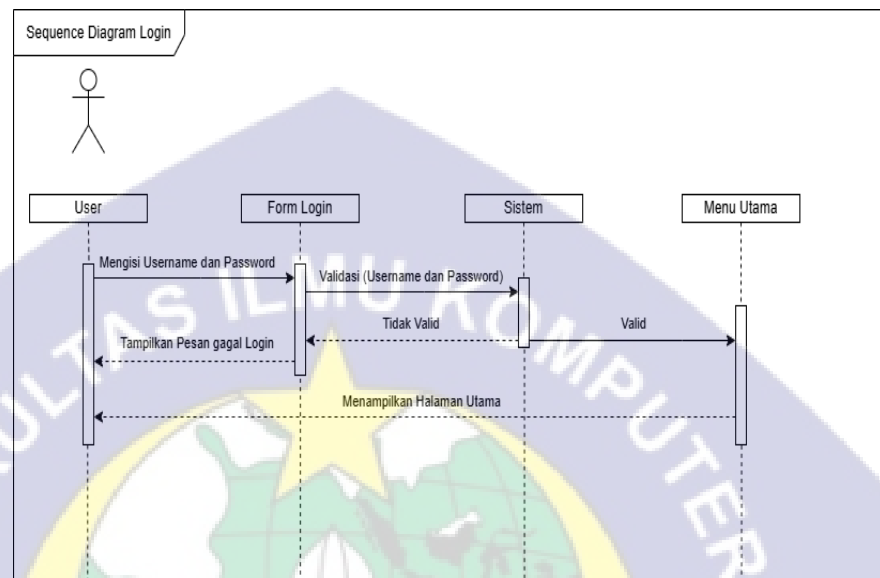


Gambar 3. 15 Activity Diagram Logout

3.2.4 Sequence Diagram

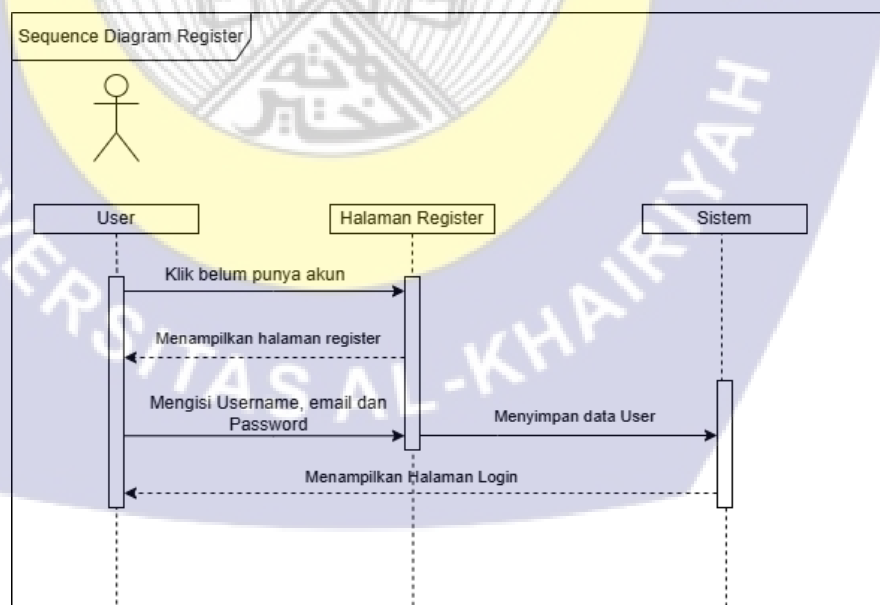
Sequence Diagram ini dibuat untuk alur interaksi kerja dan proses dalam aplikasi yang diusulkan. Berikut *Sequence Diagram* yang diusulkan:

1. Sequence Diagram Login



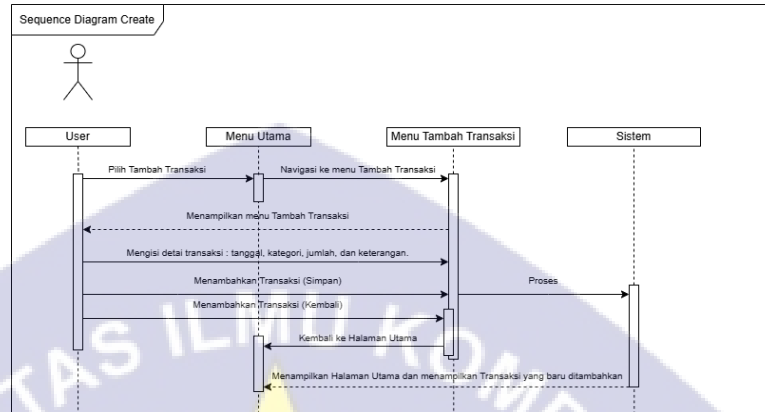
Gambar 3. 16 Sequence Diagram Login

2. Sequence Diagram Register



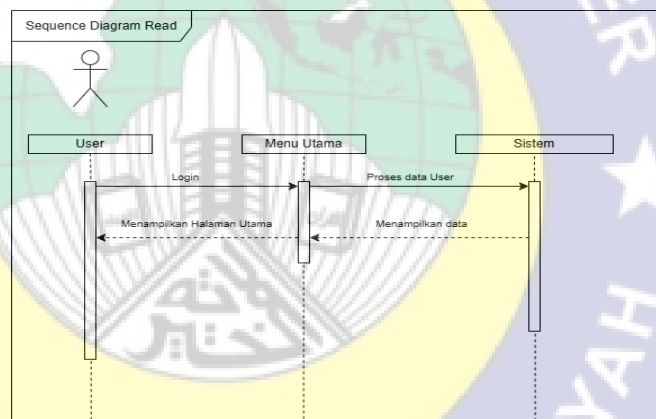
Gambar 3. 17 Sequence Diagram Register

3. Sequence Diagram Create



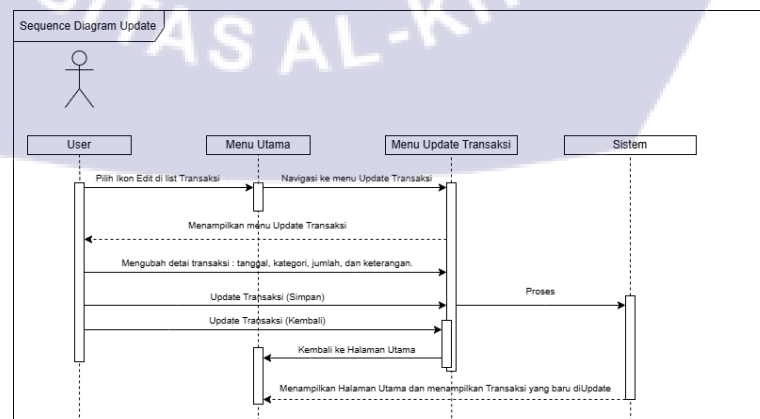
Gambar 3. 18 Sequence Diagram Create

4. Sequence Diagram Read



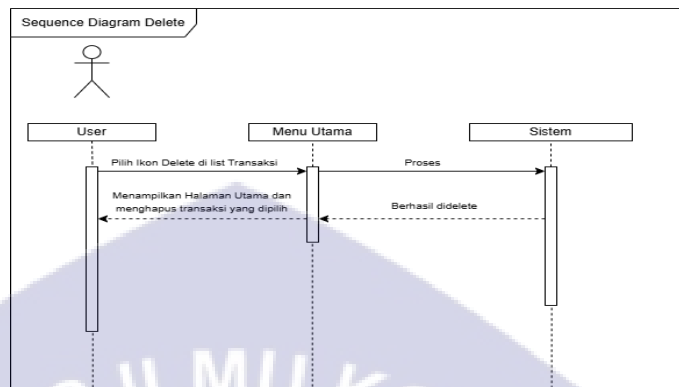
Gambar 3. 19 Sequence Diagram Read

5. Sequence Diagram Update



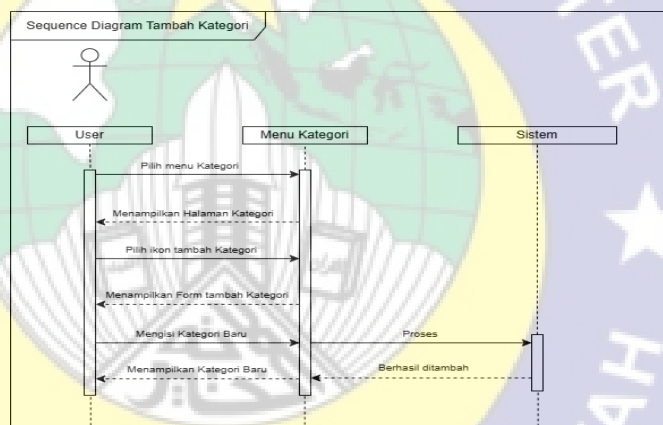
Gambar 3. 20 Sequence Diagram Update

6. Sequence Diagram Delete



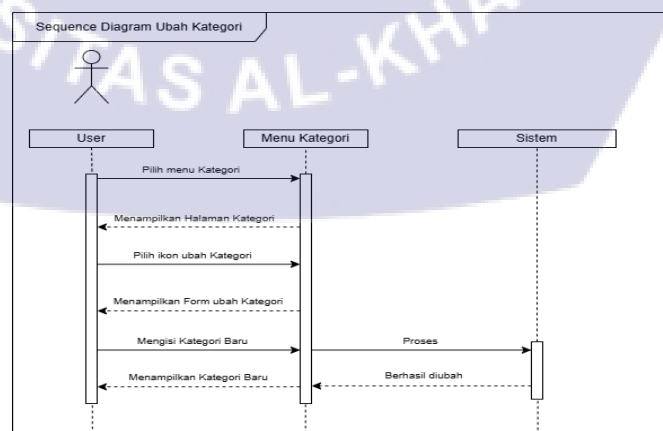
Gambar 3. 21 Sequence Diagram Delete

7. Sequence Diagram Tambah Kategori



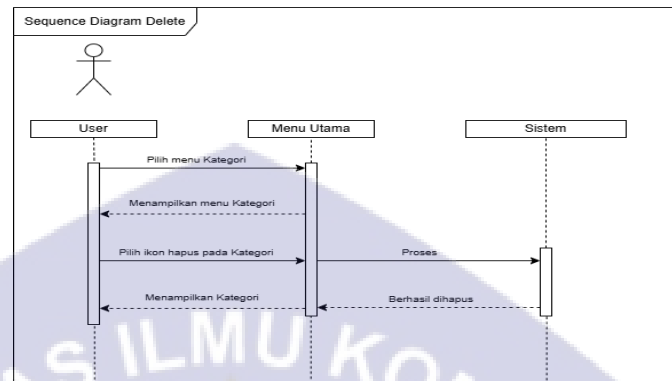
Gambar 3. 22 Sequence Diagram Tambah Kategori

8. Sequence Diagram Ubah Kategori



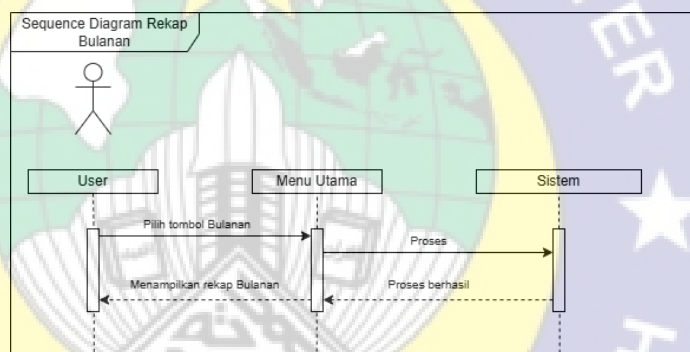
Gambar 3. 23 Sequence Diagram Ubah Kategori

9. Sequence Diagram Hapus Kategori



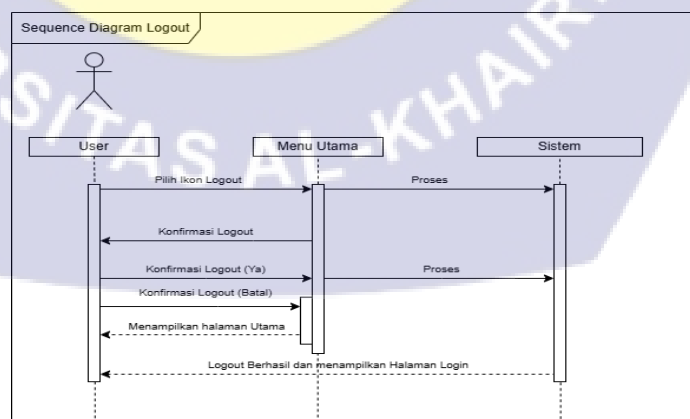
Gambar 3. 24 Sequence Diagram Hapus Kategori

10. Sequence Diagram Rekap



Gambar 3. 25 Sequence Diagram Rekap Bulanan

11. Sequence Diagram Logout



Gambar 3. 26 Sequence Diagram Logout

3.3 Perancangan *Database*

Perancangan *database* pada aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* ini menggunakan *SQLite* (*database* lokal).

Keunggulan menggunakan *SQLite* adalah :

1. Ringan dan Cepat
2. *SQLite* memiliki ukuran yang kecil dan tidak membutuhkan banyak sumber daya.
3. Tidak Memerlukan Server
4. Data disimpan dalam satu file di perangkat, sehingga lebih simpel dalam implementasi.
5. Mudah digunakan dan dikonfigurasi
6. Tidak membutuhkan instalasi atau pengaturan yang rumit
7. Cukup menyertakan file *database* (*.sqlite*) dalam aplikasi dan langsung dapat digunakan.
8. Hemat Memori dan Penyimpanan
9. Cocok untuk aplikasi *mobile* karena tidak membebani kapasitas penyimpanan perangkat.
10. Dapat digunakan pada aplikasi yang berjalan secara offline.
11. Data tetap aman meskipun terjadi error atau aplikasi crash.
12. Banyak digunakan dalam aplikasi *Android* sebagai penyimpanan data lokal.

3.4 Desain *Database*

1. Tabel *User*

Berikut adalah tabel *User* yang terdapat dalam Aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* Menggunakan *Framework Flutter*.

Tabel 3. 3 Tabel *User*

No	Nama Kolom	Tipe Data	Keterangan
1	<i>Id_user</i>	<i>Integer</i>	<i>Id User</i>
2	<i>Username</i>	<i>Text</i>	<i>Username</i> (min 3. Max 20)
3	<i>Password</i>	<i>Text</i>	<i>Password</i>

2. Tabel Kategori

Tabel 3. 4 Tabel Kategori

No	Nama Kolom	Tipe Data	Keterangan
1	<i>Id_Kategori</i>	<i>Integer</i>	Primary Key, auto increment
2	<i>Id_user</i>	<i>Integer</i>	<i>Id user</i> pemilik kategori
3	<i>Nama_Kategori</i>	<i>Text</i>	Nama kategori (min 1, max 60)
4	<i>Type_Kategori</i>	<i>Integer</i>	1 = pemasukan, 2 = pengeluaran

3. Tabel Transaksi

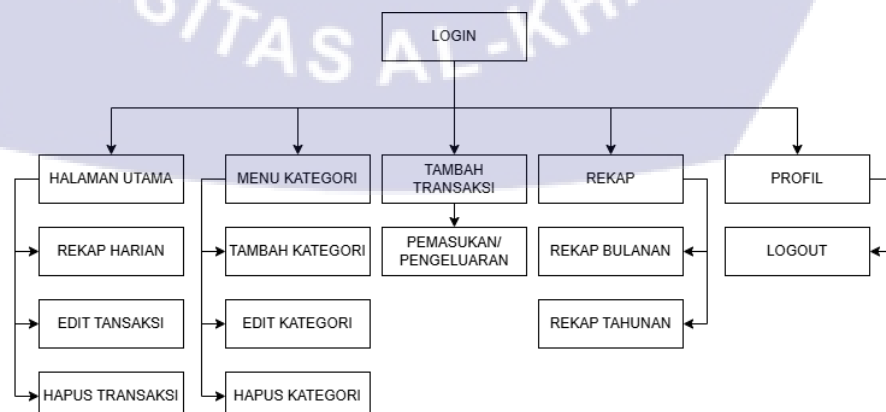
Tabel 3. 5 Tabel Transaksi

No	Nama Kolom	Tipe Data	Keterangan
1	<i>Id_Transaksi</i>	<i>Integer</i>	Primary Key, auto increment
2	<i>Id_user</i>	<i>Integer</i>	ID user pemilik transaksi
3	<i>Keterangan</i>	<i>Text</i>	Keterangan transaksi
4	<i>Id_Kategori</i>	<i>Integer</i>	ID kategori terkait
5	<i>Tanggal_Transaksi</i>	<i>Datetime</i>	Tanggal transaksi
6	<i>Jumlah</i>	<i>Integer</i>	Jumlah

3.5 Perancangan Antarmuka

Perancangan antarmuka pada aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* ini mencakup HIPO (*Hierarchy Input Proses Output*) dan UI (*User Interface*).

3.5.1 HIPO

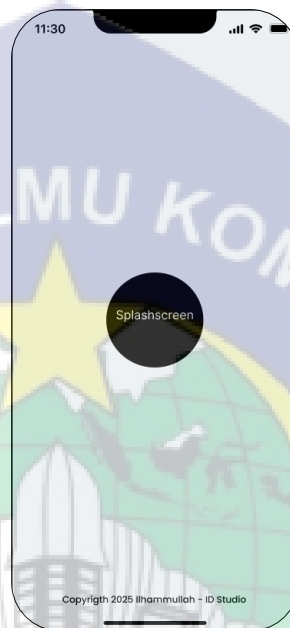


Gambar 3. 27 HIPO

3.5.2 UI (User Interface)

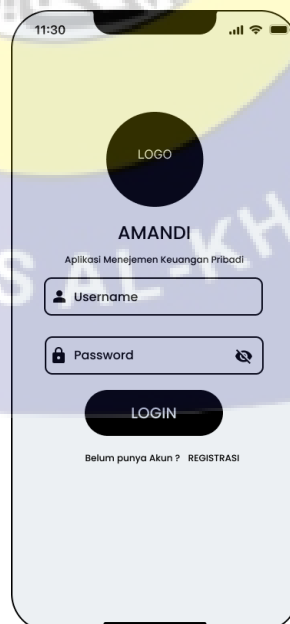
Perancangan antarmuka UI (*User Interface*) pada aplikasi Manajemen Keuangan Pribadi Berbasis *Mobile* ini menggunakan aplikasi *Figma*.

1. UI *Splashscreen*



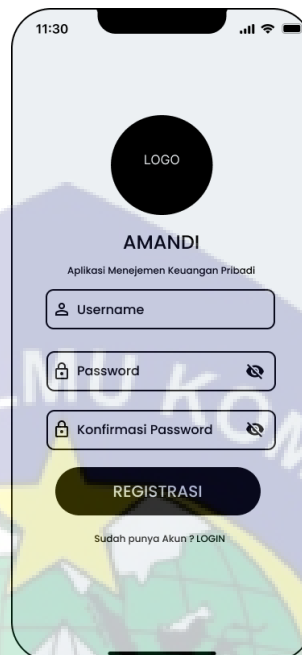
Gambar 3. 28 UI (*User Interface*) *Splashscreen*

2. UI (*User Interface*) *Login*



Gambar 3. 29 UI (*User Interface*) *Login*

3. UI (*User Interface*) Register



11:30

LOGO

AMANDI

Aplikasi Manajemen Keuangan Pribadi

Username

Password

Konfirmasi Password

REGISTRASI

Sudah punya Akun ? LOGIN

Gambar 3. 30 UI (*User Interface*) Register

4. UI (*User Interface*) Halaman Utama



11:30

Februari 2025

06 07 08 09 10
Kam Jum Sab Min Sen

Sisa Saldo Sebelumnya : Rp
Masuk Rp. 3.500.000 Keluar Rp. 25.000
Sisa Saldo Sekarang : Rp

Transaksi

Rp. 3.500.000
Transaksi Masuk

Rp. 25.000
Transaksi Keluar

Beranda Kategori Rekap Bulanan Profil

Gambar 3. 31 UI (*User Interface*) Halaman Utama

5. UI (*User Interface*) Tambah Pemasukan

11:30

← Tambah Transaksi

☒ Pemasukan

Jumlah
3.500.000

Kategori
Gaji Bulanan

Tanggal

Keterangan

Simpan

1 2 3
4 5 6
7 8 9
. 0 ✕

Gambar 3. 32 UI (*User Interface*) Tambah Pemasukan

6. UI (*User Interface*) Tambah Pengeluaran

11:30

← Tambah Transaksi

☒ Pengeluaran

Jumlah
3.500.000

Kategori
Makan

Tanggal

Keterangan

Simpan

1 2 3
4 5 6
7 8 9
. 0 ✕

Gambar 3. 33 UI (*User Interface*) Tambah Pengeluaran

7. UI (User Interface) Dropdown Kategori



Gambar 3. 34 UI (User Interface) Dropdown Kategori

8. UI (User Interface) Tanggal



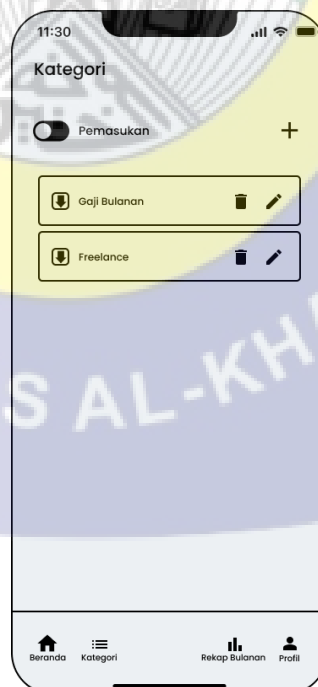
Gambar 3. 35 UI (User Interface) Tanggal

9. UI (*User Interface*) Keterangan



Gambar 3. 36 UI (*User Interface*) Login

10. UI (*User Interface*) Kategori Pemasukan



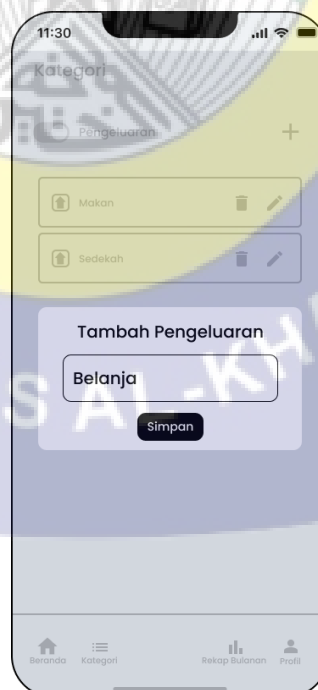
Gambar 3. 37 UI (*User Interface*) Kategori Pemasukan

11. UI (*User Interface*) Kategori Pengeluaran



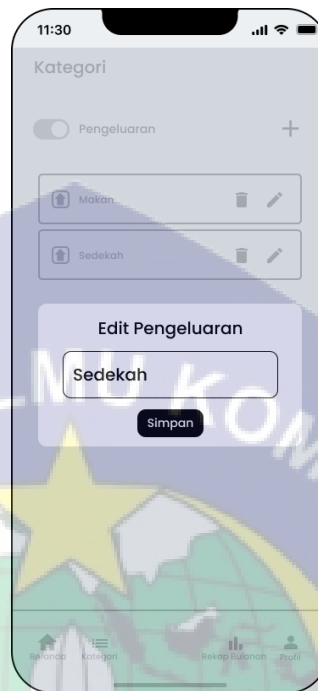
Gambar 3. 38 UI (*User Interface*) Kategori Pengeluaran

12. UI (*User Interface*) Tambah Kategori



Gambar 3. 39 UI (*User Interface*) Tambah Kategori

13. UI (*User Interface*) Ubah Kategori



Gambar 3. 40 UI (*User Interface*) Ubah Kategori

14. UI (*User Interface*) Rekap Bulanan



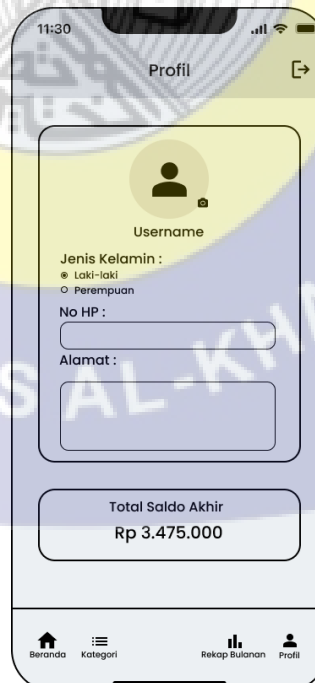
Gambar 3. 41 UI (*User Interface*) Rekap Bulanan

15. UI (User Interface) Review Rekap Bulanan



Gambar 3. 42 UI (User Interface) Review Rekap Bulanan

16. UI (User Interface) Halaman Profil



Gambar 3. 43 UI (User Interface) Halaman Profil

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi sistem Aplikasi Manajemen Keuangan Pribadi atau AMANDI dibangun menggunakan *Flutter* sebagai *framework* utama dengan bahasa *Dart*.

4.2 Implementasi Basis Data

Implementasi basis data atau *Database* pada sistem Aplikasi Manajemen Keuangan Pribadi atau AMANDI ini menggunakan *SQLite* yang diakses melalui *library Drift*, yang memungkinkan pengelolaan data secara reaktif dan efisien. Seluruh proses pengembangan dilakukan menggunakan *Android Studio* pada sistem operasi Windows 10. Berikut adalah struktur model tabel menggunakan *Drift* :

```
@DataClassName('User')
class Users extends Table {
  IntColumn get id => integer().autoIncrement();
  TextColumn get username => text().withLength(min: 3, max: 20());
  TextColumn get password => text();
}
```

Gambar 4. 1 *User*

Tabel 4. 1 *User*

No	Nama Kolom	Tipe Data	Keterangan
1	<i>Id_user</i>	<i>Integer</i>	<i>Id User</i>
2	<i>Username</i>	<i>Text</i>	<i>Username</i> (min 3. Max 20)
3	<i>Password</i>	<i>Text</i>	<i>Password</i>

```
@DataClassName('Category')
class Categories extends Table {
    IntColumn get id => integer().autoIncrement();
    IntColumn get userId => integer();
    TextColumn get name => text().withLength(min: 1, max: 60()); // Nama tidak boleh kosong
    IntColumn get type => integer().check(type.isIn([1, 2])); // Hanya 1 atau 2
```

Gambar 4. 2 Kategori

Tabel 4. 2 Kategori

No	Nama Kolom	Tipe Data	Keterangan
1	<i>Id_Kategori</i>	<i>Integer</i>	Primary Key, auto increment
2	<i>Id_user</i>	<i>Integer</i>	<i>Id user</i> pemilik kategori
3	Nama_Kategori	<i>Text</i>	Nama kategori (min 1, max 60)
4	<i>Type_Kategori</i>	<i>Integer</i>	1 = pemasukan, 2 = pengeluaran

```
class Transactions extends Table {
    IntColumn get id => integer().autoIncrement();
    IntColumn get userId => integer();
    TextColumn get description => text().withLength(max: 250());
    IntColumn get category_id => integer();
    DateTimeColumn get transaction_date => dateTime();
    IntColumn get amount => integer();
```

Gambar 4. 3 Transaksi

Tabel 4. 3 Transaksi

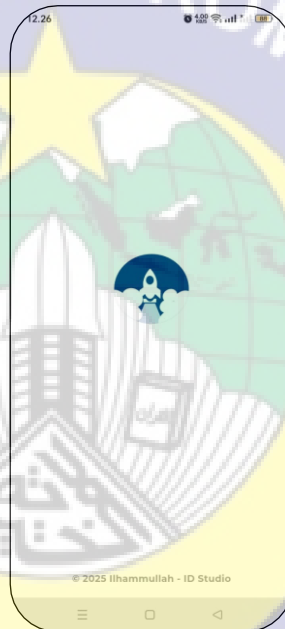
No	Nama Kolom	Tipe Data	Keterangan
1	<i>Id_Transaksi</i>	Integer	Primary Key, auto increment
2	<i>Id_user</i>	Integer	ID user pemilik transaksi
3	Keterangan	Text	Keterangan transaksi
4	<i>Id_Kategori</i>	Integer	ID kategori terkait
5	Tanggal_Transaksi	Datetime	Tanggal transaksi
6	Jumlah	Integer	Jumlah

4.3 Implementasi Program

Implementasi program merupakan tampilan antarmuka pengguna dari Aplikasi Manajemen Keuangan Pribadi AMANDI. Setiap halaman utama pada aplikasi ditampilkan dalam bentuk tangkapan layar (*screenshot*) disertai penjelasan fungsi dan komponennya.

4.3.1 Halaman *Splashscreen*

Halaman ini adalah layar awal yang ditampilkan saat aplikasi dimuat. Halaman ini menampilkan animasi *loading* dan nama pembuat.



Gambar 4. 4 Halaman *Splashscreen*

4.3.2 Halaman *Login*

Halaman ini digunakan oleh pengguna untuk masuk ke dalam aplikasi.

1. Input : Username dan password.
2. Validasi : Kolom tidak boleh kosong.
3. Fitur : Terdapat navigasi ke halaman Registrasi (Belum punya akun ? REGISTRASI). Terdapat tombol *eye/hide* pada kolom *Password*.
4. Tindakan : Login berhasil akan diarahkan ke Halaman Utama.

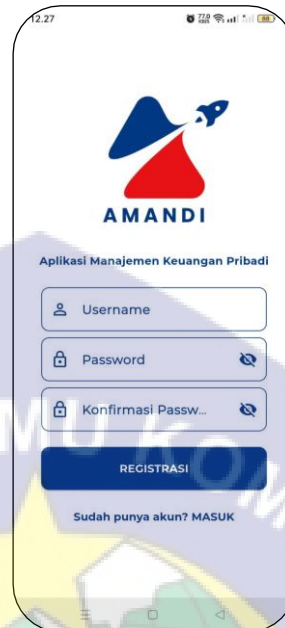


Gambar 4. 5 Halaman *Login*

4.3.3 Halaman Registrasi

Halaman ini digunakan oleh pengguna untuk membuat akun baru.

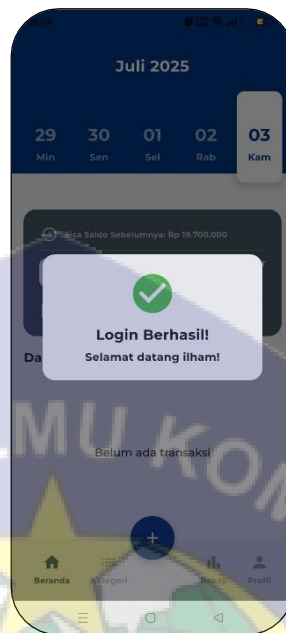
1. Input : Username, Password dan Konfirmasi Password.
2. Validasi : Field tidak boleh kosong. Username minimal 3 karakter, Password minimal 6 karakter.
3. Fitur : Terdapat navigasi ke halaman *Login* (Sudah punya akun ? MASUK). Terdapat tombol *eye/hide* pada kolom *Password* dan *Konfirmasi Password*.
4. Tindakan : Registrasi berhasil akan diarahkan ke Halaman *Login*.



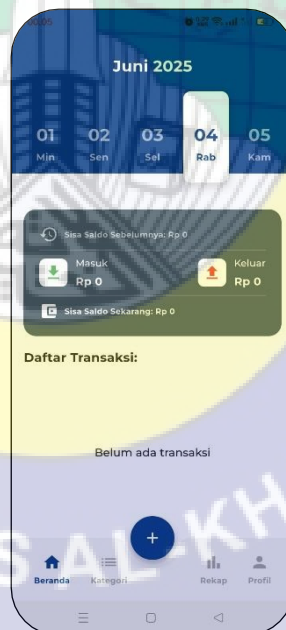
Gambar 4. 6 Halaman Registrasi

4.3.4 Halaman Utama

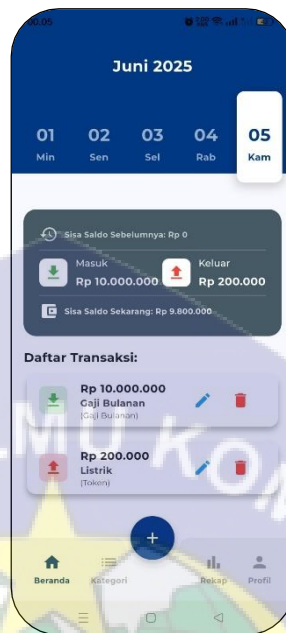
Saat pertama masuk aplikasi terdapat dialog “*Login* Berhasil!. Selamat datang (*username*)!”. Halaman Utama menampilkan tanggal hari ini, total pemasukan dan pengeluaran hari ini, serta daftar transaksi yang dilakukan pada hari tersebut. Pada halaman utama juga terdapat *Calendar appbar* yang berfungsi untuk menampilkan atau mengelola tampilan kalender dalam aplikasi.



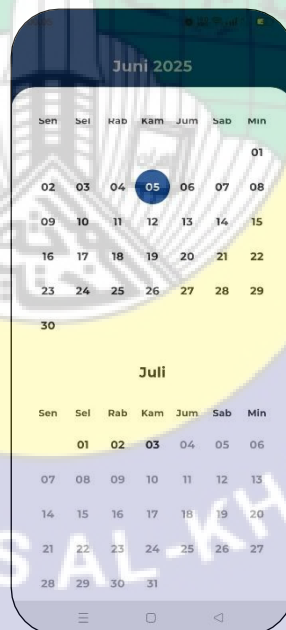
Gambar 4. 7 Halaman Utama (*Login Berhasil*)



Gambar 4. 8 Halaman Utama (*Transaksi kosong*)



Gambar 4. 9 Halaman Utama (Transaksi ada)

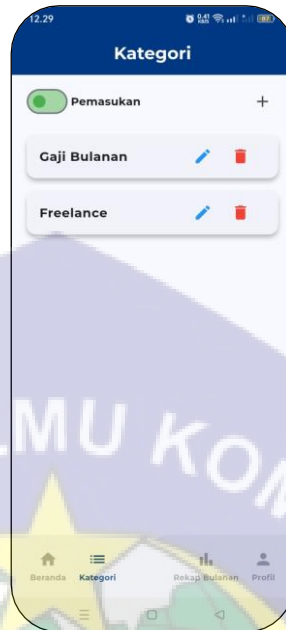


Gambar 4. 10 Halaman Utama (Calendar appbar)

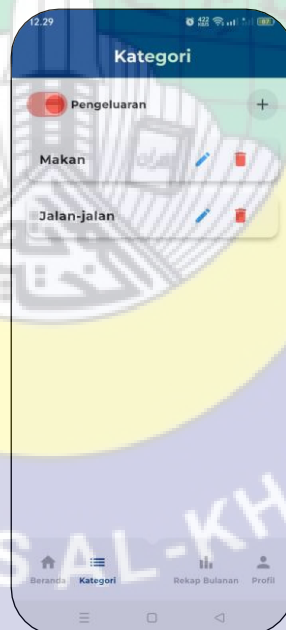
4.3.5 Halaman Kategori

Halaman Kategori digunakan untuk mengelola kategori pemasukan dan pengeluaran.

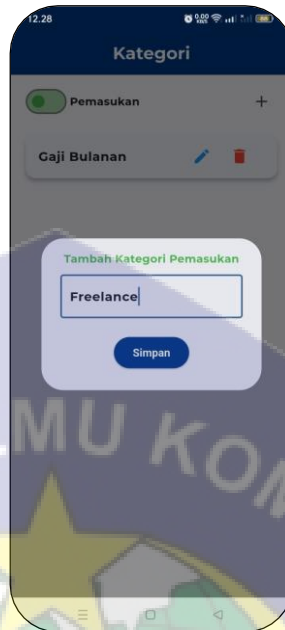
1. Fitur : Switch tipe kategori (pemasukan/pengeluaran), tombol tambah kategori, tombol ubah dan hapus kategori yang menampilkan *popup*.



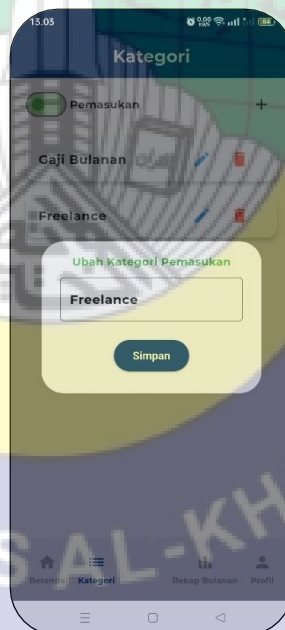
Gambar 4. 11 Halaman Kategori (Pemasukan)



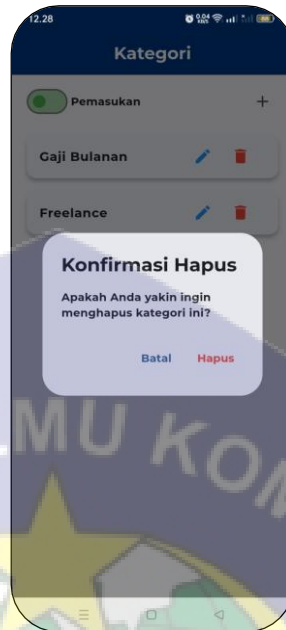
Gambar 4. 12 Halaman Kategori (Pengeluaran)



Gambar 4. 13 Halaman Kategori (Tambah Kategori)



Gambar 4. 14 Halaman Kategori (Ubah Kategori)



Gambar 4. 15 Halaman Kategori (Hapus Kategori)

4.3.6 Halaman Tambah Transaksi

Halaman tambah transaksi berisi formulir yang memungkinkan pengguna untuk menambahkan transaksi baru.

1. Fitur : Switch tipe transaksi (pemasukan/pengeluaran), input jumlah, pilih kategori, tanggal, dan keterangan.
2. Validasi : Jumlah tidak boleh 0, kategori, tanggal dan keterangan wajib diisi. Jika tambah transaksi disimpan akan diarahkan ke Halaman Utama.



The screenshot shows a mobile application interface for adding a transaction. The title bar is blue with a back arrow and the text "Tambah Transaksi". Below the title bar, there is a green toggle switch labeled "Pemasukan". The form contains the following fields: "Jumlah" (Amount) with a plus icon, "Pilih Kategori" (Select Category) with a dropdown menu showing "Gaji Bulanan", "Pilih Tanggal" (Select Date) with a calendar icon, and "Keterangan" (Description) with a text input field. A blue "Simpan" (Save) button is at the bottom.

Gambar 4. 16 Halaman Tambah Transaksi (Pemasukan)



The screenshot shows the same mobile application interface for adding a transaction, but for an expense. The title bar is blue with a back arrow and the text "Tambah Transaksi". Below the title bar, there is a red toggle switch labeled "Pengeluaran". The form contains the following fields: "Jumlah" (Amount) with a plus icon, "Pilih Kategori" (Select Category) with a dropdown menu showing "Makan", "Pilih Tanggal" (Select Date) with a calendar icon, and "Keterangan" (Description) with a text input field. A blue "Simpan" (Save) button is at the bottom.

Gambar 4. 17 Halaman Tambah Transaksi (Pengeluaran)



Gambar 4. 18 Halaman Tambah Transaksi (*Dropdown* pilih kategori)



Gambar 4. 19 Halaman Tambah Transaksi (pilih tanggal)



Gambar 4. 20 Halaman Tambah Transaksi (Keterangan)

4.3.7 Halaman Rekap

Halaman Rekap Bulanan ini menampilkan rekap pemasukan dan pengeluaran dalam satu bulan.

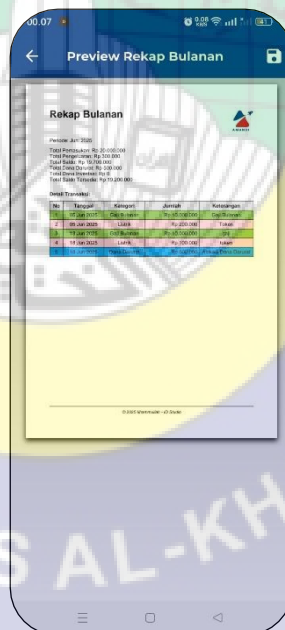
1. Fitur : Tombol pilihan bulanan dan tahunan, tombol preview PDF.



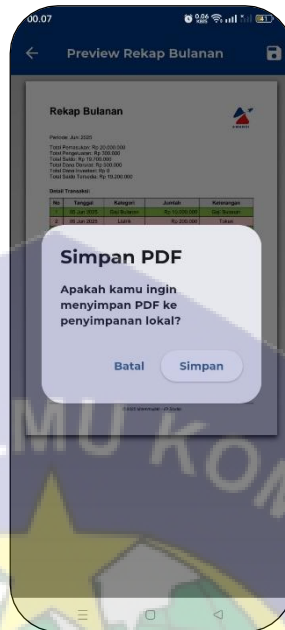
Gambar 4. 21 Halaman Rekap Bulanan



Gambar 4. 22 Halaman Rekap Bulanan (*Dropdown* bulan)



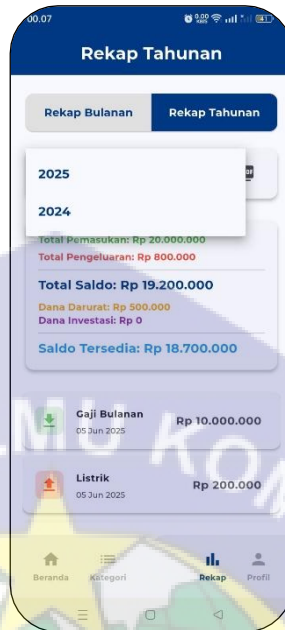
Gambar 4. 23 Halaman Rekap Bulanan (*Review* PDF)



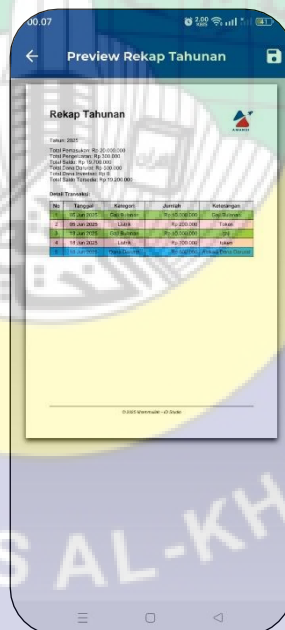
Gambar 4. 24 Halaman Rekap Bulanan (Simpan PDF ke Perangkat)



Gambar 4. 25 Halaman Rekap Tahunan



Gambar 4. 26 Halaman Rekap Tahunan (*Dropdown tahun*)

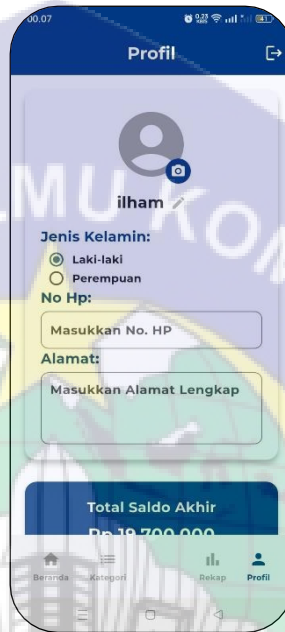


Gambar 4. 27 Halaman Rekap Tahunan (*Review PDF*)

4.3.8 Halaman Profil

Halaman Profil menampilkan foto profil, *user* yang sedang *login*, total saldo yang dimiliki *user* dan tombol *logout*.

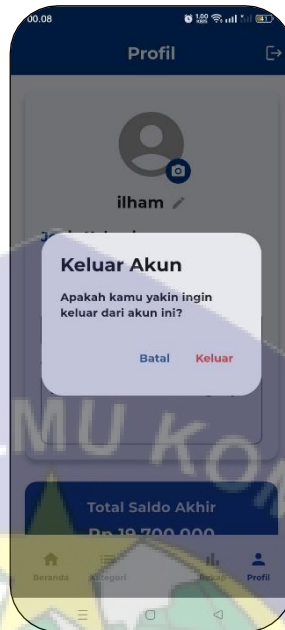
1. Fitur: Menambahkan foto profil, ubah *username* dan tombol *logout*.



Gambar 4. 28 Halaman Profil



Gambar 4. 29 Halaman Profil (*Upload Foto*)



Gambar 4. 30 Halaman Profil (*Logout*)

4.4 Hasil Pengujian (Testing)

Berdasarkan rencana pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut :

4.4.1 Metode Pengujian

Pengujian dilakukan menggunakan metode *Blackbox Testing*, yaitu metode pengujian yang hanya berfokus pada input dan output aplikasi, tanpa mengetahui struktur kode program di dalamnya.

4.4.2 Hasil Pengujian

Berikut adalah hasil pengujian menggunakan metode *Blackbox Testing*, terhadap fitur-fitur utama dalam aplikasi AMANDI :

Tabel 4. 4 Pengujian fitur Utama

No	Fitur	Input yang diberikan	Output yang diharapkan	Hasil uji	Keterangan
1	Login	Username dan password valid	Berhasil masuk ke halaman utama	Berhasil	Sesuai

2	Registrasi	Username baru dan password valid	Akun berhasil dibuat dan login otomatis	Berhasil	Validasi bekerja
3	Tambah Transaksi	Jumlah = 0	Muncul error "Jumlah tidak boleh 0"	Berhasil	Validasi tampil
4	Tambah Kategori	Input nama kategori kosong	Muncul pesan error "Kategori tidak boleh kosong"	Berhasil	Validasi tampil
5	Preview PDF Rekap	Klik tombol PDF	Halaman preview PDF muncul	Berhasil	Navigasi sesuai
6	Ubah Foto Profil	Masukkan Foto	Foto berhasil diubah dan tampil di profil	Berhasil	Foto tampil
7	Logout	Klik tombol Logout pada halaman Profil	Muncul pesan konfirmasi Logout. Jika memilih "batal" maka akan kembali ke halaman Profil dan jika memilih "keluar" maka akan ke halaman Login	Berhasil	Sesuai

4.4.3 Hasil Pengujian Detail

Berikut adalah hasil pengujian menggunakan metode *Blackbox Testing*, terhadap halaman-halaman dalam aplikasi AMANDI :

1. Halaman Login

Tabel 4. 5 Skenario Pengujian pada Halaman Login

No	Skenario Pengujian	Input yang diberikan	Output yang diharapkan	Hasil uji	Keterangan
1	Username tidak diisi, Password tidak diisi	Username (kosong) Password (kosong)	Username tidak boleh kosong, Password tidak boleh kosong	Username tidak boleh kosong, Password tidak boleh kosong	Sesuai
2	Username diisi,	Username (ada)	Password tidak boleh kosong	Password tidak boleh kosong	Sesuai

	Password tidak diisi	Password (kosong)			
3	Username tidak diisi, Password diisi	Username (kosong) Password (ada)	Username tidak boleh kosong	Username tidak boleh kosong	Sesuai
4	Username diisi, Password diisi	Username (ada-salah) Password (ada-salah)	Login Gagal Username atau Password salah!	Login Gagal Username atau Password salah!	Sesuai
5	Username diisi, Password diisi	Username (ada-benar) Password (ada-salah)	Login Gagal Username atau Password salah!	Login Gagal Username atau Password salah!	Sesuai
6	Username diisi, Password diisi	Username (ada-salah) Password (ada-benar)	Login Gagal Username atau Password salah!	Login Gagal Username atau Password salah!	Sesuai
7	Username diisi, Password diisi	Username (ada-benar) Password (ada-benar)	Login Berhasil Selamat datang "username"!	Login Berhasil Selamat datang "username"!	Sesuai

2. Halaman Registrasi

Tabel 4. 6 Skenario Pengujian pada Halaman Registrasi

No	Skenario Pengujian	Input yang diberikan	Output yang diharapkan	Hasil uji	Keterangan
1	Username tidak diisi, Password tidak diisi, Konfirmasi Password tidak diisi	Username (kosong) Password (kosong) Konfirmasi Password (kosong)	Username tidak boleh kosong, Password tidak boleh kosong, Konfirmasi Password tidak boleh kosong	Username tidak boleh kosong, Password tidak boleh kosong, Konfirmasi Password tidak boleh kosong	Sesuai
2	Username diisi, Password tidak diisi, Konfirmasi Password tidak diisi	Username (ada-kurang dari 3 karakter)) Password (kosong) Konfirmasi Password (kosong)	Username minimal 3 karakter, Password tidak boleh kosong, Konfirmasi Password tidak boleh kosong	Username minimal 3 karakter, Password tidak boleh kosong, Konfirmasi Password tidak boleh kosong	Sesuai

3	Username diisi, Password tidak diisi, Konfirmasi Password tidak diisi	Username (ada-3 karakter) Password (kosong) Konfirmasi Password (kosong)	Password tidak boleh kosong, Konfirmasi Password tidak boleh kosong	Password tidak boleh kosong, Konfirmasi Password tidak boleh kosong	Sesuai
4	Username diisi, Password diisi, Konfirmasi Password tidak diisi	Username (ada-3 karakter) Password (ada-kurang dari 6 karakter) Konfirmasi Password (kosong)	Password minimal 6 karakter, Konfirmasi Password tidak boleh kosong	Password minimal 6 karakter, Konfirmasi Password tidak boleh kosong	Sesuai
5	Username diisi, Password diisi, Konfirmasi Password tidak diisi	Username (ada-3 karakter) Password (ada- 6 karakter) Konfirmasi Password (kosong)	Konfirmasi Password tidak boleh kosong	Konfirmasi Password tidak boleh kosong	Sesuai
6	Username diisi, Password diisi, Konfirmasi Password diisi	Username (ada-3 karakter) Password (ada- 6 karakter) Konfirmasi Password (ada-salah)	Konfirmasi Password tidak cocok	Konfirmasi Password tidak cocok	Sesuai
7	Username diisi, Password diisi, Konfirmasi Password diisi	Username (ada-3 karakter) Password (ada- 6 karakter) Konfirmasi Password (ada-benar)	Akun berhasil dibuat! Silahkan Login (Navigasi ke halaman Login)	Akun berhasil dibuat! Silahkan Login (Navigasi ke halaman Login)	Sesuai

3. Halaman Utama

Tabel 4. 7 Skenario Pengujian pada Halaman Utama

No	Skenario Pengujian	Input yang diberikan	Output yang diharapkan	Hasil uji	Keterangan
1	Menu Kategori	Klik Menu Kategori	Membuka Halaman Kategori	Membuka Halaman Kategori	Sesuai
2	Tambah Transaksi	Klik Tombol Tambah Transaksi	Membuka Halaman Tambah Kategori	Membuka Halaman Tambah Kategori	Sesuai
3	Menu Rekap Bulanan	Klik Menu Rekap Bulanan	Membuka Halaman Rekap Bulanan	Membuka Halaman Rekap Bulanan	Sesuai
4	Menu Profil	Klik Menu Profil	Membuka Halaman Profil	Membuka Halaman Profil	Sesuai
5	Appbar	Klik Appbar	Membuka Kalender Appbar	Membuka Kalender Appbar	Sesuai
6	Tombol Kembali	Klik Tombol Kembali	Muncul pesan konfirmasi Logout. Jika memilih “batal” maka akan kembali ke halaman Profil dan jika memilih “keluar” maka akan ke halaman Login	Muncul pesan konfirmasi Logout. Jika memilih “batal” maka akan kembali ke halaman Profil dan jika memilih “keluar” maka akan ke halaman Login	Sesuai
7	Edit Transaksi	Klik ikon Edit Transaksi	Membuka Halaman Edit Transaksi	Membuka Halaman Edit Transaksi	Sesuai
8	Hapus Transaksi	Klik ikon Hapus Transaksi	Muncul pesan konfirmasi Hapus. Jika memilih “batal” maka akan kembali ke halaman Utama dan jika memilih “hapus” maka transaksi akan terhapus	Muncul pesan konfirmasi Hapus. Jika memilih “batal” maka akan kembali ke halaman Utama dan jika memilih “hapus” maka transaksi akan terhapus	Sesuai

4. Halaman Kategori

Tabel 4. 8 Skenario Pengujian pada Halaman Kategori

No	Skenario Pengujian	Input yang diberikan	Output yang diharapkan	Hasil uji	Keterangan
1	Switch	Klik Switch	Berubah sesuai yang dipilih (Pemasukan/ Pengeluaran)	Berubah sesuai yang dipilih (Pemasukan/ Pengeluaran)	Sesuai
2	Tambah Kategori	Klik ikon tambah Kategori	Menampilkan popup tambah Kategori	Menampilkan popup tambah Kategori	Sesuai
3	Edit Kategori	Klik ikon Edit Kategori	Menampilkan popup edit Kategori	Menampilkan popup edit Kategori	Sesuai
4	Hapus Kategori	Klik ikon Hapus Kategori	Muncul pesan konfirmasi Hapus. Jika memilih “batal” maka akan kembali ke halaman Kategori dan jika memilih “hapus” maka kategori akan terhapus	Muncul pesan konfirmasi Hapus. Jika memilih “batal” maka akan kembali ke halaman Kategori dan jika memilih “hapus” maka kategori akan terhapus	Sesuai

5. Halaman Tambah Transaksi

Tabel 4. 9 Skenario Pengujian pada Halaman Tambah Transaksi

No	Skenario Pengujian	Input yang diberikan	Output yang diharapkan	Hasil uji	Keterangan
1	Switch	Klik Switch	Berubah sesuai yang dipilih (Pemasukan/ Pengeluaran)	Berubah sesuai yang dipilih (Pemasukan/ Pengeluaran)	Sesuai
2	Jumlah tidak diisi, Tanggal tidak diisi, Keterangan tidak diisi	Jumlah (kosong) Tanggal (kosong) Keterangan (kosong)	Jumlah tidak boleh kosong, Tanggal tidak boleh kosong, Keterangan tidak boleh kosong	Jumlah tidak boleh kosong, Tanggal tidak boleh kosong, Keterangan tidak boleh kosong	Sesuai
3	Jumlah diisi, Tanggal tidak diisi,	Jumlah (ada-0)	Jumlah tidak boleh 0, Tanggal tidak boleh	Jumlah tidak boleh 0, Tanggal tidak	Sesuai

	Keterangan tidak diisi	Tanggal (kosong) Keterangan (kosong)	kosong, Keterangan tidak boleh kosong	boleh kosong, Keterangan tidak boleh kosong	
4	Jumlah diisi, Tanggal diisi, Keterangan tidak diisi	Jumlah (ada-lebih besar dari 0) Tanggal (ada) Keterangan (kosong)	Keterangan tidak boleh kosong	Keterangan tidak boleh kosong	Sesuai
	Jumlah diisi, Tanggal diisi, Keterangan diisi	Jumlah (ada-lebih besar dari 0) Tanggal (ada) Keterangan (ada)	Transaksi berhasil ditambahkan. Navigasi ke halaman Utama	Transaksi berhasil ditambahkan. Navigasi ke halaman Utama	Sesuai

6. Halaman Rekap

Tabel 4. 10 Skenario Pengujian pada Halaman Rekap

No	Skenario Pengujian	Input yang diberikan	Output yang diharapkan	Hasil uji	Keterangan
1	<i>Switch</i>	Klik rekap bulanan/tahunan	Menampilkan transaksi bulanan/tahunan	Menampilkan transaksi bulanan/tahunan	Sesuai
2	<i>Dropdown</i>	Klik Drop down	Menampilkan pilihan bulan	Menampilkan pilihan bulan	Sesuai
3	PDF	Klik ikon PDF	Menampilkan Review Rekap Bulanan	Menampilkan Review Rekap Bulanan	Sesuai
4	Ikon simpan	Klik ikon Simpan	Menampilkan dialog konfirmasi simpan PDF	Menampilkan dialog konfirmasi simpan PDF	Sesuai

7. Halaman Profil

Tabel 4. 11 Skenario Pengujian pada Halaman Profil

No	Skenario Pengujian	Input yang diberikan	Output yang diharapkan	Hasil uji	Keterangan
1	Foto Profil	Klik ikon Kamera	Membuka galeri dan mengupload foto	Membuka galeri dan mengupload foto	Sesuai
2	Logout	Klik ikon Logout	Muncul pesan konfirmasi Logout. Jika memilih “batal” maka akan kembali ke halaman Profil dan jika memilih “keluar” maka akan ke halaman Login	Muncul pesan konfirmasi Logout. Jika memilih “batal” maka akan kembali ke halaman Profil dan jika memilih “keluar” maka akan ke halaman Login	Sesuai

4.4.4 Kesimpulan Dari Hasil Pengujian

Berdasarkan hasil pengujian menggunakan metode *Black Box Testing* terhadap fitur-fitur aplikasi AMANDI, dapat disimpulkan bahwa seluruh fungsi berjalan sesuai dengan kebutuhan fungsional sistem. Setiap fitur yang diuji memberikan output yang sesuai terhadap input yang diberikan. Hal ini menunjukkan bahwa aplikasi telah siap digunakan oleh pengguna secara fungsional dan telah memenuhi tujuan utama yaitu membantu pencatatan serta pengelolaan keuangan pribadi pengguna.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan dan implementasi yang telah dilakukan pada aplikasi AMANDI (Aplikasi Manajemen Keuangan Pribadi Berbasis Mobile Menggunakan Framework Flutter), maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi AMANDI berhasil dibangun dengan menggunakan *framework Flutter* dan dapat berjalan pada *platform mobile (Android)*.
2. Aplikasi ini memiliki fitur utama seperti login, register, tambah transaksi (pemasukan dan pengeluaran), manajemen kategori, rekap bulanan, serta pengelolaan profil pengguna.
3. Fitur *preview* dan simpan PDF rekap bulanan berhasil diimplementasikan, sehingga memudahkan pengguna dalam melihat dan menyimpan laporan keuangan.
4. Pengujian menggunakan metode *Blackbox Testing* menunjukkan bahwa seluruh fitur berjalan sesuai dengan kebutuhan fungsional.

5.2 Saran

Adapun saran yang dapat diberikan untuk pengembangan aplikasi AMANDI di masa mendatang adalah sebagai berikut:

1. Menambahkan fitur grafik atau diagram untuk visualisasi data transaksi pengguna.
2. Menyediakan sistem notifikasi atau pengingat untuk mencatat transaksi secara berkala.
3. Mengembangkan versi iOS agar aplikasi dapat digunakan oleh lebih banyak pengguna.
4. Menyediakan fitur cadangan (backup) dan pemulihan (restore) data lokal.

DAFTAR PUSTAKA

- AWS. (2024). *Apa itu Flutter?* Retrieved from AWS: <https://aws.amazon.com/id/what-is/flutter/>
- Cakrawala, U. (2024, June 8). *Mengenal Apa Itu Sistem Operasi Android*. Retrieved from Universitas Cakrawala: <https://www.cakrawala.ac.id/berita/apa-itu-sistem-operasi-android>
- Fau, A. (2024). Pelatihan Pengenalan Dasar Framework Flutter dalam Pembangunan Aplikasi Mobile. *Vol. 01 No. 01 Januari 2024* , 23-28.
- Harahap, M. S., Andrianto, R., Dermilan, Elok, D., Khoiriah, C., & Andika, B. (2023, September). Pencatat Keuangan: Pengembangan Aplikasi Futter untuk Mencatat. *Jurnal Penelitian Teknologi Informasi Dan Sains*, 64-73.
- Juhardi, U., & Khairullah. (2020). Sistem Pencatatan dan Pengolahan Keuangan Pada Aplikasi Manajemen Keuangan E-Dompet Berbasis Android. *JTIS, Volume 2 Nomor 1, Februari 2020*, 24-29.
- Pramana, H. W. (2022, January 1). *Pengertian Aplikasi: Arti, Fungsi, Klasifikasi, dan Contoh Aplikasi*. Retrieved from appDesa: <https://appdesa.com>
- Sera, R. C. (2021). RANCANGAN BANGUN APLIKASI MONEY CHANGER. *Jurnal Universitas Palangka Raya*.
- Sudiantini, D., Suryadinata, A., Rahayu, A. S., Aprilia, A. B., & Lestari , A. D. (2023). Ruang Lingkup Manajemen KeuanganScope Of Financial Management. *Jurnal Kajian dan Penelitian Umum*, 60-65.
- Teknik Informatika, & Manajemen Informatika. (2024). *PEDOMAN SKRIPSI/TA*. CILEGON: Fakultas Ilmu Komputer Universitas Al-Khairiyah.
- Tjandra, S., Dewi, G. L., Santoso, S. P., & Prajetno, J. (2022). Pengembangan Aplikasi Pengaturan Keuangan Pribadi Berbasis. *Vol. 25 No.2, Hal.39-47 (2022)*, 39-47.
- Trivaika, E., & Senubekti, M. A. (2022). Perancangan Aplikasi Pengelola Keuangan Pribadi Berbasis Android. *Volume 16 Nomor 1, Januari 2022* , 33-40.
- Wikipedia. (2025, march 14). *Figma*. Retrieved from Wikipedia: <https://es.wikipedia.org/wiki/Figma>
- Wikipedia. (2025, May 24). *Android Studio*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Android_Studio

Lampiran 1 Halaman Biografi**HALAMAN BIOGRAFI****Data Pribadi :**

NIM : 21040039
Nama : ILHAMMULLAH
Tempat, Tanggal Lahir : Cilegon, 21 Seotember 1997
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Jl. Teunku Cik Ditiro Link. Pengampelan RT/RW
003/003 Kel. Kedaleman Kec. Cibeber Kota
Cilegon – Banten (42422)
Status : Belum Kawin
Email : ilhammullah45@gmail.com

**Riwayat Pendidikan :**

SD : SDN Kedaleman III (2004 - 2010)
SMP : SMP Islam Ar-Rochmah (2010 - 2013)
SMK : SMK N 1 Cilegon (2013 - 2016)
Perguruan Tinggi : Universitas Al-Khairiyah. Fakultas Ilmu Komputer.
Teknik Informatika (2021 - 2025)

Demikian Riwayat Hidup ini dibuat dengan sebenar-benarnya.

Cilegon, 18 Juni 2025

Ilhammullah

Lampiran 2 Nota Dinas

NOTA DINAS



Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi
Universitas Al Khairiyah
Fakultas Ilmu Komputer

Alamat : Jl.K.H.Enggus Arja No.1 Citangkil Kota Cilegon, Banten 42441
 No. telepon: (0254) 7877057 Website : unival-cilegon.ac.id

NOTA DINAS DOSEN PEMBIMBING LAPORAN SKRIPSI

NO SKRIPSI : SR-063
NIM : 21040039
Nama : ILHAMMULLAH
Program Studi : Teknik Informatika

Dalam rangka kelancaran proses bimbingan Skripsi, maka kami menunjuk Bapak/Ibu :

Tubagus Mochamad Isnaeni, S.Kom., M.TI Sebagai Pembimbing 1

Ainin Najmi., M.Kom Sebagai Pembimbing 2

Kepada Mahasiswa/Mahasiswi tersebut diatas.

Bimbingan dimulai sejak Nota Dinas ini dikeluarkan dan diselesaikan sebelum tanggal :

28 Juni 2025

Sebelum dan saat bimbingan dimulai Pembimbing memperhatikan hal-hal berikut :

1. Memperhatikan dan memberikan Petunjuk Sesuai Buku Pedoman **Laporan Skripsi** kepada Mahasiswa selama bimbingan.
2. Mengingatkan Mahasiswa untuk memanfaatkan waktu sebelum **batas waktu berakhir**
3. Jika sampai tanggal berakhirnya Nota Dinas ini Mahasiswa tidak dapat menyelesaikan Laporan Skripsi, maka Mahasiswa dianggap **GAGAL** dan diwajibkan untuk menyelesaikan biaya administrasi kembali.

Demikian Nota Dinas ini dibuat untuk dilaksanakan, atas perhatian dan kerjasamanya yang baik kami ucapkan terimakasih.

Cilegon, 6 Maret 2025

Ketua Program Studi Teknik Informatika

Didda Rahayu Yuliana., M.Kom
NIDN. 0426078601

Lampiran 3 Form Bimbingan

FORM BIMBINGAN



Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi
Universitas Al Khairiyah
Fakultas Ilmu Komputer
 Alamat : Jl.K.H.Enggus Arja No.1 Citangkil Kota Cilegon, Banten 42441
 No. telepon: (0254) 7877057 Website : unival-cilegon.ac.id

FORM BIMBINGAN SKRIPSI / TUGAS AKHIR

Nama : ILHAMMULLAH
 NPM : 21040039
 Fakultas / Program Studi : Teknik Informatika
 Dosen Pembimbing I : Tubagus Mochamad Isnaeni, S.Kom., M.TI
 Judul Skripsi / TA : Aplikasi Manajemen Keuangan Pribadi Berbasis Mobile Menggunakan Framework Flutter

No	Hari / Tanggal	Pembahasan	Status	Paraf
1	Kamis, 17-10-2024	Referensi Penelitian Relevan	Revisi	
2	Kamis, 24-10-2024	Perbaiki Latar Belakang	Revisi	
3	Selasa, 04-11-2024	Bab I & II	Revisi	
4	Kamis, 14-11-2024	Flowchart	Revisi	
5	Minggu, 29-12-2024	Bab I	OK	
6	Rabu, 08-01-2025	Bab II	Revisi	
7	Senin, 13-01-2025	Bab II	OK	
8	Kamis, 30-01-2025	Bab III	Revisi	
9	Rabu, 05-02-2025	Bab III	OK	
10	Senin, 05-05-2025	Bab IV	Revisi	
11	Selasa, 06-05-2025	BAB IV	OK	
12	Selasa, 03-06-2025	BAB V	Revisi	
13	Kamis, 05-06-2025	BAB V	OK	
14				
15				
16				

Cilegon, 05 Juni 2024/2025
 Pembimbing I

 Tubagus Mochamad Isnaeni, S.Kom.,

Jl. H. Enggus Arja No. 01 Citangkil Kota Cilegon Provinsi Banten
<https://www.unival.ac.id> Telp: 081 806 443 234 & 087 878 690 668
 @Universitas Al-Khairiyah Universitas Al-Khairiyah



Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi
Universitas Al Khairiyah
Fakultas Ilmu Komputer

Alamat : Jl.K.H.Enggus Arja No.1 Citangkil Kota Cilegon, Banten 42441
 No. telepon: (0254) 7877057 Website : unival-cilegon.ac.id

M.TI

FORM BIMBINGAN SKRIPSI / TUGAS AKHIR

Nama : ILHAMMULLAH
 NPM : 21040039
 Fakultas / Program Studi : Teknik Informatika
 Dosen Pembimbing II : Ainin Najmi, M.Kom
 Judul Skripsi / TA : Aplikasi Manajemen Keuangan Pribadi Berbasis Mobile Menggunakan Framework Flutter

No	Hari / Tanggal	Pembahasan	Status	Paraf
1	Rabu / 13 / 11 ²⁴	Kata Pengantar, penulisan	Revisi	
2	Rabu / 13 / 11 ²⁵	Bab I & Bab II	Revisi	
3	Senin / 13 / 11 ²⁵	Bab I & Bab II	OK	
4	Kamis / 30 / 11 ²⁵	Bab III	Revisi	
5	Rabu / 5 / 12 ²⁵	Bab III	OK	
6	Senin / 5 / 6 ²⁵	Bab IV & Bab V	Revisi	
7	Selasa / 3 / 6 ²⁵	Bab V	Revisi	
8	Kamis / 5 / 6 ²⁵	Bab IV & Bab V	OK	
9				
10				
11				
12				
13				
14				
15				
16				

Cilegon, 05 Juni 2024 2025.
 Pembimbing II

Ainin Najmi, M.Kom

Lampiran 4 Form Revisi Penguji

FORM REVISI PENGUJI

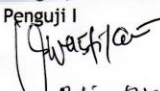


Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi
Universitas Al Khairiyah
Fakultas Ilmu Komputer
 Alamat : Jl.K.H.Enggus Arja No.1 Citangkil Kota Cilegon, Banten 42441
 No. telepon: (0254) 7877057 Website : unival-cilegon.ac.id

FORM REVISI SIDANG AKHIR SKRIPSI/TA

NPM : 21040039
 Nama Mahasiswa : Ilhammullah
 Judul SKRIPSI : Aplikasi Manajemen Keuangan Pribadi Berbasis Mobile Menggunakan Framework Flutter

No	Revisi	Halaman revisi
	Sevorka dengan Catatan & laporan.	

Cilegon, 12 Juni 2025
 Penguji I

 Rulin Waska.

Jl. H. Enggus Arja No. 01 Citangkil Kota Cilegon, Provinsi Banten
<https://www.unival.ac.id> Telp: 081 806 443 234 & 087 878 690 668
 @Universitas Al-Khairiyah  Universitas Al-Khairiyah



Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi
Universitas Al Khairiyah
Fakultas Ilmu Komputer
 Alamat : Jl.K.H.Enggus Arja No.1 Citangkil Kota Cilegon, Banten 42441
 No. telepon: (0254) 7877057 Website : unival-cilegon.ac.id

FORM REVISI SIDANG AKHIR SKRIPSI/TA

NPM : 210900309
 Nama Mahasiswa : Ihammullah
 Judul SKRIPSI : Aplikasi Manajemen keuangan pribadi berbasis Mobile menggunakan framework flutter

No	Revisi	Halaman revisi
1	form profil dilengkapi ✓	
2	bersih foto profil ✓	
3	page interest ✓	
4	data darurat ✓	
5	Pelaporan dikeas logo ✓	
6	buat kategori pemasukan & pengeluaran ✓	

Cilegon, 12 Juni 2025
 Penguji II

[Signature]
 Teuku Fadzry Shindar

Lampiran 5 Daftar Hadir

DAFTAR HADIR



Kementerian Pendidikan Tinggi, Sains, dan Teknologi
Universitas Al Khairiyah
Fakultas Ilmu Komputer
 Alamat : Jl.K.H.Enggus Arja No.1 Citangkil Kota Cilegon, Banten 42441
 No. telepon: (0254) 7877057 Website : unival-cilegon.ac.id

**DAFTAR HADIR SIDANG
 FAKULTAS ILMU KOMPUTER
 UNIVERSITAS AL - KHAIRIYAH**

No	Nama	Paraf	
1	Diky Rezaban S.	1	2
2	Giti Nugedilak		
3	Finas Yusra	3	4
4	M. Alif Wapuri		
5	Mahyudin	5	6
6	Andre Mahadie		
7		7	8
8			
9		9	10
10			
11		11	12
12			
13		13	14
14			
15		15	16
16			
17		17	18
18			
19		19	20

Lampiran 6 Berita Acara

BERITA ACARA


Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi
Universitas Al Khairiyah
Fakultas Ilmu Komputer
 Alamat : Jl.K.H.Engkus Arja No.1 Citangkil Kota Cilegon, Banten 42441
 No. telepon: (0254) 7877057 Website : unival-cilegon.ac.id

BERITA ACARA UJIAN SKRIPSI

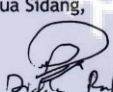
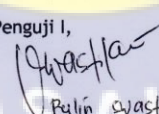
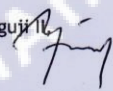
Pada hari ini Selasa tanggal 12 bulan Jan tahun 2021 Telah dilaksanakan Ujian Skripsi terhadap mahasiswa Program S1 Fakultas Ilmu Komputer (FIK) Universitas Al-Khairiyah Citangkil, sebagai berikut :

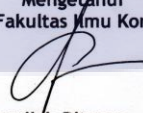
N I M : 21040039
 Nama : Ilhammullah
 Tempat/Tgl. Lahir : Cilegon, 21 September 1997
 Program Studi : Teknik Informatika
 Judul Skripsi : Aplikasi Manajemen Keuangan pribadi berbasis Mobile Menggunakan framework flutter

Dinyatakan : LULUS / ~~TIDAK LULUS~~ *) Dengan Nilai : 20
 Keterangan: *) Coret yang tidak perlu Huruf Mutu : A

Catatan : 1 minggu bimbingan

Tim Penguji Skripsi

Ketua Sidang,  <u>Dekan Fakultas Y</u> NIDN. 0426078601.	Penguji I,  <u>Pulin Swastika</u> NIDN. 0421000006 0431058401	Penguji II,  <u>Tekun Fadjar Shidek</u> NIDN. 0421068404
--	---	--

Mengetahui
 Dekan Fakultas Ilmu Komputer,

 Roy Amrullah Ritonga., M.Kom
 NIDN/ 0423038304

Jl. H. Engkus Arja No. 01 Citangkil Kota Cilegon Provinsi Banten
<https://www.unival.ac.id> Telp: 081 806 443 234 & 087 878 690 668
 @Universitas Al-Khairiyah Universitas Al-Khairiyah

Lampiran 7 Listing Program

LISTING PROGRAM

```
import 'package:calendar_appbar/calendar_appbar.dart';
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:Amandi/Authntentication/login.dart';
import 'package:Amandi/Views/category_page.dart';
import 'package:Amandi/Views/home_page.dart';
import 'package:Amandi/Views/transaction_page.dart';
import 'package:Amandi/models/database.dart';
import 'package:Amandi/Views/rekap_bulanan_page.dart';
import 'package:Amandi/Views/profil_page.dart';
import 'package:flutter/services.dart';

class Notes extends StatefulWidget {
  final String username;
  final bool showWelcome;

  const Notes({
    super.key,
    required this.username,
    this.showWelcome = false, // DEFAULT FALSE
  });

  @override
  State<Notes> createState() => _NotesState();
}

class _NotesState extends State<Notes> {
```

```

int currentIndex = 0;
final AppDb database = AppDb();
final GlobalKey<HomePageState> homePageKey =
GlobalKey<HomePageState>();

void _showWelcomeDialog() {
  showDialog(
    context: context,
    barrierDismissible: true, // <- Biar bisa dismiss kalau klik luar dialog
    builder: (context) => GestureDetector(
      behavior: HitTestBehavior.opaque,
      onTap: () {
        Navigator.of(context).pop(); // Tutup dialog saat layar disentuh
      },
      child: Dialog(
        shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(16)),
        child: Padding(
          padding: const EdgeInsets.all(20),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              const Icon(Icons.check_circle, color: Colors.green, size: 60),
              const SizedBox(height: 10),
              const Text(
                'Login Berhasil!',
                style: TextStyle(fontWeight: FontWeight.bold, fontSize: 18),
                textAlign: TextAlign.center,
              ),
            ],
          ),
        ),
      ),
    ),
  );
}

```

```

const SizedBox(height: 5),
Text(
  'Selamat datang ${widget.username}!',
  style: const TextStyle(fontSize: 14),
  textAlign: TextAlign.center,
),
],
),
),
),
),
);

Future.delayed(const Duration(seconds: 3), () {
  if (Navigator.canPop(context)) {
    Navigator.of(context).pop();
  }
});
}

void showSuccessDialog(String message) {
  showDialog(
    context: context,
    barrierDismissible: true,
    builder: (context) => GestureDetector(
      behavior: HitTestBehavior.opaque,
      onTap: () {
        if (Navigator.canPop(context)) Navigator.of(context).pop();
      },
    ),
  );
}

```



```

child: Dialog(
  shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(16)),
  child: Padding(
    padding: const EdgeInsets.all(20),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.min,
      children: [
        const Icon(Icons.check_circle, color: Colors.green, size: 60),
        const SizedBox(height: 10),
        Text(
          message,
          style: const TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 16,
          ),
          textAlign: TextAlign.center,
        ),
      ],
    ),
  ),
),
);

// Auto-close dialog after 2.5 seconds
Future.delayed(const Duration(seconds: 3), () {
  if (Navigator.canPop(context)) {
    Navigator.of(context).pop();  }  });

```

```

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

  // DIALOG SELAMAT DATANG USER
  if (widget.showWelcome) {
    WidgetsBinding.instance.addPostFrameCallback((_) =>
      _showWelcomeDialog());
  }

  selectedDate = DateTime.now();
  _children = []; // Mencegah error sebelum updateView dipanggil
  updateView(0, selectedDate);
  _checkLoginStatus();
}

Future<void> _checkLoginStatus() async {
  SharedPreferences prefs = await SharedPreferences.getInstance();
  bool isLoggedIn = prefs.getBool("isLoggedIn") ?? false;

  // Jika user belum login, arahkan ke halaman login.
  if (!isLoggedIn) {
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => LoginPage()),
    );
  }
}

```



```

void updateView(int index, DateTime? date) {
  if (currentIndex == index && date == null) return;

  setState(() {
    if (date != null) {
      selectedDate = date;
    }

    _children = [
      HomePage(selectedDate: selectedDate),
      CategoryPage(),
      RekapBulananPage(),
      ProfilPage(),
    ];
    currentIndex = index;
  });
}

@override
Widget build(BuildContext context) {
  return PopScope(
    canPop: false, // Supaya kita kontrol manual aksi back
    onPopInvoked: (didPop) async {
      if (!didPop) {
        if (currentIndex != 0) {
          updateView(0, selectedDate);
        } else {
          bool? exitApp = await showDialog(

```

```

context: context,
builder: (context) => AlertDialog(
  title: const Text('Keluar Aplikasi'),
  content: Text(
    'Apakah kamu yakin ingin keluar dari aplikasi?',
    style: TextStyle(fontWeight: FontWeight.normal),
  ),
  actions: [
    TextButton(
      onPressed: () => Navigator.of(context).pop(false),
      child: const Text('Batal'),
    ),
    TextButton(
      onPressed: () => Navigator.of(context).pop(true),
      child: const Text('Keluar', style: TextStyle(color: Colors.red)),
    ),
  ],
),
);

if (exitApp == true) {
  SystemNavigator.pop(); // Keluar dari aplikasi dengan benar
}
}
},
child: Scaffold(
  bottomNavigationBar: BottomAppBar(

```

```

shape: CircularNotchedRectangle(),
notchMargin: 8.0,
child: SizedBox(
  height: 60,
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceAround,
    children: [
      _buildNavItem(0, Icons.home, 'Beranda'),
      _buildNavItem(1, Icons.list, 'Kategori'),
      const SizedBox(width: 40),
      _buildNavItem(2, Icons.bar_chart, 'Rekap'),
      _buildNavItem(3, Icons.person, 'Profil'),
    ],
  ),
),
),
),
),

floatingActionButton: Visibility(
  visible: currentIndex == 0,
  child: FloatingActionButton(
    onPressed: () async {
      bool? result = await Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => TransactionPage(
            transactionWithCategory: null,
            onEmptyCategoryTap: () {
              updateView(1, null); // pindah ke halaman Kategori
            },
          ),
        ),
      );
    },
  ),
),
);

```

```

        if (result == true) {
            updateView(currentIndex, selectedDate);
            showSuccessDialog("Transaksi berhasil ditambahkan!");
        }
    },
    backgroundColor: Color(0xFF003884),
    shape: const CircleBorder(),
    child: const Icon(Icons.add, color: Colors.white),
),
),
floatingActionButtonLocation:
FloatingActionButtonLocation.centerDocked,
body: _children[currentIndex],
appBar: _buildAppBar(),
),
);
}

Widget _buildNavItem(int index, IconData icon, String label) {
    bool isSelected = currentIndex == index;
    Color color = isSelected ? Color(0xFF003884) : Colors.grey;

    return GestureDetector(
        onTap: () => updateView(index, null),
        child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            mainAxisAlignment: MainAxisAlignment.center,
            children: [

```

```

Icon(icon, color: color),
const SizedBox(height: 4),
Text(
  label,
  style: TextStyle(
    fontSize: 10,
    color: color,
    fontWeight: isSelected ? FontWeight.bold : FontWeight.normal,
  )),),), ); }

```

```

PreferredSizeWidget? _buildAppBar() {
  if (currentIndex == 0) {
    return CalendarAppBar(
      fullCalendar: true,
      backButton: false,
      accent: Color(0xFF003884),
      locale: 'in',
      onChanged: (date) => updateView(0, date),
      lastDate: DateTime.now(),
    );
  }
  return null;
}
}

```