

RANCANG BANGUN APLIKASI KEUANGAN DAN PENYIMPANAN DOKUMEN BERBASIS *WEB* DAN *MOBILE* MENGGUNAKAN *FLUTTER* DAN *FIREBASE* DI PT. INTI MAS MULIA

PROYEK AKHIR

Disusun Oleh:

Muhammad Fajri Hidayatullah

3312311058

Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya
Teknik Informatika



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

2026

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan tugas akhir dengan baik dan tepat waktu.

Dalam proses penyusunan laporan ini, penulis menyadari bahwa penyelesaian tugas akhir ini tidak terlepas dari dukungan, arahan, serta kontribusi dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan apresiasi dan terima kasih kepada:

1. Bapak Ir. Bambang Hendrawan, ST., MSM., CIPMP., CISCP. selaku Direktur Politeknik Negeri Batam
2. Bapak Ahmad Hamim Thohari, S.S.T., M.T. selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Batam
3. Ibu Yeni Rokhayati, S.Si., M.Sc selaku Ketua Program Studi Teknik Informatika
4. Ibu Rusyda Nazhirah Yunus S.S., M.Si. selaku dosen pembimbing yang telah memberikan arahan, bimbingan, serta masukan selama proses pengerjaan tugas akhir dan penyusunan laporan.
5. Bapak/Ibu dosen penguji yang telah memberikan saran dan evaluasi demi penyempurnaan laporan ini.
6. Nahara Tiara Dita, S.Tr.Log. yang sudah membantu memberikan arahan saran terhadap penyusunan laporan tersebut.
7. Kedua orang tua dan keluarga penulis yang senantiasa memberikan doa, dukungan, serta motivasi.

Penulis menyadari bahwa laporan tugas akhir ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun demi perbaikan di masa yang akan datang. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan menjadi referensi dalam pengembangan aplikasi dan sistem informasi.

DAFTAR ISI

KATA PENGANTAR	1
DAFTAR ISI.....	2
DAFTAR TABEL.....	4
DAFTAR GAMBAR.....	5
DAFTAR LAMPIRAN.....	6
ABSTRAK.....	7
BAB 1	8
PENDAHULUAN	8
1.1 Latar Belakang	8
1.2 Rumusan Masalah.....	10
1.3 Batasan Masalah	10
1.4 Tujuan Penelitian	10
1.5 Manfaat Penelitian	11
1.6 Pembagian Peran dan Tanggung Jawab Tim.....	12
BAB 2	13
TINJAUAN PUSTAKA	13
2.1 Tinjauan Solusi dan Sistem Terkait	13
2.2 Landasan Konsep dan Teknologi.....	18
2.3 Landasan Teori.....	19
2.4 Metode Pengembangan Produk	22
1. Metode <i>Waterfall</i>	22
2. <i>Flutter</i>	23
3. <i>Dart</i>	23
4. <i>Firebase</i>	24
BAB 3	25
ANALISIS DAN PERANCANGAN	25
3.1 Analisis Kebutuhan.....	25
3.1.1 Proses Bisnis Berjalan.....	25
3.1.2 Gambaran Umum Sistem.....	26
3.2 Perancangan	27
3.2.1 Kebutuhan Fungsional	27
3.2.2 Kebutuhan <i>Non-Fungsional</i>	28
3.2.3 Diagram <i>Use-Case</i>	28

3.2.4 Skenario <i>Use-Case</i>	29
3.2.5 <i>Activity Diagram</i>	35
3.2.6 Penjelasan Dari Hasil Penelitian.....	48
3.2.7 Perancangan Antarmuka (<i>Wireframe</i>) / Desain Perangkat Keras (<i>IoT</i>).....	49
BAB 4	57
IMPLEMENTASI DAN PENGUJIAN	57
4.1. Hasil Implementasi	57
4.2. Pengujian <i>Black-Box Testing</i>	67
4.3 Hasil Persentase Pengujian.....	69
BAB 5	69
KESIMPULAN.....	69
5.1 Kesimpulan	69
5.2 Saran	69
DAFTAR PUSTAKA.	70
LAMPIRAN.....	71
A.....	73
B.....	73
C.....	73

DAFTAR TABEL

Tabel 1. 1 Tinjauan Solusi dan Sistem Terkait.....	5
Tabel 3. 1 Kebutuhan Fungsional.....	19
Tabel 3. 2 Kebutuhan Non Fungsional.....	19
Tabel 3. 3 <i>Login</i> Aplikasi.....	20
Tabel 3. 4 Halaman <i>Dashboard</i> Utama.....	21
Tabel 3. 5 Halaman Keuangan & Menginput Keuangan.....	21
Tabel 3. 6 Halaman Dokumen.....	21
Tabel 3. 7 Laporan Keuangan.....	22
Tabel 3. 8 Edit Keuangan.....	22
Tabel 3. 9 Hapus Keuangan & Simpan.....	22
Tabel 3. 10 Melihat Dokumen.....	23
Tabel 3. 11 <i>Usecase</i> Buat Dokumen.....	23
Tabel 3. 12 Edit Dokumen.....	23
Tabel 3. 13 Hapus Dokumen.....	24
Tabel 3. 14 Halaman Profile.....	24
Tabel 3. 15 <i>Logout</i>	24
Tabel 3. 16 Akses <i>Offline</i>	24
Tabel 3. 17 <i>Generate</i> Laporan PDF.....	24
Tabel 4. 1 Pengujian Fitur <i>Register</i>	65
Tabel 4.2 Pengujian Fitur <i>Login</i>	65
Tabel 4.3 Pengujian Tambah Keuangan.....	66
Tabel 4.4 Pengujian Tambah Dokumen.....	66
Tabel 4.5 Pengujian Upload Dokumen.....	68
Tabel 4.6 Hasil Persentase Pengujian.....	69

DAFTAR GAMBAR

Gambar 2. 1 Metode <i>Waterfall</i>	
Gambar 3. 1 Alur Sebelum Menggunakan Aplikasi.....	17
Gambar 3. 2 Alur Sesudah Menggunakan Aplikasi.....	17
Gambar 3. 3 Gambaran Umum.....	18
Gambar 3. 4 Diagram <i>Use Case</i>	20
Gambar 3. 5 <i>Activity Diagram</i> Halaman <i>Login</i>	25
Gambar 3. 6 <i>Activity Diagram</i> Halaman Utama.....	26
Gambar 3. 7 <i>Activity Diagram</i> Input Keuangan.....	26
Gambar 3. 8 <i>Activity Diagram</i> Lihat Transaksi.....	27
Gambar 3. 9 <i>Activity Diagram</i> Edit Transaksi.....	28
Gambar 3. 10 <i>Activity Diagram</i> Hapus Transaksi.....	28
Gambar 3. 11 <i>Activity Diagram</i> Upload Dokumen.....	29
Gambar 3. 12 <i>Activity Diagram</i> Buat Dokumen.....	30
Gambar 3. 13 <i>Activity Diagram</i> Lihat Dokumen.....	30
Gambar 3. 14 <i>Activity Diagram</i> Edit Dokumen.....	31
Gambar 3. 15 <i>Activity Diagram</i> Hapus Dokumen.....	32
Gambar 3. 16 <i>Activity Diagram</i> Profile.....	32
Gambar 3. 17 <i>Activity Diagram</i> <i>Logout</i>	33
Gambar 3. 18 <i>UI</i> Aplikasi Halaman <i>Login</i>	34
Gambar 3. 19 <i>UI</i> Aplikasi Halaman Lupa Password.....	35
Gambar 3. 20 <i>UI</i> <i>Register</i>	35
Gambar 3. 21 <i>UI</i> <i>Dashboard</i> Utama.....	36
Gambar 3. 22 <i>UI</i> Halaman Keuangan, Tambah dan Hapus Transaksi.....	36
Gambar 3. 23 <i>UI</i> Halaman dan Tambah Dokumen.....	37
Gambar 3. 24 <i>UI</i> Profile dan <i>Logout</i>	37
Gambar 3. 25 <i>Register</i>	38
Gambar 3. 26 Aplikasi <i>Mobile</i> Halaman <i>Login</i>	39
Gambar 3. 27 Lupa <i>Password</i>	39
Gambar 3. 28 <i>Dashboard</i> Utama.....	40
Gambar 3. 29 <i>Dashboard</i> Keuangan, Transaksi Data dan Hapus Transaksi.....	40
Gambar 3. 30 <i>Dashboard</i> dan Tambah Dokumen.....	41
Gambar 3. 31 Halaman <i>Profil</i> dan <i>Logout</i>	41

DAFTAR LAMPIRAN

Lampiran A Link Proyek.....	68
Lampiran B Link Dokumen Pengujian UAT.....	68
Lampiran C Link Figma.....	68
Lampiran D Pengujian Dokumentasi.....	68

ABSTRAK

Perusahaan beralih dari sistem manual ke sistem digital sebagai akibat dari kemajuan teknologi informasi. khususnya dalam pengelolaan keuangan dan penyimpanan dokumen. PT. Inti Mas Mulia masih menghadapi berbagai kendala dalam pencatatan keuangan dan pengarsipan dokumen, seperti kesulitan dalam pencarian dokumen. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun aplikasi keuangan dan penyimpanan dokumen berbasis *web* dan *mobile* yang terintegrasi.

Analisis kebutuhan, perancangan, pengembangan, pengujian, dan implementasi adalah semua langkah dalam pengembangan sistem yang digunakan melalui metode waterfal. Aplikasi ini di kembangkan menggunakan *framework Flutter* untuk mendukung pengembangan lintas platform, serta firebase sebagai *backend* untuk penyimpanan data, autentikasi pengguna, dan pengelolaan dokumen secara real-time.

Hasil dari penelitian ini adalah sebuah aplikasi yang mampu memfasilitasi pencatatan transaksi keuangan, pengelolaan dokumen digital, serta pembuatan laporan dalam format PDF. Aplikasi ini juga mendukung akses secara *online* dan *offline* serta memiliki fitur sinkronisasi data secara otomatis. Berdasarkan hasil pengujian menggunakan metode *Black-Box Testing*, aplikasi dapat berjalan sesuai dengan fungsi yang diharapkan.

Kesimpulan dari penelitian ini adalah bahwa aplikasi yang dikembangkan mampu meningkatkan efisiensi, keakuratan, dan keamanan dalam pengelolaan keuangan serta penyimpanan dokumen di PT. Inti Mas Mulia, Dengan demikian, aplikasi ini dapat menjadi Solusi yang efektif mendukung operasional perusahaan di era digital.

Kata Kunci: Aplikasi keuangan, penyimpanan dokumen, *Flutter*, *Firestore*, *web*, *mobile*, dan *Black-Box*

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang sangat pesat telah membawa perubahan besar dalam cara organisasi dan individu perusahaan mengelola data. Salah satu aspek penting adalah pencatatan transaksi keuangan dan penyimpanan dokumen. Pencatatan keuangan yang masih dilakukan secara manual sering terjadi kesalahan, seperti kesalahan perhitungan, keterlambatan pelaporan, hingga sulit melakukan evaluasi keuangan. Sementara itu, penyimpanan dokumen secara fisik juga sering menghadapi kendala, seperti resiko kerusakan, kehilangan, serta keterbatasan ruang arsip, kedua permasalahan tersebut dapat menghambat efektivitas kerja serta menurunkan kualitas pengambilan keputusan dalam organisasi maupun usaha (Anjilna, 2025). Untuk menyelesaikan masalah tersebut dibutuhkan sebuah sistem yang dapat mengintegrasikan pencatatan keuangan dan penyimpanan dokumen secara digital. Aplikasi keuangan dan penyimpanan dokumen hadir sebagai solusi modern yang memungkinkan pengguna untuk mencatat transaksi keuangan secara cepat, akurat, dan rapi, sekaligus menyimpan dokumen penting dalam bentuk digital yang lebih aman serta mudah diakses. Dengan adanya sistem ini pencatatan keuangan dan penyimpanan dokumen dapat berjalan lebih efisien serta resiko kehilangan data dapat diminimalisasi. Hal ini sejalan dengan hasil penelitian sebelumnya yang menekankan pentingnya digitalisasi arsip dalam meningkatkan efektivitas kerja organisasi (Sudarsana, 2024).

PT. Inti Mas Mulia salah satu perusahaan yang sedang mengembangkan teknologi digital juga menghadapi tantangan tersebut. Seiring meningkatnya aktivitas operasional bisnis, perusahaan membutuhkan sistem pencatatan keuangan dan penyimpanan dokumen yang lebih aman, terorganisir, serta mudah diakses kapan saja dan di mana saja. Dengan adanya sistem berbasis *web* dan *mobile*, perusahaan dapat meningkatkan efisiensi dalam pencatatan transaksi keuangan sekaligus mempermudah pengelolaan dokumen secara digital.

Suatu perusahaan dapat bertahan dan terus berkembang apabila mampu mencermati kondisi serta kinerja perusahaan secara berkelanjutan. Untuk mengetahui kondisi dan kinerja tersebut, diperlukan analisis yang tepat, salah satunya melalui pencatatan keuangan dan pengelolaan arsip dokumen. Pencatatan keuangan memberikan gambaran mengenai hasil serta

perkembangan usaha perusahaan. Pada awalnya, pencatatan keuangan hanya berfokus pada catatan transaksi masuk dan keluar, serta laporan keuangan sederhana. Namun, seiring dengan semakin kompleksnya kegiatan operasional, pencatatan keuangan yang akurat dan terstruktur menjadi sangat penting.

Selain itu, penyimpanan dokumen juga merupakan aspek vital dalam administrasi perusahaan. Dokumen berfungsi sebagai sumber informasi, rekam jejak, pusat ingatan, alat pengambilan keputusan, dan bukti organisasi. Oleh karena itu, penyimpanan dokumen dengan baik dan terorganisir sangat diperlukan untuk mendukung efektivitas dan efisiensi kerja. Di era perkembangan teknologi saat ini, kebutuhan akan informasi semakin penting. Hampir seluruh perusahaan memanfaatkan sistem informasi untuk menunjang operasionalnya, termasuk dalam hal pencatatan keuangan dan penyimpanan dokumen.

Berdasarkan masalah tersebut, perusahaan membutuhkan sebuah aplikasi yang mampu mencatat keuangan sekaligus menyimpan dokumen perusahaan secara lebih aman dalam satu platform terintegrasi. Aplikasi ini dirancang menggunakan *Flutter*, yang memungkinkan pengembangan untuk *web* dan *mobile* hanya dengan satu basis kode, sehingga lebih efisien dalam hal waktu maupun biaya pengembangan. Sementara itu, *Firebase* digunakan sebagai *backend* karena memiliki berbagai keunggulan, antara lain penyimpanan data berbasis *cloud*, tingkat keamanan yang terjamin, serta kemampuan untuk membaca dan memperbarui data secara *real-time*.

Rancang bangun aplikasi keuangan dan penyimpanan dokumen ini diharapkan dapat membantu perusahaan PT. Inti Mas Mulia untuk meningkatkan kinerja perusahaan, serta menghilangkan resiko kesalahan dalam pencatatan keuangan, memberikan kemudahan bagi karyawan untuk mengakses data keuangan dan arsip dokumen perusahaan. Dengan adanya inovasi ini, perusahaan diharapkan tidak hanya meningkatkan efisiensi internal, tetapi juga memperkuat daya saing di era digital. Aplikasi keuangan dan penyimpanan dokumen ini menjadi semakin fondasi penting bagi PT. Inti Mas Mulia dalam mewujudkan sistem kerja yang lebih profesional, transparan, dan siap menghadapi tantangan bisnis di masa depan.

1.2 Rumusan Masalah

Berdasarkan informasi ini, dapat disimpulkan rumusan masalah sebagai berikut:

1. Bagaimana merancang dan membangun aplikasi pencatatan keuangan dan penyimpanan dokumen perusahaan yang terintegrasi berbasis *web* dan *mobile*?
2. Bagaimana merancang aplikasi berbasis *web* dan *mobile* menggunakan *flutter* agar dapat diakses oleh pengguna secara lintas platform dengan tampilan dan fungsi yang lebih konsisten?
3. Bagaimana memanfaatkan *Firebase* sebagai *backend* untuk mendukung penyimpanan data, autentikasi pengguna serta keamanan dokumen perusahaan?

1.3 Batasan Masalah

1. Aplikasi yang dikembangkan difokuskan pada dua fitur utama yaitu pencatatan keuangan dan penyimpanan dokumen.
2. Platform yang akan digunakan yaitu aplikasi berbasis *web* dan *mobile* dengan *framework Flutter*.
3. *Backend* dan layanan penyimpanan hanya menggunakan *Firebase (Database dan Authentication)*
4. Penyimpanan dokumen hanya mendukung format umum seperti *PDF* dan gambar (*PNG/JPEG*)
5. Pengujian aplikasi hanya dilakukan dengan metode *Black-Box Testing*.
6. Sistem keamanan yang digunakan hanya sebatas fitur bawaan dari *Firebase* tanpa pengembangan enkripsi khusus.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan penelitian ini adalah sebagai berikut:

1. Mengembangkan aplikasi keuangan dan penyimpanan dokumen berbasis *web* dan *mobile* yang terintegrasi dan mudah digunakan di PT. Inti Mas Mulia.
2. Merancang antarmuka aplikasi dengan *Flutter* agar dapat berjalan di lintas platform (*Web* dan *Mobile*) dengan tampilan yang konsisten dan responsif.
3. Memanfaatkan *Firebase* sebagai *backend* untuk mendukung penyimpanan data, autentikasi pengguna, serta pengelolaan dokumen secara aman dan *real-time*.

1.5 Manfaat Penelitian

1. Teoritis

- a. Penelitian ini dapat membantu pengembangan ilmu pengetahuan, khusus pada bidang rekayasa perangkat lunak dan sistem informasi
- b. Hasil penelitian dapat digunakan sebagai referensi akademik bagi siswa dan peneliti berikutnya.
- c. Dapat memberikan gambaran mengenai integrasi antara pengembangan aplikasi lintas platform

2. Praktis

1. Bagi Peneliti

- a. Dapat menjadi sarana pembelajaran serta pengalaman secara praktis dalam mengembangkan aplikasi berbasis *Flutter*
- b. Dapat memberikan pemahaman lebih dalam mengenai penerapan teknologi modern

2. Bagi Perusahaan

- a. Dapat membantu perusahaan dalam mengelola keuangan serta penyimpanan dokumen agar lebih terorganisir
- b. Dapat menyediakan sistem yang dapat meningkatkan validitas dan keakuratan informasi keuangan maupun arsip dokumen
- c. Dapat memberikan manfaat langsung bagi HRD dengan menyediakan akses informasi keuangan dan arsip dokumen secara *real-time*

3. Kontribusi Terhadap Sustainable Development Goals (SDGs)

- a Mendukung pekerjaan yang layak dan pertumbuhan ekonomi melalui peningkatan efisiensi dan produktivitas dalam pengelolaan dokumen.
- b Mendukung industri, inovasi, dan infrastruktur melalui penerapan teknologi digital dalam pengembangan aplikasi.
- c Mendukung kelembagaan yang Tangguh dengan meningkatkan transparansi dan akuntabilitas dalam pengelolaan data keuangan dan dokumen.

1.6 Pembagian Peran Dan Tanggung Jawab Tim

Pelaksanaan penelitian ini tidak terlepas dari adanya kolaborasi dan sinergi antara penulis dengan berbagai pihak. Dalam proses pengembangan sistem, penulis

bekerja sama dengan pihak perusahaan, yaitu PT. Inti Mas Mulia, sebagai objek sistem yang akan dikembangkan dan serta memenuhi kebutuhan sistem. Selain itu, penulis juga mendapatkan arahan dan bimbingan dari dosen pembimbing dalam proses penyusunan skripsi, mulai dari tahap perencanaan, perancangan sistem, hingga penyusunan laporan penelitian.

Kolaborasi juga dilakukan dengan pengguna (*user*) sistem untuk mengetahui kebutuhan serta melakukan pengujian terhadap aplikasi yang dikembangkan. Dengan adanya kerja sama ini, diharapkan sistem yang dihasilkan dapat sesuai dengan kebutuhan pengguna serta memberikan manfaat yang optimal

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Solusi dan Sistem Terkait

Tinjauan solusi dan sistem terkait merupakan bagian yang membahas berbagai aplikasi atau sistem yang telah dikembangkan sebelumnya dan memiliki keterkaitan dengan topik proyek ini. Tujuan dari pembahasan ini adalah untuk mengetahui pendekatan yang telah digunakan, fitur yang disediakan, teknologi yang diterapkan, serta kelebihan dan keterbatasan dari masing – masing sistem. Dengan adanya tinjauan ini, diharapkan dapat menjadi dasar pertimbangan dalam merancang dan mengembangkan solusi yang lebih sesuai dengan kebutuhan pengguna

Tabel 1. 1 Tinjauan Aplikasi Sejenis

Tahun	Judul	Fitur Utama	Teknologi	Hasil	Kelebihan	Keterbatasan
(Pengelolaan Keuangan Berbasis Digital Pengguna Aplikasi MONEY LOVERS, n.d.)	Aplikasi MONEY LOVER	Pencatatan pemasukan, dan pengeluaran, pembuatan anggaran bulanan, dan laporan keuangan otomatis	Aplikasi berbasis <i>mobile</i> (Android/i OS) Menggunakan aplikasi digital Money Lover sebagai tools Berbasis teknologi digital / financial technology (fintech) Menggunakan perangkat: Smartphone Internet (untuk	Menunjukkan bahwa siswa memiliki antusiasme yang tinggi dan mampu memahami serta mengoperasikan aplikasi MONEY LOVERS dengan baik.	Memberikan manfaat bagi pengguna dan juga dapat menampilkan keterampilan praktis dalam mengelola keuangan pribadi.	Lingkup penelitian sangat terbatas hanya melibatkan siswa Trisakti Jaya Banda Lampung, Fokus hanya pada pelatihan, bukan pengembangan sistem, tidak membahas evaluasi sistem secara teknis.

Tahun	Judul	Fitur Utama	Teknologi	Hasil	Kelebihan	Keterbatasan
			instalasi & penggunaan aplikasi)			
(Alfonso & Setiawati, 2024)	Aplikasi Manajemen Uang Kas Kelas Berbasis <i>Mobile</i> Dengan <i>Framework Flutter</i>	Pencatatan transaksi kas, rekapitulasi pembayaran siswa, pembuatan laporan keuangan otomatis, notifikasi pengingat pembayaran melalui WhatsApp, input dan pengelolaan data siswa, monitoring siswa yang sudah dan belum membayar	<i>Framework Flutter</i> (<i>mobile</i> Android & iOS) Bahasa pemrograman <i>Dart</i> <i>Database MongoDB (NoSQL)</i> Metode pengembangan: <i>Prototype</i> Metode analisis: <i>PIECES</i> Pengujian sistem: <i>Black Box Testing</i>	Aplikasi berhasil membantu bendahara dalam mengelola kas kelas Mampu: merekap transaksi secara otomatis menghasilkan laporan keuangan mengirim pengingat pembayaran Sistem dinilai: lebih efisien mengurangi kesalahan pencatatan sesuai dengan kebutuhan pengguna	Mempermudah pencatatan keuangan (tidak manual lagi) Mengurangi human error dalam perhitungan Laporan keuangan otomatis Ada fitur notifikasi (lebih modern & interaktif) Data tersimpan secara online (lebih aman) <i>User-friendly</i> untuk bendahara	Hanya digunakan dalam lingkup kas kelas (terbatas) Belum ada fitur penyimpanan dokumen integrasi sistem lain Fitur notifikasi masih bisa dikembangkan (belum otomatis penuh) Tidak ada pembahasan keamanan data secara mendalam Pengguna terbatas pada bendahara & admin kelas
(Renaldy & Pakereng, 2023)	Aplikasi Arsip Surat <i>Mobile</i> Menggunakan <i>Framework Flutter</i> (Studi Kasus : PT. BPD Maluku, Malut)	Input / tambah data surat (surat masuk & keluar) Upload foto/gambar surat Melihat daftar arsip surat	<i>Framework Flutter</i> Bahasa pemrograman <i>Dart</i> <i>API (Application</i>	Aplikasi berhasil: mempermudah pengarsipan surat menggantikan sistem manual (buku)	Mengubah sistem manual digital (lebih efisien) Akses fleksibel (berbasis <i>mobile</i>)	Hanya memiliki 1 jenis <i>user</i> (<i>user</i> saja) Belum ada: sistem <i>login multi-user</i> manajemen hak akses

Tahun	Judul	Fitur Utama	Teknologi	Hasil	Kelebihan	Keterbatasan
		Pencarian surat (berdasarkan nomor, asal, atau perihal) Melihat detail surat Penyimpanan arsip dalam database	<i>Programming Interface</i> Sistem berbasis <i>mobile</i> (<i>Android/iOS</i>) Metode pengembangan: Waterfall Perancangan sistem: UML Pengujian sistem: <i>Black Box Testing</i>	Membantu pegawai dalam: mengelola surat masuk & keluar mengorganisir data dengan lebih rapi Sistem berjalan dengan baik dan sesuai kebutuhan pengguna	Bisa menyimpan foto surat (lebih lengkap) Pencarian data lebih cepat Tampilan menarik (<i>Flutter UI</i>) Mempermudah pekerjaan pegawai	Belum terintegrasi dengan sistem lain Fitur masih sederhana (arsip saja) Penggunaan masih terbatas pada satu instansi Disarankan pengembangan: penambahan fitur akun untuk divisi lain
(Fahmi Aziz & Sanjaya, 2023)	Aplikasi Kas Berbasis <i>Flutter</i> untuk Meningkatkan Efisiensi Pencatatan Transaksi Keuangan	<i>Login</i> pengguna (admin, bendahara, anggota) <i>Dashboard</i> informasi kas (saldo, kas masuk & keluar) Input transaksi kas (pemasukan & pengeluaran) Riwayat transaksi kas Pengelolaan data anggota & bendahara	<i>Framework Flutter (mobile)</i> Bahasa pemrograman: <i>Dart (mobile)</i> <i>Laravel (web backend)</i> Database: <i>MySQL</i> Desain <i>UI/UX</i> : Figma	Aplikasi berhasil: meningkatkan efisiensi pencatatan keuangan mengurangi kesalahan perhitungan meningkatkan transparansi & akuntabilitas Sistem: berjalan dengan baik	Mengatasi pencatatan manual yang lambat Sistem lebih efisien & terstruktur Mendukung <i>multi-user</i> (admin, bendahara, anggota) Laporan keuangan otomatis	Fitur laporan masih bisa dikembangkan lebih detail Belum ada: penyimpanan dokumen integrasi sistem lain Fokus hanya pada organisasi tertentu (IASS) Belum membahas keamanan sistem secara mendalam

Tahun	Judul	Fitur Utama	Teknologi	Hasil	Kelebihan	Keterbatasan
		Pembuatan dan export laporan kas Manajemen data wilayah & komisariat	Metode pengembangan: Waterfall Pengujian sistem: Functional Suitability Testing	sesuai kebutuhan pengguna Hasil pengujian menunjukkan: tingkat kesesuaian fungsi 100% (valid)	Transparansi keuangan meningkat Tampilan sederhana dan mudah digunakan	Bergantung pada koneksi sistem (<i>web + mobile</i>)
(Lendo & Mapaly, 2023)	Rancang Bangun Aplikasi Perpustakaan Digital Berbasis <i>Mobile</i> Menggunakan <i>Framework Flutter</i>	Manajemen konten digital (upload & kelola) Peminjaman, baca, dan pengembalian online Fitur baca PDF dalam aplikasi Pencarian katalog buku Keamanan (watermark & anti-copy) <i>Multi-role user</i> (admin, operator, anggota, tamu) Berbasis <i>mobile</i> (<i>Flutter</i>)	<i>Flutter</i> , <i>Dart</i> , <i>Node.js</i> + <i>Express.js</i> , <i>MongoDB</i> , <i>Firebase</i> , <i>Railway</i> , <i>REST API</i> , <i>Syncfusion</i> , <i>PDF</i> , <i>Viewer</i> <i>pdf-lib</i> ,	Berhasil dibuat aplikasi perpustakaan digital berbasis <i>mobile</i> Sistem dapat mengelola konten (upload, edit, hapus) Mendukung peminjaman, membaca, dan pengembalian buku secara online Pengguna tidak perlu datang ke perpustakaan Fitur keamanan berjalan (watermark & anti-copy) Aplikasi berjalan responsif menggunakan <i>Flutter</i>	Akses kapan saja & di mana saja Proses lebih cepat dan efisien Cross-platform (1 kode banyak platform) Data lebih terstruktur Keamanan konten lebih baik Tampilan <i>modern & user friendly</i>	Bergantung pada internet Belum mendukung <i>offline</i> Performa <i>backend</i> belum optimal Keamanan tidak sepenuhnya aman Konsumsi data cukup besar Bergantung pada server & teknologi tertentu

Tahun	Judul	Fitur Utama	Teknologi	Hasil	Kelebihan	Keterbatasan
				Sistem berhasil diuji dan fungsi berjalan sesuai kebutuhan (<i>blackbox testing</i>)		

Berdasarkan Tabel diatas, dapat disimpulkan bahwa aplikasi keuangan dan pengelolaan dokumen digital telah berkembang dengan fitur yang semakin lengkap dan penggunaan teknologi modern seperti *Flutter*. Sistem yang ada mampu meningkatkan efisiensi, akurasi, dan kemudahan dalam pencatatan serta pengarsipan data. Namun, masih terdapat keterbatasan seperti lingkup penggunaan yang sempit, belum adanya integrasi antara keuangan dan dokumen, serta kurangnya pembahasan keamanan data. Oleh karena itu, diperlukan pengembangan aplikasi yang lebih terintegrasi dan komprehensif sesuai kebutuhan pengguna.

2.2 Landasan Konsep dan Teknologi

1. Konsep Sistem Informasi

Dalam penelitian ini, sistem informasi digunakan untuk mengelola dokumen perusahaan serta transaksi keuangan secara terpusat, sehingga informasi yang dihasilkan lebih akurat, cepat, dan mudah diakses oleh pengguna. Istilah "sistem informasi" mengacu pada sekumpulan komponen yang saling terintegrasi yang digunakan untuk mengumpulkan, mengelola, menyimpan, dan mendistribusikan informasi untuk membantu proses pengambilan keputusan organisasi.

2. Konsep Rekayasa Perangkat Lunak

Dalam penelitian ini, rekayasa perangkat lunak digunakan untuk membuat aplikasi berkualitas tinggi yang sesuai dengan kebutuhan pengguna dan dapat diandalkan, yang mencakup tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan sistem.

3. Konsep CRUD

CRUD merupakan operasi dasar dalam pengelolaan data pada sistem informasi, yang terdiri atas Create (menambahkan data), Read (membaca data), Update (memperbaharui data), dan Delete (menghapus data). Dalam aplikasi yang dikembangkan,

konsep CRUD diterapkan pada pengelolaan transaksi keuangan maupun dokumen, sehingga pengguna dapat mengelola data secara fleksibel dan terstruktur.

4. Konsep *Database* dan *Cloud Computing*

Database adalah kumpulan data yang tersimpan secara sistematis dan dapat diakses serta dikelola dengan mudah. Dalam penelitian ini, *database* yang digunakan berbasis *cloud* melalui *Firebase*, sehingga data dapat diakses secara real - time dari berbagai perangkat. *Cloud Computing* memungkinkan penyimpanan data secara online dan tingkat keamanan, skalabilitas, dan ketersediaan yang tinggi.

5. Keamanan Data

Keamanan data merupakan aspek penting dalam sistem informasi, terutama yang berkaitan dengan data keuangan dan dokumen penting. Sistem keamanan meliputi autentikasi penggunaan, kontrol akses, serta perlindungan terhadap kehilangan dan penyalahgunaan data. Dalam penelitian ini, keamanan data didukung oleh fitur *Firebase Authentication* dan sistem penyimpanan *cloud*

2.3 Landasan Teori

1. Definisi dan Konsep Utama Inventory

Menurut (Apisca et al., 2024) Aplikasi keuangan merupakan aplikasi perangkat lunak yang dirancang untuk mencatat, mengolah dan menganalisis data transaksi keuangan yang secara akurat. Dengan adanya aplikasi keuangan tersebut setiap pemasukan dan pengeluaran dapat direkam dalam kategori tertentu, serta dapat disajikan dalam bentuk laporan hal ini bertujuan agar proses pengambilan dan keputusan yang dilakukan secara tepat dan *real-time*. Selain itu, Aplikasi keuangan terdapat pula sistem untuk menyusun, menyimpan, serta mengelola dokumen dalam bentuk digital sistem manajemen dokumen ini memungkinkan proses pengarsipan dilakukan secara elektronik, sehingga dokumen fiks dapat dikonversikan menjadi file digital. Menurut (Renaldy & Pakereng, 2023) dokumen yang telah disimpan dalam bentuk digital dapat disimpan diclode, yang sangat memudahkan pengguna untuk menemukan dokumen tertentu secara cepat dan tepat tanpa harus membuka satu persatu arsip. dengan demikian, proses ini menjadi salah satu pengelolaan dokumen menjadi lebih praktis dan nyaman.

Prinsip keamanan data menjadi salah satu prioritas utama dalam pengembangan aplikasi keuangan dan penyimpanan dokumen. Setiap akses terhadap

keuangan maupun dokumen. Mendukung pengembangan lintas platform ini digunakan *flutter* yaitu *framework open-source* dari *google* memungkinkan pembangunan aplikasi android dan *web* dalam satu basis kode. *Flutter* sangat fleksibel dalam pengembangan antarmuka penggunaan berbasis widget, mendukung produktivitas fitur fitur hot reload yang dapat mempermudah proses perbaikan dan pengujian aplikasi secara cepat.

Pada sisi *backend* aplikasi ini memanfaatkan *firebase* sebagai platform layanan berbasis *cloud*, *firebase* juga menyediakan fitur utama seperti *cloud firestore* untuk menyimpan data transaksi secara *real-time*. *Firebase authentication* untuk sistem autentikasi pengguna, *firebase storage* sebagai penyimpanan media dan pengarsipan dokumen, *cloud functions* dapat digunakan untuk menjalankan proses otomatisasi tertentu. Integrasi antara *flutter* dan *firebase* memberikan dukungan penuh terhadap pengembangan aplikasi keuangan dan penyimpanan dokumen ini, kombinasi keduanya tidak hanya menawarkan kecepatan dan pengembangan aplikasi tetapi juga menghadirkan tingkat keamanan dan *skalabilitas* serta efisien yang tinggi. Aplikasi yang dibangun mampu menjawab kebutuhan perusahaan dalam mengelola keuangan dan penyimpanan dokumen secara digital, terintegrasi, serta modern.

2. Uraian singkat mengenai studi kasus/objek penelitian

Objek penelitian ini adalah PT. Inti Mas Mulia, sebuah perusahaan yang bergerak dibidang distribusi dan perbaikan mesin-mesin kapal adapun mesin-mesin perusahaan industri. perusahaan tersebut berkembang setiap harinya dan memiliki banyak aktivitas operasional yang melibatkan pencatatan transaksi keuangan dan pengelolaan dokumen penting, aktivitas tersebut meliputi pencatatan hasil penjualan barang dan pengeluaran biaya operasional, pembayaran kepada vendor, serta pengelolaan administrasi proyek. Selain itu, perusahaan juga harus mengelola berbagai dokumen pendukung, seperti nota pembelian, faktur penjualan, kontrak kerja sama, pelaporan keuangan, hingga arsip dokumen. Pada kondisi saat ini, proses pencatatan keuangan masih dilakukan secara manual dan dilakukan menggunakan aplikasi seperti spreadsheet cara ini memang dapat membantu pencatatan dalam skala kecil, namun semakin tidak efisien ketika jumlah transaksi yang terus meningkat. Beberapa kendala yang muncul antara lain adalah resiko terjadi kesalahan pencatatan, duplikat data, dan keterlambatan dalam penyusunan laporan, serta dapat kesulitan dalam melakukan analisis data keuangan

secara cepat. Kondisi ini berdampak langsung pada proses pengambilan keputusan manajemen, yang seringkali terhambat karena data keuangan yang tidak tersedia secara akurat dan *real-time*.

Disisi lain, pengelolaan dokumen perusahaan juga masih banyak dilakukan secara konvensional. Dokumen penting seperti nota dan kontrak banyak yang masih berbentuk fisik dan disimpan dalam arsip fisik. Dokumen memang sudah berbentuk digital, tetapi tersebar di berbagai perangkat komputer karyawan tanpa adanya sistem penyimpanan terpusat. Akibatnya dokumen sering sulit ditemukan saat dibutuhkan, beresiko hilang, juga serta rawan terhadap masalah keamanan data.

Alur kerja yang berjalan di perusahaan sebagai berikut:

1. Transaksi keuangan dicatat secara manual atau di dalam file terpisah
2. Bukti transaksi dan dokumen yang pendukung disimpan dalam bentuk fisik atau dalam bentuk digital (*PDF*).
3. Penyusunan laporan keuangan memerlukan waktu yang sangat lama karena harus menggabungkan tanpa sistem pengarsipan.
4. Akses dokumen terbatasnya pada penyimpanan lokal komputer, sehingga sulit jika pemimpin perusahaan membutuhkan dokumen.

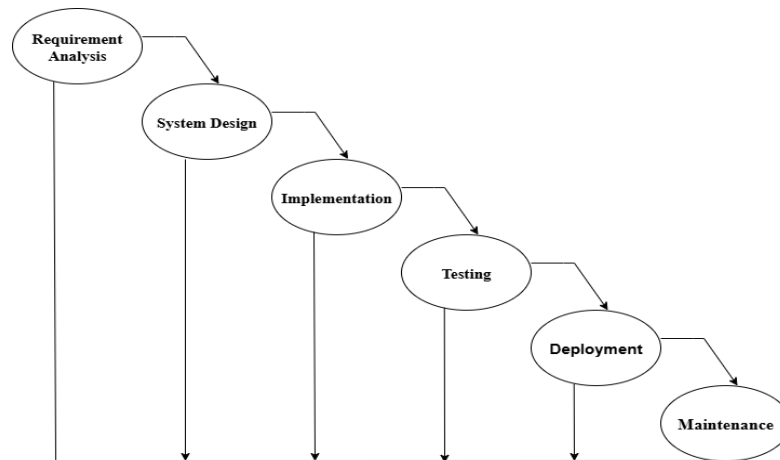
Dengan adanya kondisi tersebut, perusahaan menghadapi tantangan dalam hal efisien, akurat, dan keamanan pengelolaan data. Karena itu, dibutuhkan sebuah sistem yang mampu mengintegrasikan pencatatan transaksi keuangan dengan pengelolaan dokumen di dalam satu platform. Aplikasi yang dirancang berbasis *web* dan *mobile* menggunakan *flutter* akan memungkinkan mengakses data dari berbagai perangkat secara fleksibel. Selain itu, dukungan layanan *cloud firebase* dapat menyediakan penyimpanan secara *real-time*, autentikasi pengguna, serta penyimpanan dokumen yang sangat aman. Dengan penerapan sistem ini, diharapkan PT.Inti Mas Mulia dapat meningkatkan efektivitas operasional perusahaan dan dapat mempercepat proses laporan keuangan serta meminimalisir resiko kehilangan data dan dokumen penting.

2.4 Metode Pengembangan Produk

1. Metode *Waterfall*

Metode waterfall adalah pendekatan linier untuk pengembangan sistem yang menggambarkan tahapan pengembangan yang dilakukan secara berurutan, mulai dari tahap analisis kebutuhan hingga tahap pemeliharaan. Setiap tahap memiliki tujuan,

tanggung jawab, dan hasil yang ditetapkan sebelum melanjutkan ke tahap berikutnya (Fahmi Aziz & Sanjaya, 2023). Adapun tahapan metode *waterfall* pada gambar berikut:



Gambar 2. 1 Metode Waterfall

A. Analisis Kebutuhan

Analisis kebutuhan adalah langkah pertama dalam pengembangan aplikasi. Penulis melakukan analisis ini dengan menggunakan pendekatan basis kode objek. Setelah itu, penulis memilih kelas yang akan memenuhi kebutuhan aplikasi. Hasil dari langkah-langkah ini adalah spesifikasinya secara menyeluruh.

B. Desain

Menurut (Diana & Putra, 2021), pada tahap ini penulis membuat kerangka aplikasi terlebih dahulu yang terdiri dari Use Case Diagram, *Activity Diagram*, dan *Class Diagram*. Selain itu, peneliti juga mendesain *user interface* menggunakan figma. Figma dipilih karena merupakan alat desain *UI* yang memungkinkan kolaborasi secara *real-time* dan membantu untuk membuat desain tersebut.

C. Pengembangan

Tahap pengembangan merupakan proses mengimplementasikan desain sistem ke dalam bentuk kode program dengan menggunakan metode *Waterfall*, yang merupakan model pengembangan yang dilakukan secara bertahap dan sistematis. Tahapan metode ini meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Pada tahap analisis dilakukan pengumpulan data melalui observasi, wawancara, dan studi pustaka untuk menentukan kebutuhan sistem, kemudian dilanjutkan dengan perancangan menggunakan alat bantu seperti flowchart, Data Flow Diagram (DFD), dan relasi tabel. Selanjutnya, sistem diimplementasikan menggunakan bahasa pemrograman PHP dan *database MySQL*, kemudian dilakukan pengujian untuk memastikan sistem berjalan sesuai kebutuhan, dan diakhiri dengan tahap pemeliharaan untuk memperbaiki serta mengembangkan sistem agar lebih optimal. (Harjono & Kristianus Jago Tute, 2022)

D. Pengujian

Setelah pengembangan selesai, sistem harus melalui proses pengujian untuk aplikasi, maka peneliti akan merencanakan menggunakan metode pengujian *Black-Box*. Metode ini dipilih karena aplikasi yang dibangun adalah aplikasi sederhana. Oleh karena itu, hanya fungsionalitasnya yang diuji tanpa memeriksa struktur internalnya. Karena metode pengujian *Black-Box* hanya mengamati *input* dan *output* perangkat

lunak tanpa mengetahui struktur kodenya, pengujian ini dilakukan pada akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak berfungsi dengan baik atau tidak. ((Fafions Osama Effendy et al., 2025)).

E. Deployment

Setelah pengujian selesai dan sistem yang ditentukan sudah memenuhi semua persyaratan, tahap implementasi dimulai. Pada tahap ini aplikasi yang di instal dan di implementasikan ke dalam lingkungan produksi. Sehingga pengguna dapat mengakses aplikasi dalam bentuk *web* dan *mobile*. *Deployment* biasanya dilakukan secara bertahap, ada yang dalam bentuk beta release dan ada juga *full deployment*. Adanya akhir dari percobaan adalah mengevaluasi hasilnya. Ini melibatkan mempelajari karakteristik umum dari setiap aplikasi yang digunakan dan membandingkannya untuk mengetahui bagaimana masing-masing aplikasi berbeda dari yang lain. (Ramdan et al., 2021)

F. Pemeliharaan

Setelah tahap *deployment* selesai, tahap pemeliharaan dimulai. Pemeliharaan dapat dapat mencakup perbaikan kesalahan yang memungkinkan setelah aplikasi digunakan, peningkatan performa serta menambahkan fitur fitur yang jika dibutuhkan. Tahap ini biasanya sangat penting karena teknologi yang dibutuhkan selalu berkembang. Tanpa pemeliharaan aplikasi berpotensi menjadi usang serta tidak relevan dengan kebutuhan perusahaan. Tahap ini merupakan tahap pemeliharaan software. Semua software yang telah dibuat harus melewati tahap ini karena memungkinkan penambahan fitur baru dan perbaikan apabila terdapat error pada sistem. (Soepandi et al., 2023).

2. Flutter

Flutter adalah sebuah *SDK* atau *framework open source* yang dikembangkan oleh Perusahaan *google* yang mengembangkan aplikasi yang dapat berjalan di *Android* dan *web flutter* menggunakan bahasa pemrograman *dart* dan menyediakan widget bawaan yang kaya serta fleksibel sehingga dapat memudahkan pengguna untuk mengembangkan dalam merancang antarmuka yang sangat interaktif, responsif, dan konsisten keunggulan utama *flutter* yaitu *hot reload* yang mempercepat proses pengembangan dan pengujian aplikasi. Pada saat ini *flutter* juga menjadi salah satu solusi yang sangat mudah untuk mengembangkan aplikasi serta mempermudah proses *cross-platform development* (Renaldy & Pakereng, 2023)

3. Dart

Dart, sebuah pemrograman yang dikembangkan oleh *Google*, dapat digunakan untuk berbagai *platform*, seperti *web*, *mobile server*, dan *IOT*. Karena itu, *dart* bersifat *open source*, sehingga siapa pun dapat mengembangkan dan mempelajarinya. Menurut (Renaldy & Pakereng, 2023) *dart* memiliki banyak sintaks yang sederhana, mirip dengan bahasa pemrograman yang populer sehingga sangat mudah dipahami oleh pemula. Selain itu *dart* juga memiliki performa yang tinggi karena dapat kompilasi langsung menjadi kode mesin maupun dalam *javascript* untuk pengembangan *web*.

4. Firebase

Firebase merupakan model layanan yang bekerja di belakang layar dan menghubungkan aplikasi *web* dan *mobile* ke *cloud storage*. *Firebase* disimpan dalam bentuk *JSON* yang disinkronkan secara *real-time* ke setiap klien yang terhubung. Ketika penulis membuat aplikasi lintas-platform dengan *SDK android*, *IOS*, dan

javascript, semua klien akan berbagi sebuah *instance real-time database* dan menerima update data terbaru secara otomatis. Menurut (Astuti & Hermawan, n.d.), selain itu *firebase* juga sangat membantu dalam pengembangan dan pengelolaan infrastruktur *backend* tanpa perlu membangun server dari awal, sehingga dapat memfokuskan pada pengembangan fitur aplikasi.

5. Black-Box

Untuk pengujian aplikasi, penelitian merencanakan akan menggunakan metode pengujian *black-box*. Metode pengujian perangkat lunak atau aplikasi yang dipilih, mengingat karena aplikasi dibangun adalah aplikasi yang sederhana. Sehingga yang diuji hanya tentang fungsionalitasnya tanpa melihat struktur internalnya.

Metode pengujian *Black-Box* Testing merupakan pengujian yang dilakukan utamanya hanya untuk mengamati hasil input dan output dari perangkat lunak saja, tanpa perlu mengetahui struktur kode perangkat lunak yang digunakan. Menurut (Fafions Osama Effendy et al., 2025)

BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan

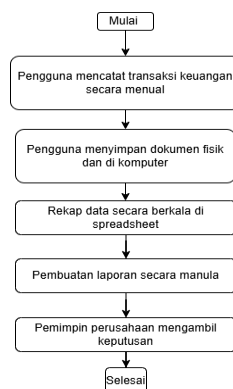
Analisis kebutuhan dilakukan agar mengetahui kondisi proses bisnis yang sedang berjalan sekaligus merumuskan gambaran awal dari aplikasi keuangan dan penyimpanan dokumen yang masih dilakukan secara normal dengan spreadsheet dan penyimpanan file secara terpisah. Hal ini sering terjadi kendala, seperti pencatatan yang lambat, kesalahan input data, serta kesulitan dalam mencari dokumen kembali yang sudah disimpan.

Berdasarkan masalah tersebut dikembangkan *system* yang dapat mengelolah dengan cepat dan aman serta terintegrasi dari data keuangan dan arsip dokumen. Sistem yang dibangun adalah aplikasi *web* dan *mobile* yang berbasis *flutter* sebagai *framework* utama dan *firebase* sebagai data base sekaligus penyimpanan dokumen. Ada beberapa kebutuhan utama untuk pengguna yang antara lain adalah autentikasi pengguna, pencatatan keuangan, pengelolaan dan penyimpanan dokumen secara terpusat, pencarian data dengan lebih mudah dan cepat serta dapat membuat laporan keuangan dalam format digital dan bisa juga dicetak/print. Dengan adanya aplikasi tersebut diharapkan proses berjalan dengan lancar dan akurat.

3.1.1 Proses Bisnis Berjalan

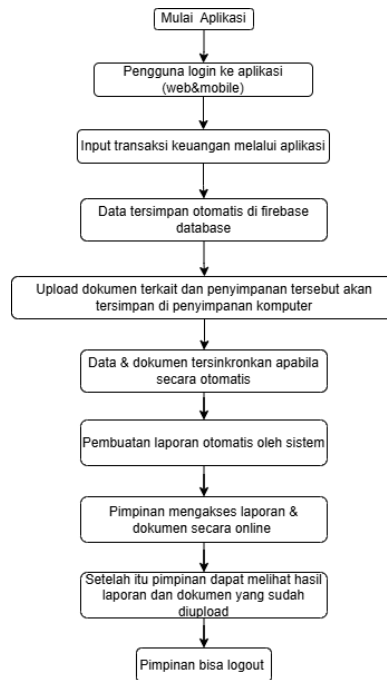
Saat ini, *flutter* dan *firebase* digunakan untuk membuat aplikasi keuangan dan penyimpanan dokumen berbasis *web* dan *mobile* yang digunakan untuk mencatat transaksi keuangan, menyimpan dokumen, mengelola hak akses, serta membuat laporan. Proses ini berjalan cukup lancar karna data tersimpan di *cloud* secara *real-time* dan dapat diakses kapan saja. Namun, masih terdapat beberapa kendala, seperti ketergantungan pada internet, penyimpanan dokumen yang kurang baik, laporan keuangan masih sederhana, hak akses terbatas, kapasitas penyimpanan terbatas, sehingga beberapa pekerjaan masih dilakukan secara manual. Berikut gambaran *Flowchart* sebelum dan sesudah adanya aplikasi

c. *Flowchart* (Alur Sebelum Menggunakan Aplikasi)



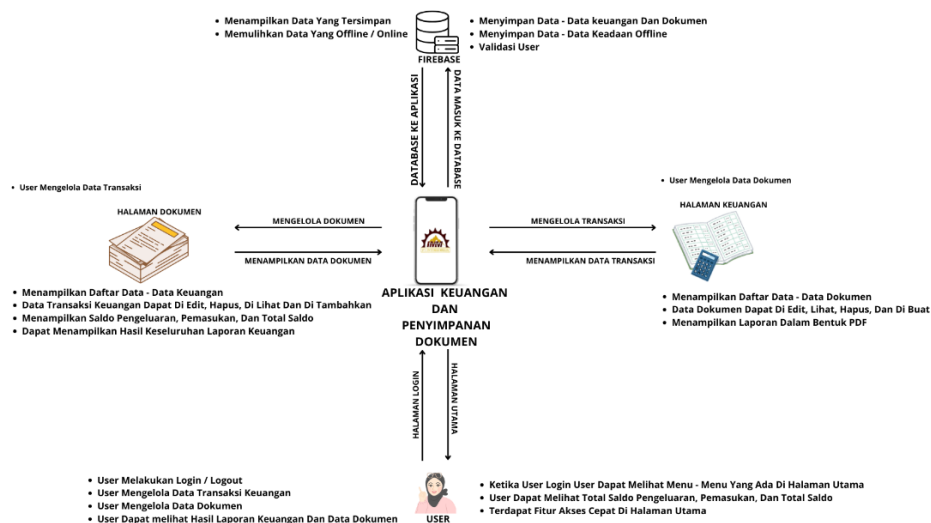
Gambar 3. 1 Alur Sebelum Menggunakan Aplikasi

- **Flowchart (Alur Sesudah Menggunakan Aplikasi)**



Gambar 3. 2 Alur Sesudah Menggunakan Aplikasi

3.1.2 Gambaran Umum Sistem



Gambar 3. 3 Gambaran Umum

Gambar diatas merupakan ilustrasi dari gambaran umum pada aplikasi keuangan dan penyimpanan dokumen berbasis *web* dan *mobile*. Peran dalam tiap-tiap elemen dalam gambar berikut yaitu:

1. *Firebase*, menyimpan data transaksi keuangan dan data dokumen, menyimpan data dalam keadaan *offline* maupun *online*, mengembalikan data yang tersimpan untuk ditampilkan kembali ke aplikasi, melakukan validasi *user login*, menyinkronkan data antara *database* secara *real-time*
2. *User*, *User* melakukan *login* dan *logout*, dapat mengelola data transaksi keuangan dan dokumen, *user* dapat melihat laporan keuangan dan data dokumen di halaman utama.
3. Halaman Keuangan, di halaman keuangan tersebut menampilkan daftar data transaksi keuangan, data transaksi keuangan dapat diedit, dihapus, dilihat, dan ditambahkan.
4. Halaman Dokumen, halaman dokumen tersebut dapat menampilkan daftar data dokumen yang tersimpan, data dokumen tersebut dapat diedit, dilihat, dihapus, dan dibuat.
5. Halaman Utama, menampilkan menu – menu utama untuk navigasi, menampilkan total saldo pemasukan, pengeluaran, dan saldo akhir, menyediakan fitur – fitur akses cepat ke halaman keuangan dan dokumen.

3.2 Perancangan

Perancangan Aplikasi dalam penelitian ini terdiri dari analisa kebutuhan dan *use-case* diagram.

3.2.1 Kebutuhan Fungsional

Tabel 3. 1 Kebutuhan Fungsional

No	Kebutuhan Fungsional	Deskripsi
1	Autentikasi <i>User</i>	Sistem menyediakan fitur <i>login</i> dan <i>logout</i>
2	Validasi <i>User</i>	Sistem harus memvalidasi akun yang sudah terdaftar dan menolak akses yang tidak terdaftar
3	Halaman Utama	Setelah <i>Login</i> Sistem akan mengarahkan <i>user</i> ke halaman utama yang berisikan data keseluruhan transaksi keuangan keluar masuk serta dapat melihat menu halaman keuangan dan halaman dokumen.
4	Halaman Keuangan	<i>User</i> dapat melihat daftar data transaksi keuangan seperti transaksi keluar, masuk, dan saldo keseluruhan.
5	Input Transaksi	<i>User</i> dapat mencatat keuangan atau menambahkan data transaksi keuangan yang ada di halaman keuangan.
6	Kelola Transaksi	Sistem menyediakan fitur untuk melihat laporan transaksi, mengedit, dan menghapus data transaksi
7	Halaman Dokumen	<i>User</i> dapat melihat daftar data data yang sudah ada di halaman dokumen
8	Upload Dokumen	Sistem memungkinkan <i>user</i> untuk mengunggah dokumen (Gambar, Pdf dan lain-lainnya).
9	Simpan Dokumen	Dokumen yang sudah diunggah disimpan kedalam penyimpanan lokal perangkat dan juga disinkronkan ke <i>cloud</i> .
10	Simpan Transaksi	Sistem menyimpan data transaksi yang sudah di input kedalam <i>database</i> lokal perangkat dan disinkronkan ke <i>cloud</i> .
12	Kelola Dokumen	Sistem Menyediakan fitur fitur untuk melihat, mengedit, membuat, serta menghapus data dokumen.
13	Hapus Dokumen	Sistem menyediakan fitur untuk menghapus dokumen apabila <i>user</i> salah memasukan dokumen

No	Kebutuhan Fungsional	Deskripsi
14	Akses <i>Offline</i>	User dapat melakukan aktifitas di dalam aplikasi ketika koneksi internet dalam keadaan mati (<i>Offline</i>)
15	Generate Laporan PDF	Jika user ingin mencetak laporan atau melihat laporan – laporan yang sudah di buat user dapat melakukannya dengan men generate laporan tersebut ke PDF agar dapat mencetak dan melihat laporan secara detail.

Kebutuhan *Fungsional* adalah jenis kebutuhan yang berisis proses-proses yang nantinya akan dilakukan oleh Aplikasi tersebut, selain itu berisi tentang informasi-informasi yang dapat dihasilkan oleh aplikasi.

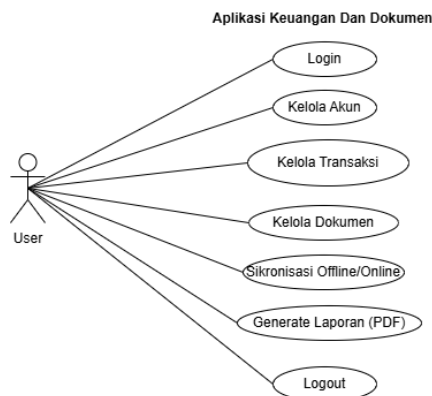
3.2.2 Kebutuhan *Non-Fungsional*

Tabel 3. 2 Kebutuhan Non Fungsional

No	Kebutuhan Non-Fungsional	Deskripsi
1	Keamanan	Sistem ini harus menjamin keamanan data dan autentikasi user.
2	Ketersediaan	Sistem dapat digunakan dalam kondisi <i>offline</i> karena data transaksi dan dokumen tersimpan di penyimpanan lokal perangkat
3	<i>Performance</i>	Sistem mampu menyimpan dan menampilkan tampilan transaksi maupun dokumen secara cepat
4	Skalabilitas	Sistem dapat dikembangkan lebih lanjut untuk mendukung penyimpanan berbasis <i>cloud</i>
5	Portabilitas	Aplikasi dapat dijalankan diberbagai perangkat <i>mobile</i> serta dapat dijalankan melalui <i>website</i>
6	<i>Reliabilitas</i>	Sistem mampu menyimpan data transaksi dan dokumen dengan konsisten
7	Pemeliharaan	Sistem dirancang dengan struktur kode yang rapi agar mudah dilakukan untuk pemeliharaan
8	Efisiensi Penyimpanan	Sistem mengelola penyimpanan lokal perangkat secara efisien agar file dokumen dan transaksi tidak memakan ruang yang berlebihan

Kebutuhan *Non-Fungsional* adalah sebuah kebutuhan aplikasi yang terkait dengan kualitas dan keadaan aplikasi tersebut

3.2.3 Diagram *Use-Case*



Gambar 3. 4 Diagram Use Case

Diagram *use-case* ini menggambarkan bahwa *User* sebagai aktor utama dapat berinteraksi dengan berbagai fitur dalam Aplikasi Keuangan dan Dokumen, mulai dari melakukan *Login*, mengelola informasi akun, serta melakukan proses *CRUD* pada data transaksi dan pengelolaan dokumen. Selain itu, aplikasi menyediakan fungsi Sinkronisasi *Offline/Online* untuk memastikan data tetap konsisten meskipun terjadi perubahan kondisi jaringan. *User* juga dapat menghasilkan laporan dalam format PDF melalui fitur generate laporan. Setelah menyelesaikan aktivitasnya, *User* dapat keluar dari sistem menggunakan fitur *Logout*.

3.2.4 Skenario *Use-Case*

Tabel 3. 3 Login Aplikasi

Nama Use Case	<i>Login Aplikasi</i>
Aktor	<i>User</i>
Deskripsi	Pengguna masuk ke dalam aplikasi menggunakan username dan password yang sudah terdaftar
Skenario	
Aksi	Memasukan username dan password lalu tekan tombol <i>login</i>
Sistem	Sistem memvalidasi data <i>login</i> dengan <i>firebase authentication</i>

Tabel 3. 4 Halaman Utama

Nama Use Case	Halaman Utama
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat melihat halaman utama keuangan dan dokumen serta saldo transaksi keuangan pemasukan, pengeluaran, dan total saldo
Skenario	
Aksi	<i>User</i> bisa memilih halaman keuangan atau halaman dokumen yang sudah bisa memiliki fitur akses cepat
Sistem	<i>User</i> bisa mengklik bar yang ada di halaman utama untuk mengakses ke halaman keuangan maupun dokumen

Tabel 3. 5 Halaman Keuangan & Menginput Keuangan

Nama Use Case	Halaman Keuangan & Menginput Keuangan
----------------------	---------------------------------------

Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat Melihat daftar data data transaksi keuangan serta mencatat transaksi keuangan pengeluaran dan pemasukan
Skenario	
Aksi	<i>User</i> dapat mengklik tombol buat untuk membuat catatan keuangan pemasukan dan pengeluaran, lalu user bisa mengklik simpan
Sistem	Sistem menyimpan data transaksi ke <i>firebase</i> , setelah itu akan muncul dihalaman keuangan yang sudah kita buat tadi

Tabel 3. 6 Halaman Dokumen

Nama Use Case	Halaman Dokumen
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat melakukan pengunggahan dokumen yang dipilih dari perangkat serta dapat membuat dan mengedit dokumen
Skenario	
Aksi	<i>User</i> mengklik tombol <i>upload</i> dan memilih dari file perangkat serta <i>user</i> dapat mengklik tombol buat ketika <i>user</i> ingin membuat dokumen sendiri
Sistem	Sistem dapat menyimpan file ke perangkat metadata ke <i>firebase</i> lalu menampilkan daftar dokumen yang sudah diupload dan yang sudah dibuat

Tabel 3. 7 Laporan Keuangan

Nama Use Case	Halaman Laporan Keuangan
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat melihat ringkasan laporan keuangan
Skenario	

Aksi	<i>User</i> mengklik tombol invoice untuk melihat ringkasan keuangan
Sistem	Sistem akan menampilkan laporan singkat keuangan dalam bentuk PDF

Tabel 3. 8 Edit Keuangan & Simpan Keuangan

Nama Use Case	Edit Keuangan
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat mengedit keuangan untuk merubah isi didalam keuangan tersebut
Skenario	
Aksi	<i>User</i> bisa mengklik tombol edit jika ingin merubah isi didalamnya seperti isi, deskripsi, harga.
Sistem	Sistem akan mengupdate perubahan yang sudah dilakukan oleh <i>user</i>

Tabel 3. 9 Hapus Keuangan

Nama Use Case	Hapus Keuangan
Aktor	<i>User</i>
Deskripsi	<i>User</i> bisa menghapus inputan keuangan apabila tidak dipakai lagi
Skenario	
Aksi	<i>User</i> tinggal mengklik tombol hapus
Sistem	Sistem akan mengkonfirmasi hapus dan data terhapus

Tabel 3. 10 Simpan Transaksi

Nama Use Case	Simpan Transaksi
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat menyimpan transaksi ke dalam penyimpanan lokal perangkat
Skenario	
Aksi	<i>User</i> mengklik tombol simpan yang sudah ada di format keuangan
Sistem	Sistem akan menyimpan data transaksi kedalam perangkat

Tabel 3. 11 Simpan Dokumen

Nama Use Case	Simpan Dokumen
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat menyimpan dokumen kedalam penyimpanan lokal perangkat
Skenario	
Aksi	<i>User</i> mengklik tombol simpan yang sudah ada di format dokumen
Sistem	Sistem akan menyimpan data dokumen ke dalam perangkat

Tabel 3. 12 Edit Dokumen

Nama Use Case	Edit Dokumen
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat mengedit dokumen yang salah dalam pengisian dokumen baru sebelumnya
Skenario	
Aksi	<i>User</i> dapat mengklik tombol edit yang sudah ada di halaman dokumen
Sistem	Sistem akan memperbarui data dokumen yang sudah diedit oleh <i>user</i> serta menyimpannya kembali ke dalam <i>database</i>

Tabel 3. 13 Hapus Dokumen

Nama Use Case	Hapus Dokumen
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat melakukan penghapusan dokumen yang tidak terpakai lagi
Skenario	
Aksi	<i>User</i> dapat mengklik tombol hapus yang sudah tersedia di bawah tombol edit
Sistem	Sistem akan mengkonfirmasi hapus dokumen tanpa menampilkan pesan ke <i>user</i>

Tabel 3. 14 Upload Dokumen

Nama Use Case	Upload Dokumen
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat mengupload dokumen yang ingin ditambahkan kedalam halaman dokumen
Skenario	
Aksi	<i>User</i> membuka halaman dokumen dengan mengklik tombol upload yang ada di halaman dokumen
Sistem	Sistem akan menampilkan lokal direktori pada perangkat untuk mengupload dokumen yang ingin <i>user</i> upload

Tabel 3. 15 Logout

Nama Use Case	<i>Logout</i>
Aktor	Semua pengguna aplikasi
Deskripsi	Setelah melakukan aktivitas yang ada didalam aplikasi <i>user</i> bisa langsung keluar dari aplikasi
Skenario	
Aksi	<i>User</i> dapat mengklik tombol <i>logout</i> yang sudah ada di dalam halaman profile
Sistem	Sistem akan menghapus sesi masuk yang aktif di aplikasi

Tabel 3. 16 Akses Offline

Nama Use Case	Akses <i>Offline</i>
Aktor	<i>User</i>
Deskripsi	<i>User</i> dapat melakukan aktivitas terhadap aplikasi jika tidak ada koneksi internet
Skenario	

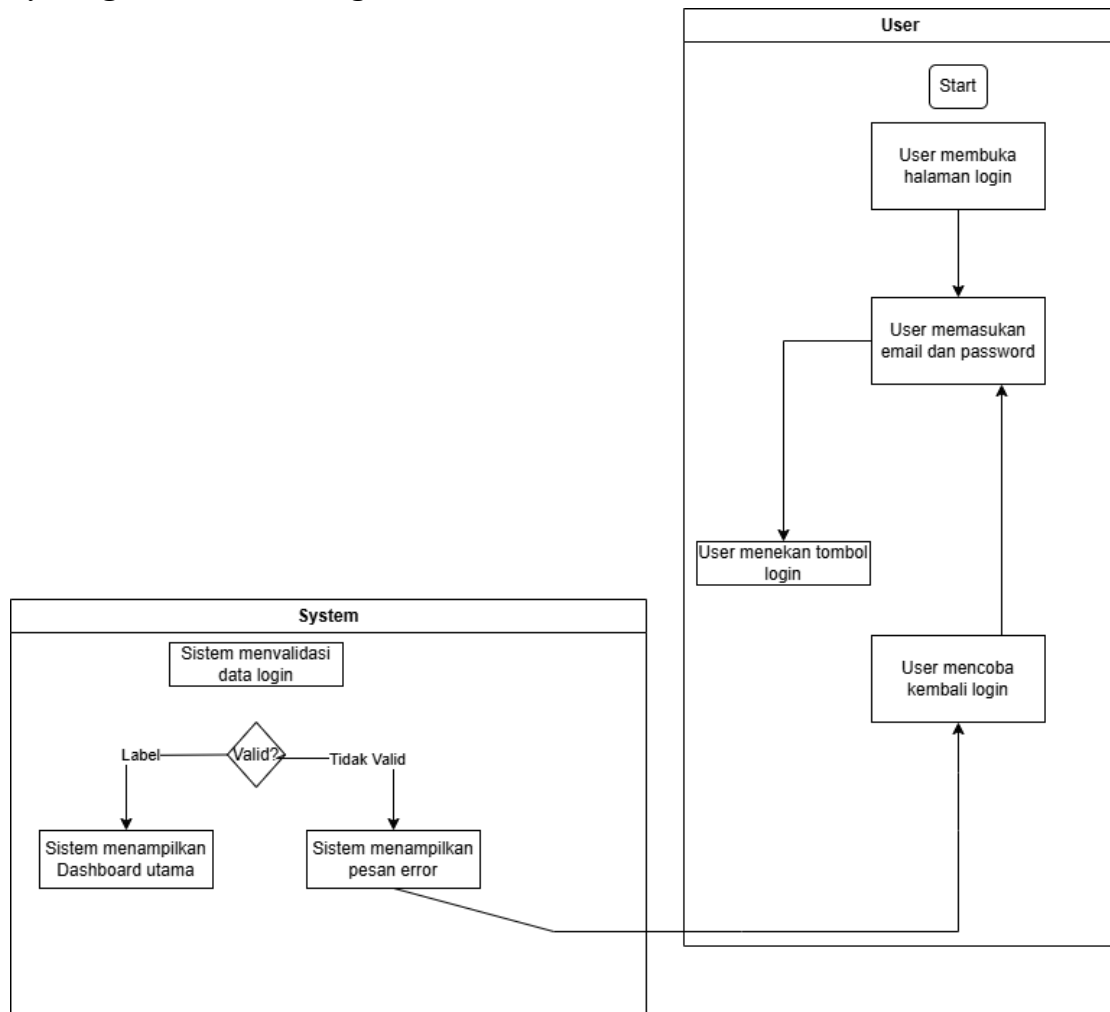
Aksi	Jika <i>user</i> melakukan aktivitas di dalam aplikasi maka aktivitas itu akan terjadi sementara dikarenakan tidak ada koneksi internet
Sistem	Setelah <i>user</i> melakukan aktivitas tersebut sistem akan langsung menyimpan data sementara. Ketika koneksi internet sudah terhubung maka data tersebut akan terkonfirmasi oleh sistem

Tabel 3. 17 Generate laporan PDF

Nama Use Case	Generate Laporan PDF
Aktor	<i>User</i>
Deskripsi	Jika <i>user</i> ingin melihat dan mengunduh laporan tersebut <i>user</i> bisa melihatnya dalam bentuk PDF
Skenario	
Aksi	<i>User</i> memilih menu laporan yang sudah ada di keuangan dan dokumen. <i>User</i> memilih jenis laporan yang ingin dilihat dan diunduh. <i>User</i> menekan tombol Generate PDF untuk membuat laporan dalam bentuk PDF.
Sistem	Sistem akan menerima permintaan dan mengambil data laporan yang diperlukan. Sistem kemudian memproses dan menghasilkan file PDF berdasarkan data yang diminta oleh <i>user</i> . Setelah PDF berhasil dibuat, sistem akan menampilkan opsi bagi <i>user</i> untuk melihat atau mengunduh.

3.2.5 Activity Diagram

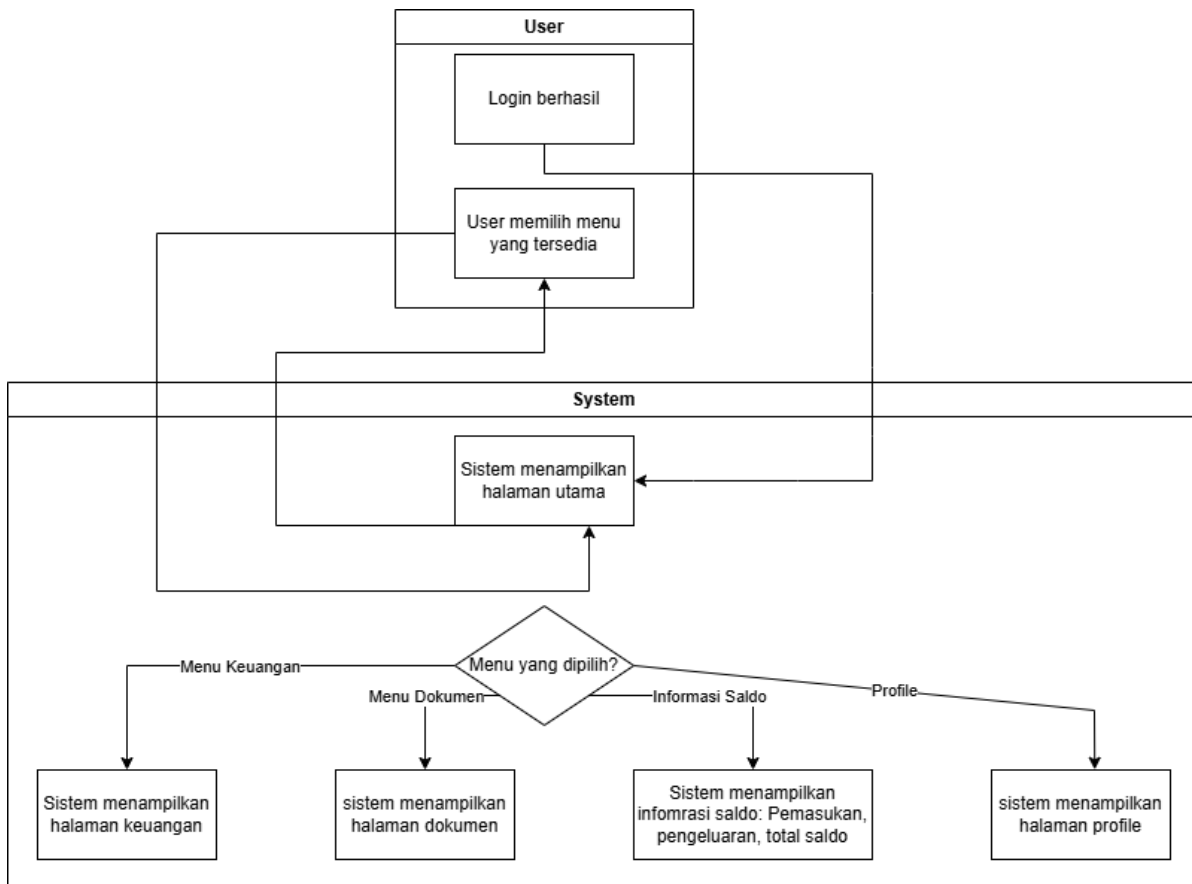
1. Activity Diagram Halaman Login



Gambar 3. 5 Activity Diagram Halaman Login

Activity diagram halaman *login* pengguna. Pengguna membuka halaman *login*, memasukkan email dan password, lalu menekan tombol *login*. Sistem kemudian memvalidasi input; jika data tidak valid, sistem langsung menampilkan pesan error. Jika valid, sistem melakukan autentikasi ke *firebase Auth*. Apabila autentikasi gagal, sistem kembali menampilkan pesan error pada pengguna, dan jika berhasil, *UI* akan mengarahkan pengguna ke halaman utama.

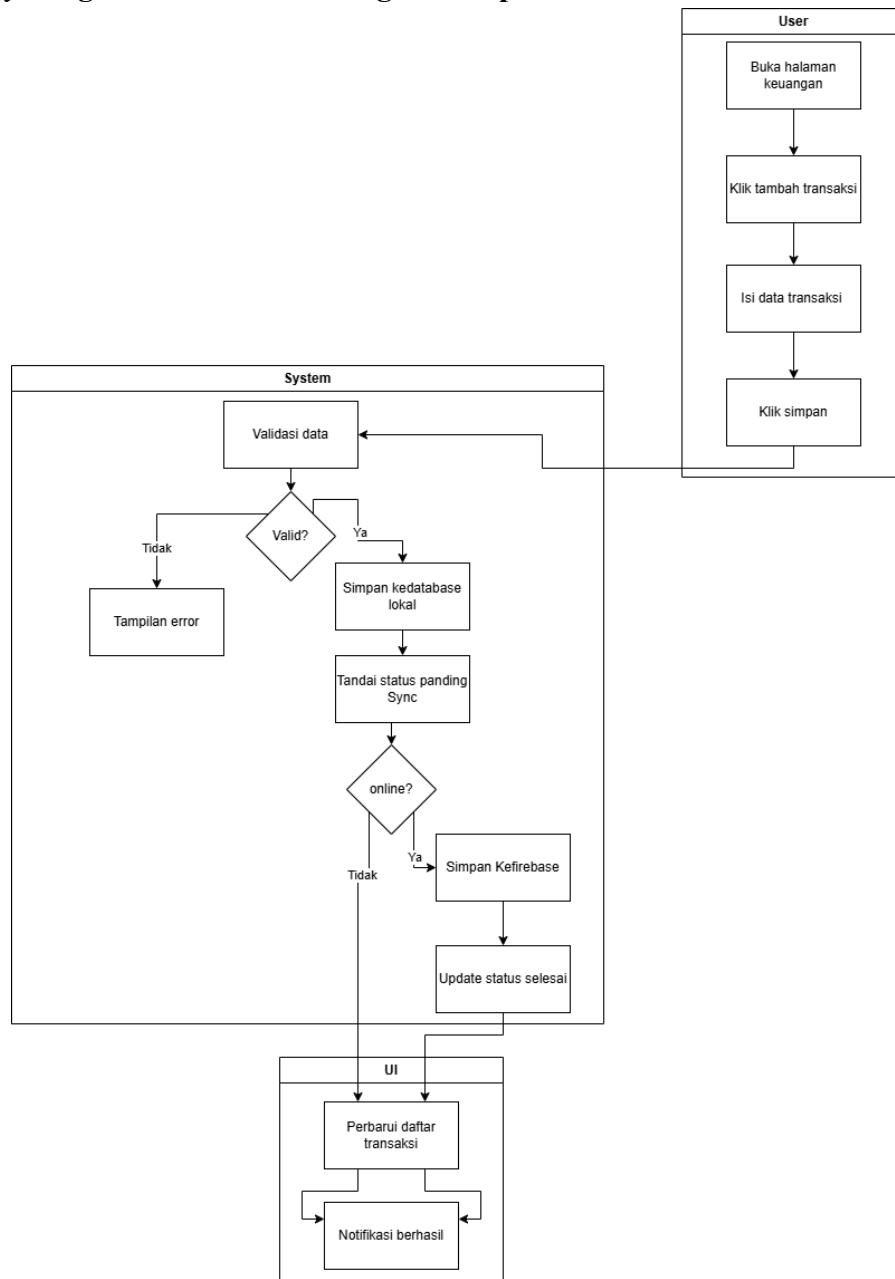
2. Activity Diagram Halaman Utama



Gambar 3. 6 Activity Diagram Halaman Utama

Activity diagram tersebut menggambarkan proses saat pengguna membuka halaman utama. Ketika halaman utama dibuka, sistem mengambil data transaksi dan dokumen dari *database* lokal terlebih dahulu. Setelah itu, sistem akan mengecek apakah koneksi internet tersedia. Jika tidak ada koneksi, sistem langsung menampilkan statistik dan daftar data dari *database* lokal. Jika koneksi tersedia, sistem melakukan sinkronisasi data ke *firebase* dan kemudian memperbarui *database* lokal dengan data terbaru sebelum akhirnya *UI* menampilkan statistik dan daftar data kepada pengguna.

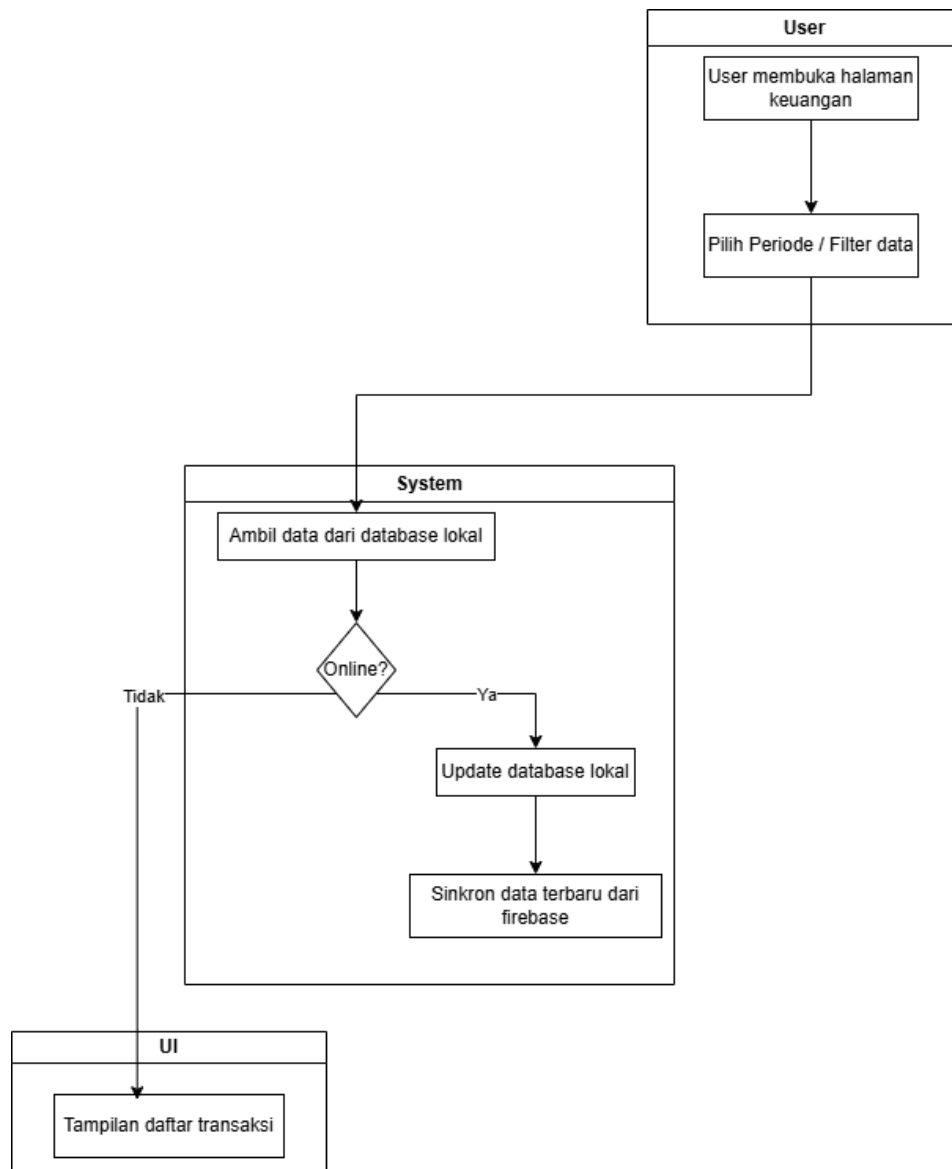
3. Activity Diagram Halaman keuangan & Input Transaksi



Gambar 3. 7 Activity Diagram Input Keuangan

Activity diagram tersebut menjelaskan alur ketika pengguna menambahkan transaksi pada halaman keuangan. Pengguna membuka halaman keuangan, menekan tombol tambah transaksi, mengisi data, lalu menekan tombol simpan. Sistem kemudian memvalidasi data; jika tidak valid, sistem menampilkan pesan error. Jika valid, data disimpan ke *database* lokal dan ditandai sebagai pending *sync*. Selanjutnya, sistem mengecek koneksi internet; jika *offline*, UI langsung memperbarui daftar transaksi. Jika *online*, sistem melakukan sinkronisasi ke *Firebase*, memperbarui status transaksi menjadi selesai, kemudian UI menampilkan daftar transaksi terbaru beserta notifikasi berhasil.

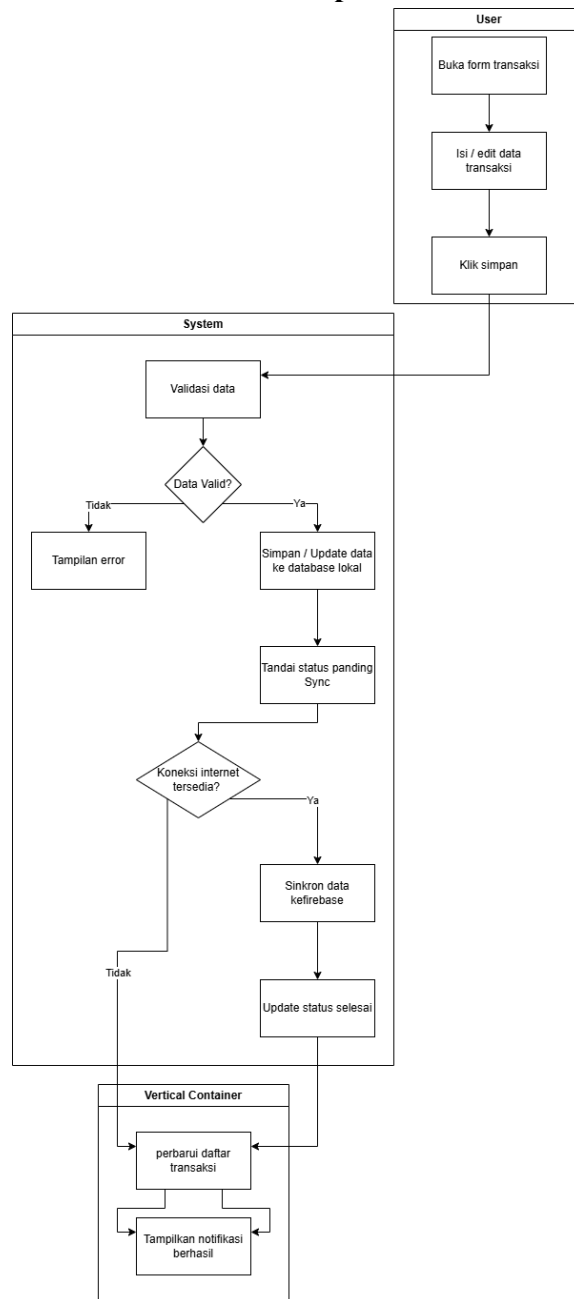
4. Activity Diagram Lihat Transaksi



Gambar 3. 8 Activity Diagram Lihat Transaksi

Activity diagram tersebut menggambarkan proses ketika pengguna membuka halaman transaksi dan memilih periode atau filter data. Setelah itu, sistem mengambil data dari *database* lokal sebagai sumber utama tampilan. Kemudian sistem memeriksa ketersediaan koneksi internet; jika tidak ada koneksi, *UI* langsung menampilkan daftar transaksi berdasarkan data lokal. Jika koneksi internet tersedia, sistem melakukan sinkronisasi dengan mengambil data terbaru dari *firebase* dan memperbarui *database* lokal, lalu *UI* menampilkan daftar transaksi yang sudah diperbarui.

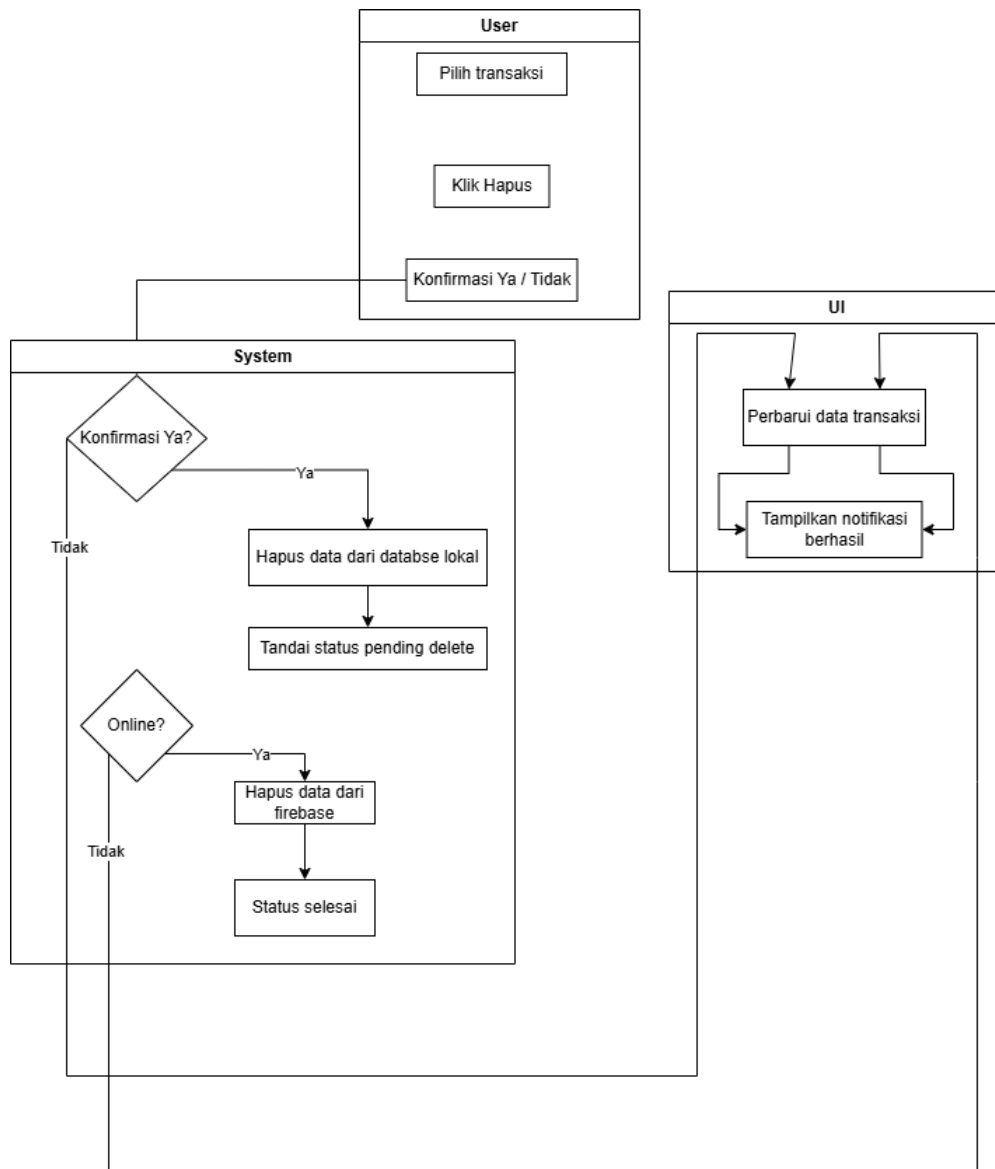
5. Activity Diagram Edit Transaksi & Simpan Transaksi



Gambar 3. 9 Activity Diagram Edit Transaksi

Activity diagram tersebut menggambarkan alur ketika pengguna membuka form transaksi untuk menambah atau mengedit data, kemudian mengisi informasi dan menekan tombol simpan. Sistem akan memvalidasi data yang dimasukan; jika tidak valid, sistem menampilkan pesan error. Jika data valid, sistem akan menyimpan atau memperbarui data ke *database* lokal dan menandainya sebagai pending *sync*. Setelah itu, sistem memeriksa apakah koneksi internet tersedia; jika *online*, sistem melakukan sinkronisasi data ke *Firestore* dan memperbarui status menjadi selesai, sedangkan jika *offline* proses sinkron akan menunggu. Terakhir, *UI* memperbarui daftar transaksi dan menampilkan notifikasi berhasil kepada pengguna.

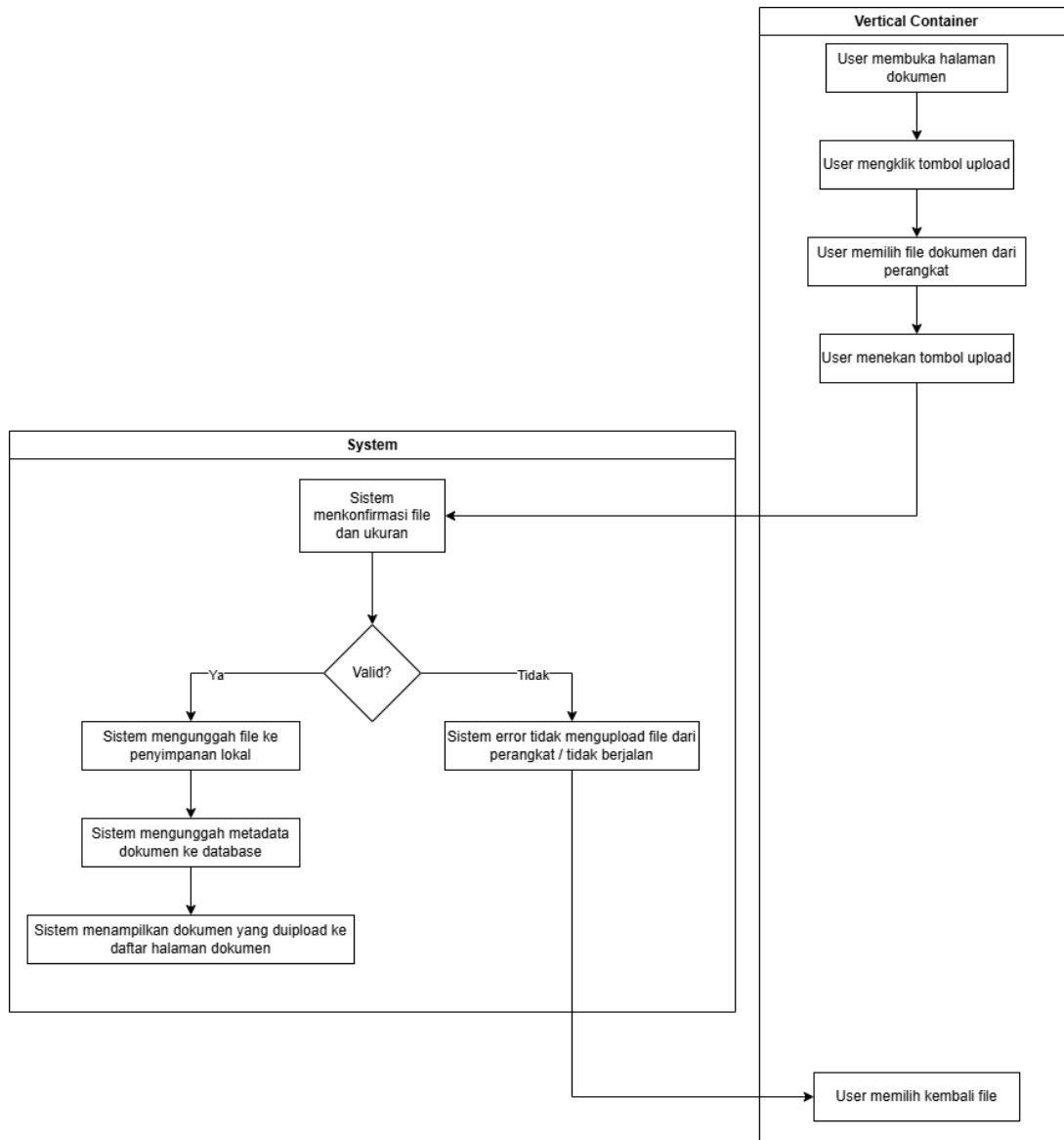
6. Activity Diagram Hapus Transaksi



Gambar 3. 10 Activity Diagram Hapus Transaksi

Activity diagram tersebut menjelaskan alur penghapusan transaksi, di mana pengguna memilih transaksi dan mengkonfirmasi Tindakan hapus. Jika pengguna menekan “Ya”, sistem menghapus data data dari *database* lokal dan menandainya sebagai *pending delete*. Selanjutnya, sistem akan mengecek koneksi internet; jika *online*, data langsung dihapus dari *Firebase* dan status selesai; jika tidak, penghapusan *Firebase* ditunda sampai perangkat terhubung ke koneksi internet. Setelah proses ini, *UI* memperbarui daftar transaksi dan menampilkan notifikasi bahwa penghapusan berhasil.

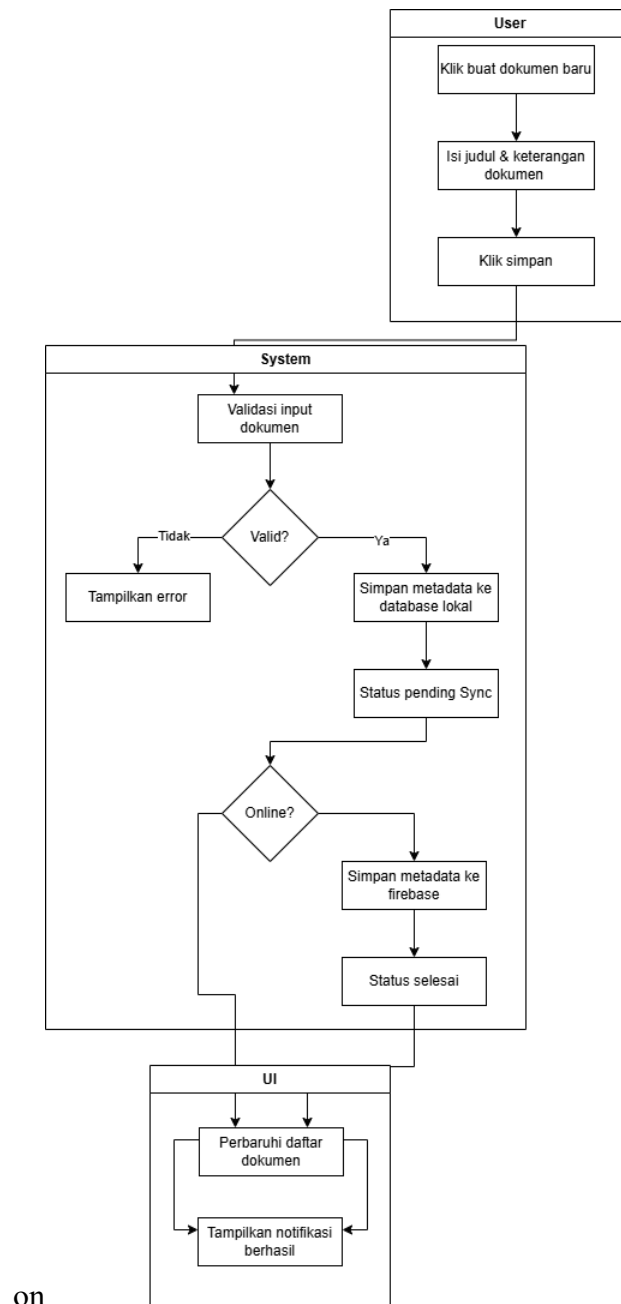
7. Activity Diagram Halaman Dokumen & Upload Dokumen



Gambar 3. 11 Activity Diagram Upload Dokumen

Activity diagram tersebut menggambarkan proses upload dokumen, dimulai ketika *user* membuka halaman dokumen, memilih file, dan menekan tombol upload. Sistem kemudian memeriksa validitas file berdasarkan format dan ukuran; jika file tidak valid, sistem menampilkan pesan error dan meminta *user* memilih file lain. Jika valid, sistem mengunggah file ke penyimpanan lokal, menyimpan metadata dokumen ke *database*, dan akhirnya menampilkan dokumen yang berhasil terupload pada daftar halaman dokumen.

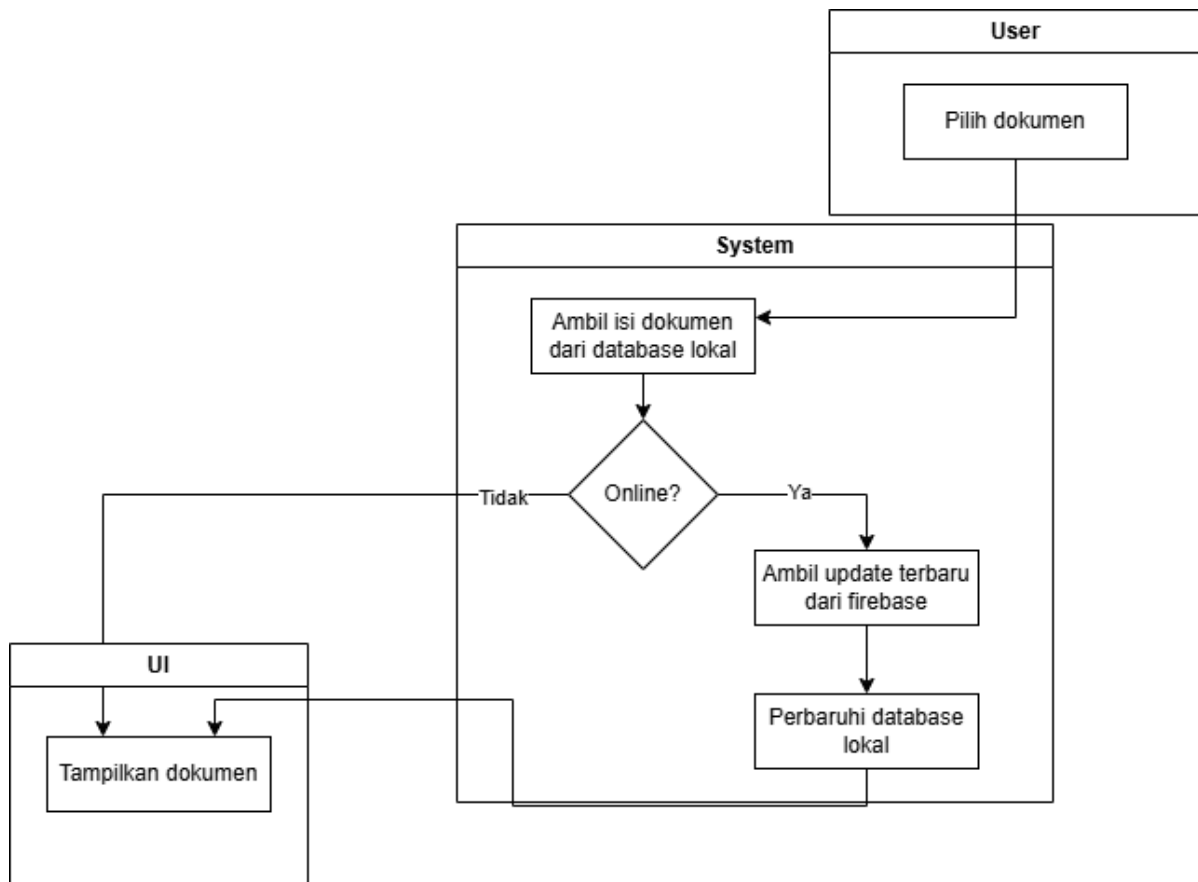
8. Activity Diagram Buat Dokumen



Gambar 3. 12 Activity Diagram Buat Dokumen

Activity diagram tersebut menggambarkan alur pembuatan dokumen baru, dimulai ketika *user* mengisi judul dan keterangan lalu menekan simpan. Sistem memvalidasi input, dan jika tidak valid, muncul pesan error. Jika valid, metadata tersimpan ke *database* lokal dan diberi status pending *sync*. Sistem kemudian mengecek koneksi; jika *online*, metadata langsung dikirim ke *Firebase* hingga status menjadi selesai; jika *offline*, sinkronisasi dilakukan nanti. Setelah itu, *UI* memperbarui daftar dokumen dan menampilkan notifikasi bahwa proses berhasil.

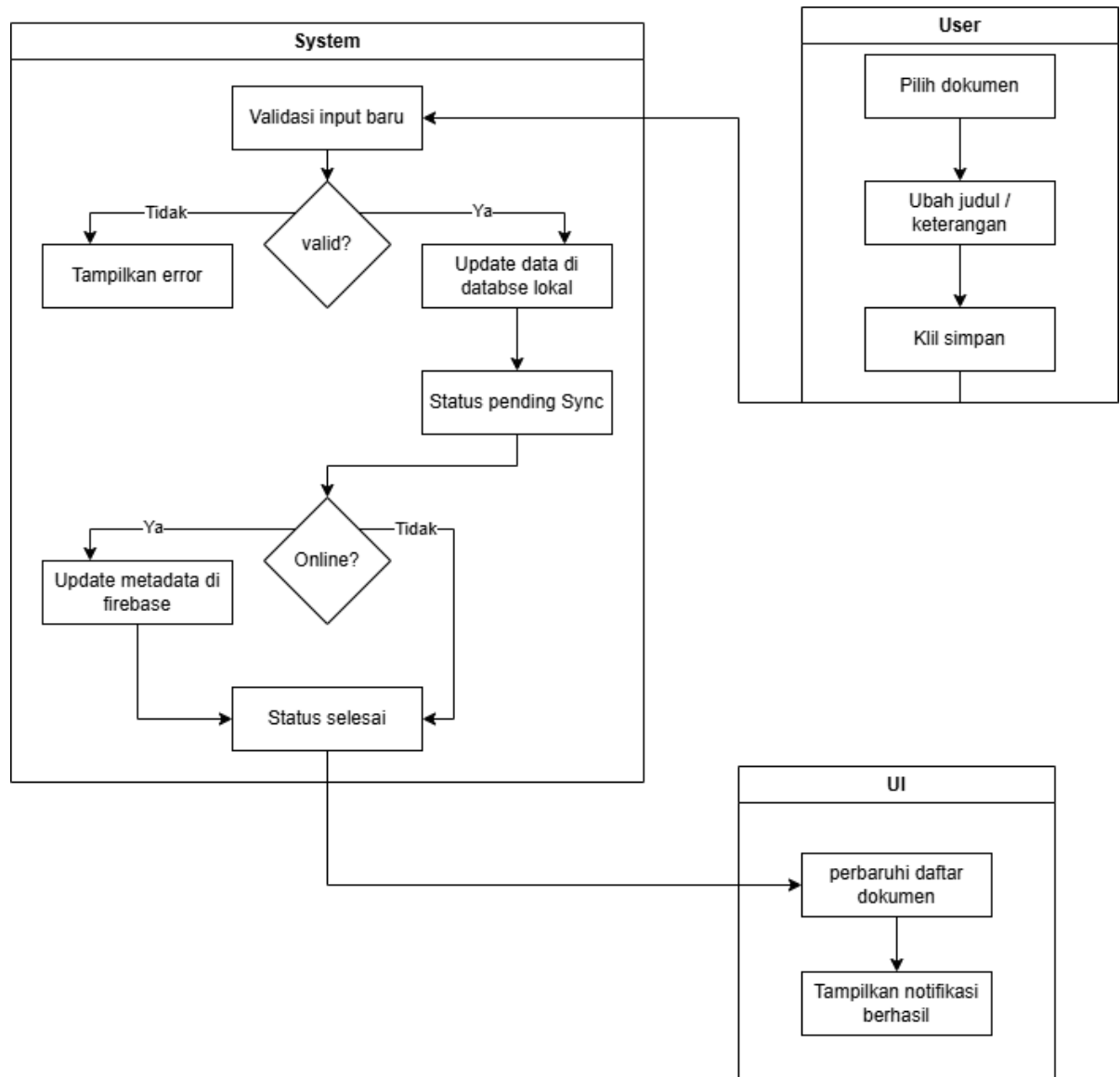
9. Activity Diagram Lihat Dokumen



Gambar 3. 13 Activity Diagram Lihat Dokumen

Activity diagram tersebut menggambarkan alur ketika pengguna memilih dokumen: sistem pertama – tama mengambil isi dokumen dari *database* lokal, kemudian memeriksa apakah perangkat sedang *online*. Jika *online*, sistem mengambil pembaruan terbaru ke *Firebase* dan memperbarui database lokal; jika tidak, proses langsung berlanjut. Setelah itu, antarmuka (*UI*) menampilkan dokumen kepada pengguna.

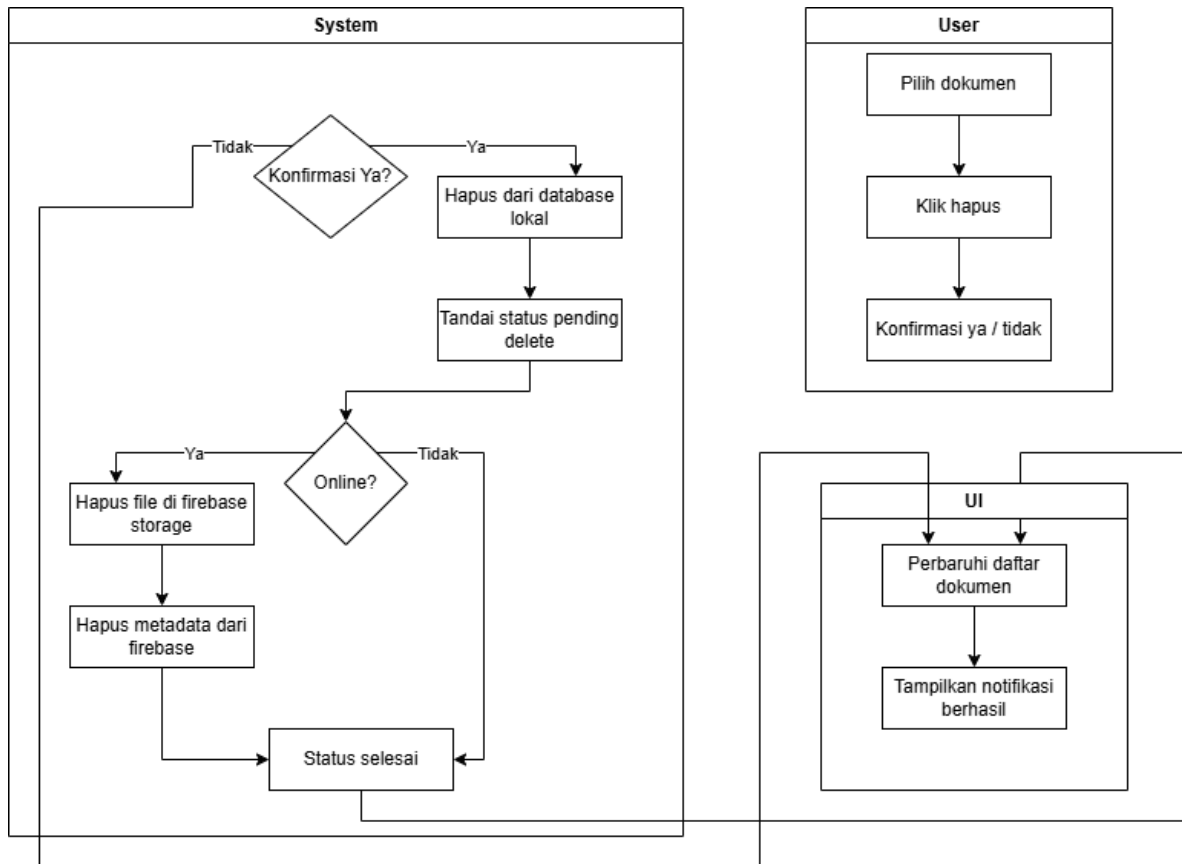
10. Activity Diagram Edit Dokumen



Gambar 3. 14 Activity Diagram Edit Dokumen

Activity diagram tersebut menjelaskan alur ketika pengguna mengubah judul atau keterangan dokumen lalu menyimpannya; sistem terlebih dahulu memvalidasi input baru, dan jika tidak valid maka menampilkan pesan error. Jika valid, data diperbarui ke *database* lokal dan diberi status “pending sync”. Setelah itu sistem mengecek koneksi; jika *online*, metadata dokumen diperbarui ke *Firebase* hingga berstatus sukses. Terakhir, *UI* memperbarui daftar dokumen dan menampilkan notifikasi bahwa proses berhasil.

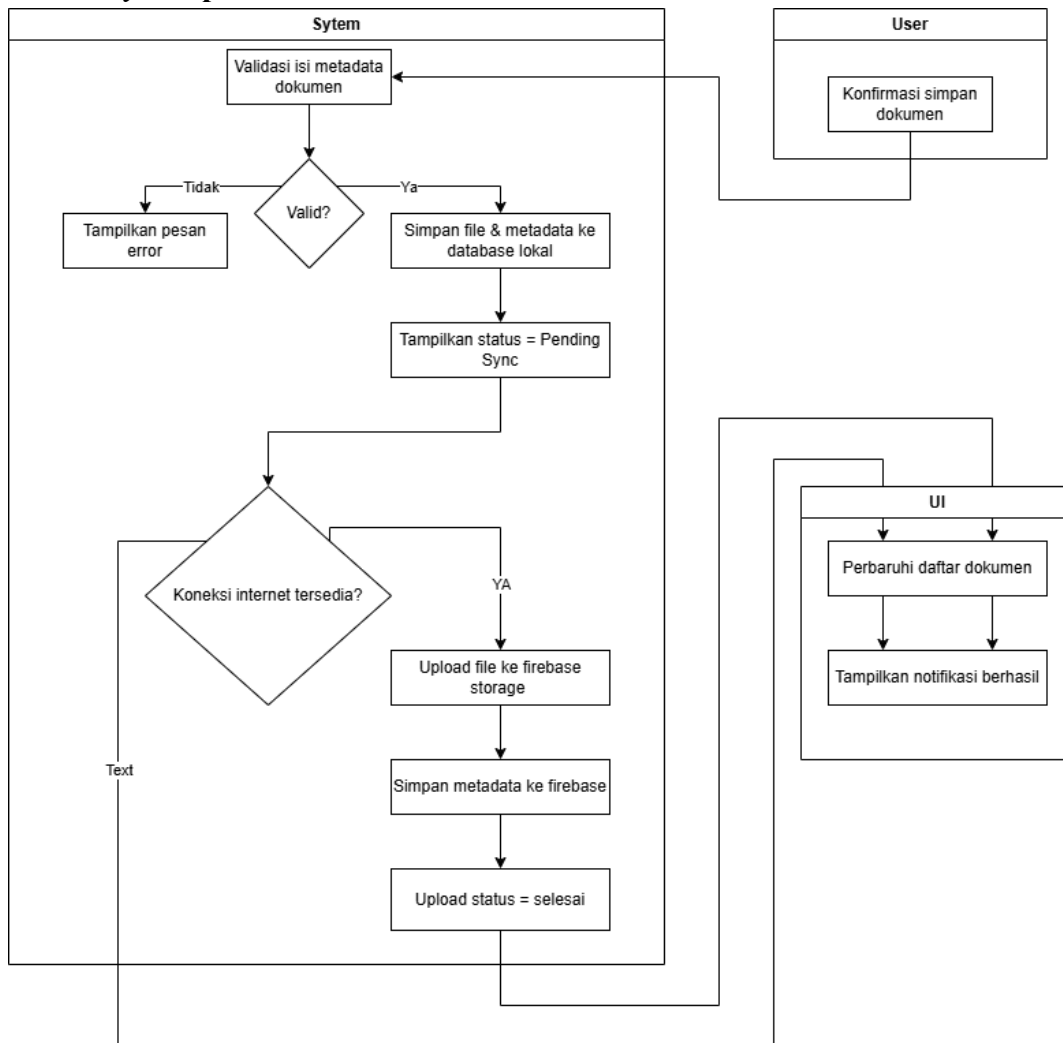
11. Activity Diagram Hapus Dokumen



Gambar 3. 15 Activity Diagram Hapus Dokumen

Activity diagram tersebut menggambarkan proses ketika pengguna menghapus dokumen: setelah pengguna memilih dokumen dan menekan tombol hapus, sistem meminta konfirmasi. Jika pengguna membatalkan, proses berhenti; jika setuju, sistem menghapus data dari *database* lokal dan menandainya sebagai “pending delete”. Selanjutnya sistem akan memeriksa koneksi internet-jika *online*, file dihapus di *Firebase* Storage, metadata dihapus dari *Firebase*, dan status ditandai selesai. Terakhir, *UI* memperbarui daftar dokumen dan menampilkan notifikasi bahwa penghapusan berhasil

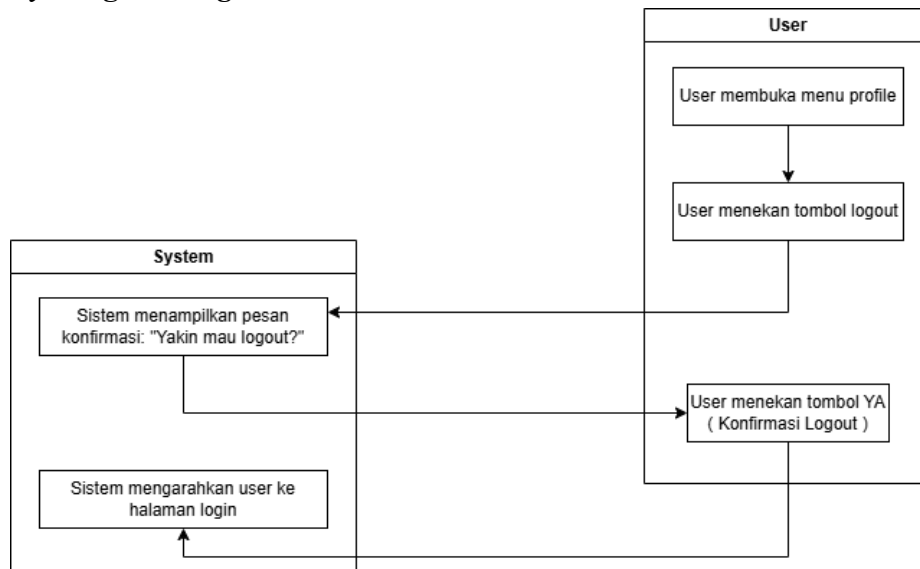
12. Activity Simpan Dokumen



Gambar 3. 16 Activity Simpan Dokumen

Activity diagram tersebut menggambarkan alur simpan dokumen, dimulai ketika *user* mengkonfirmasi penyimpanan dokumen, lalu sistem melakukan validasi metadata. Jika metadata tidak valid, sistem akan menampilkan pesan error; jika valid, file dan metadata disimpan ke *database* lokal dan diberi status *pending sync*. Setelah itu, sistem memeriksa koneksi internet: ketika tidak ada koneksi, proses berhenti sampai koneksi internet tersedia; ketika ada, sistem mengunggah file ke *Firebase Storage*, menyimpan metadata ke *Firebase*, dan memperbarui status menjadi selesai. Terakhir, *UI* memperbarui daftar dokumen dan menampilkan notifikasi berhasil.

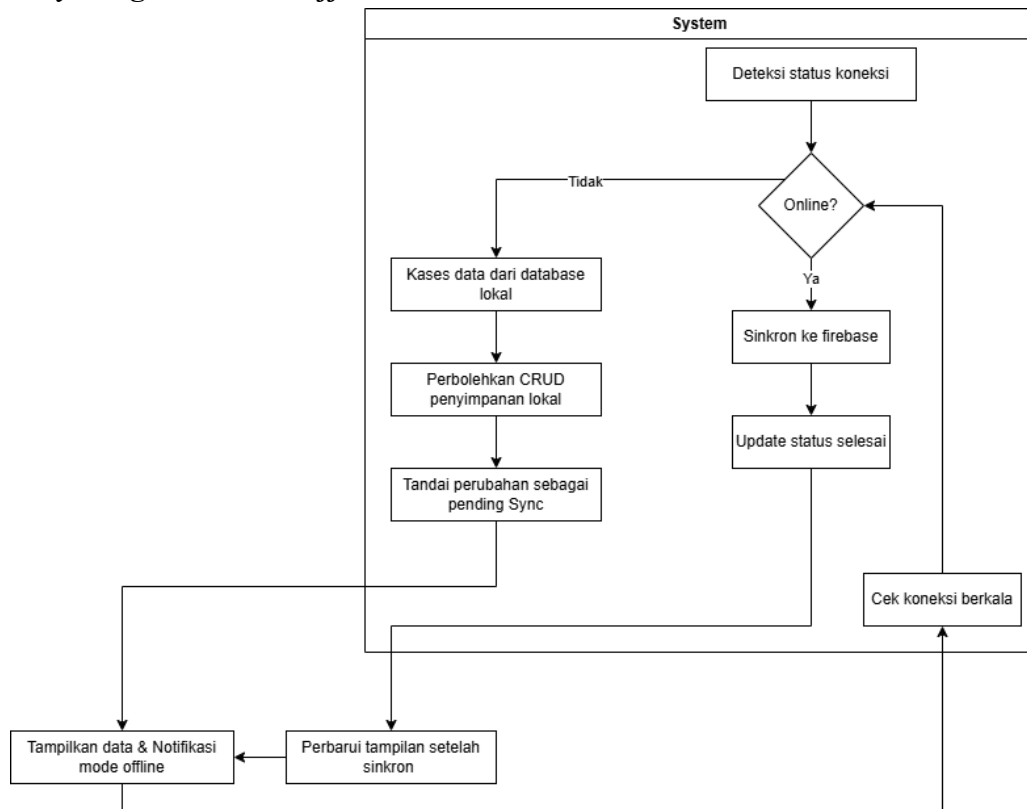
13. Activity Diagram Logout



Gambar 3. 17 Activity Diagram Logout

Activity diagram tersebut menjelaskan alur ketika pengguna ingin melakukan *logout*. Dimulai ketika pengguna membuka menu profil dan menekan tombol *logout*, sistem kemudian menampilkan pesan konfirmasi untuk memastikan apakah pengguna benar – benar ingin keluar. Jika pengguna menekan tombol “Ya” untuk mengkonfirmasi, sistem memproses permintaan tersebut dan mengarahkan pengguna kembali ke halaman *login*.

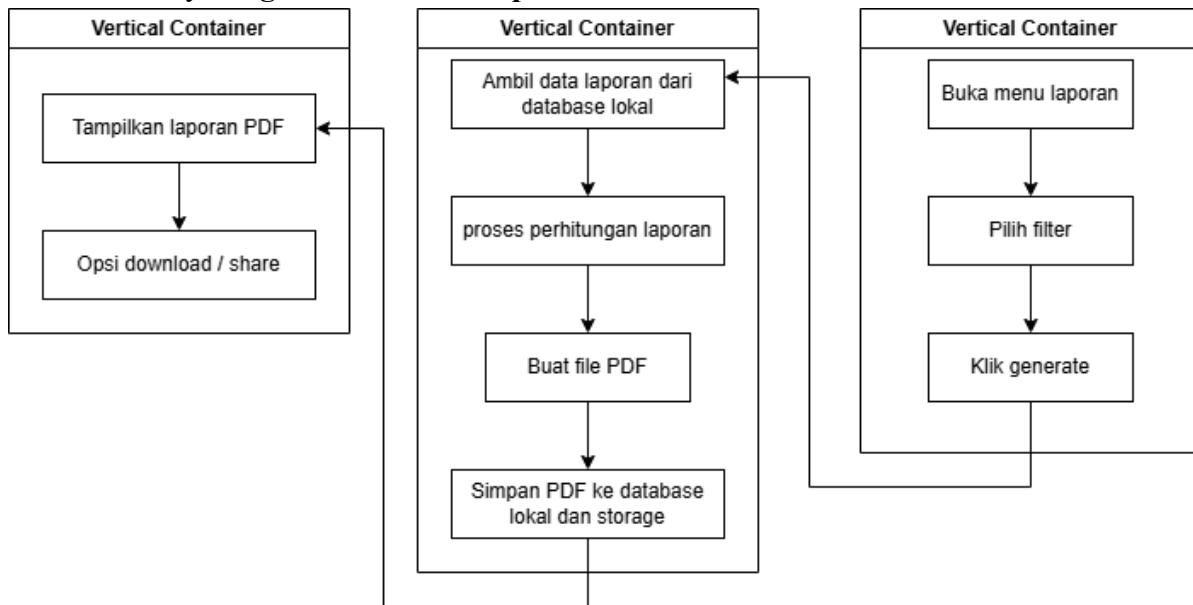
14. Activity Diagram Akses Offline



Gambar 3. 18 Activity Diagram Akses Offline

Activity diagram tersebut menggambarkan bagaimana sistem menangani mode *online* dan *offline*. Ketika sistem mendeteksi status koneksi, jika perangkat *offline* maka data diambil dari *database* lokal, pengguna dapat melakukan CRUD secara lokal, dan setiap perubahan ditandai sebagai *pending sync*, sementara *UI* menampilkan data beserta notifikasi mode *offline*. Jika koneksi tersedia, sistem melakukan sinkronisasi data ke *Firebase*, memperbarui status menjadi selesai, dan *UI* memperbarui tampilan setelah sinkronisasi. Sistem kemudian terus memeriksa koneksi secara berkala untuk memastikan data tetap sinkron dengan server.

15. Activity Diagram Generate Laporan PDF



Gambar 3. 19 Activity Diagram Generate Laporan PDF

Activity diagram tersebut menggambarkan proses pembuatan laporan dimulai ketika pengguna membuka menu laporan, memilih periode atau filter, lalu menekan tombol *Generate*. Sistem kemudian mengambil data laporan dari *database* lokal, melakukan perhitungan yang diperlukan, dan menghasilkan file PDF. Setelah file selesai dibuat, sistem menyimpannya ke *database* lokal atau storage yang digunakan. Terakhir, *UI* menampilkan laporan PDF tersebut kepada pengguna dan menyediakan opsi mengunduh dan membagikannya.

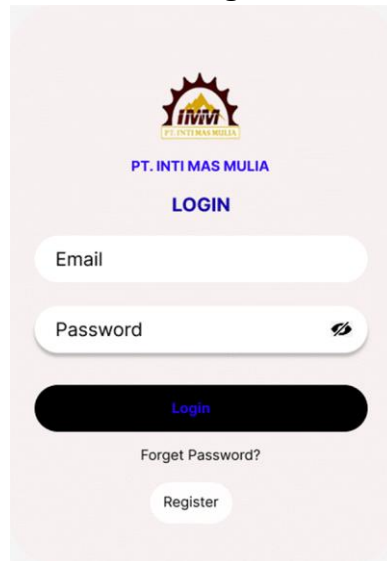
3.2.6 Penjelasan Dari Hasil Penelitian

Dari hasil penelitian ini merancang aplikasi penyimpanan keuangan dan penyimpanan dokumen. Penelitian ini membantu perusahaan yang mengalami kesulitan dalam melakukan penyimpanan keuangan dan penyimpanan dokumen, dimana perusahaan tersebut masih menggunakan manual yaitu dengan mencatat transaksi – transaksi keuangan secara manual menggunakan *excel* maupun *spreadsheet*, serta menyimpan dokumen secara fisik dan di dalam penyimpanan dokumen. Penelitian ini dapat membantu perusahaan yang dimana perusahaan tersebut mengoperasional keluar masuk barang – barang industri serta memperbaiki mesin – mesin industri, yang setiap masuk dan keluar barang atau mesin tersebut merupakan barang atau mesin dengan harga yang sangat tinggi. Dengan penelitian ini perusahaan dapat dengan mudah mencatat harga barang atau mesin yang masuk maupun keluar.

3.2.7 Perancangan Antarmuka (*Wireframe*) / Desain Perangkat Keras (*IoT*)

A. Perancangan *UI* Aplikasi Berbasis *Web*

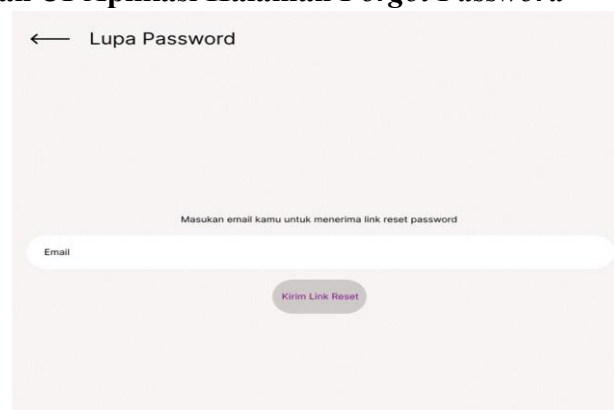
1. Perancangan *UI* Aplikasi Halaman *Login*



Gambar 3. 21 *UI* Aplikasi Halaman *Login*

Gambar diatas merupakan contoh gambar *UI* aplikasi halaman *login* yang dirancang untuk memfasilitasi pengguna dalam mengakses aplikasi dengan memasukan email dan password yang sudah terdaftar. Setelah itu sudah tersedia tombol *login* sebagai fungsi utama, serta opsi pada *forget password* untuk memulihkan akun apabila pengguna lupa dengan password yang terdaftar. Selain itu ada juga tombol *register* yang sudah disediakan untuk pengguna baru yang belum memiliki akun.

2. Perancangan *UI* Aplikasi Halaman *Forgot Password*



Gambar 3. 22 *UI* Aplikasi Halaman *Lupa Password*

Halaman *Forgot Password* dirancang ketika pengguna lupa kata sandi akun. Pada halaman ini, pengguna diminta untuk memasukan *email* yang sudah terdaftar untuk menerima tautan reset *password*. Setelah email dimasukan, pengguna dapat menekan tombol kirim link reset sistem kemudian mengirimkan tautan email yang dapat digunakan untuk membuat kata sandi baru.

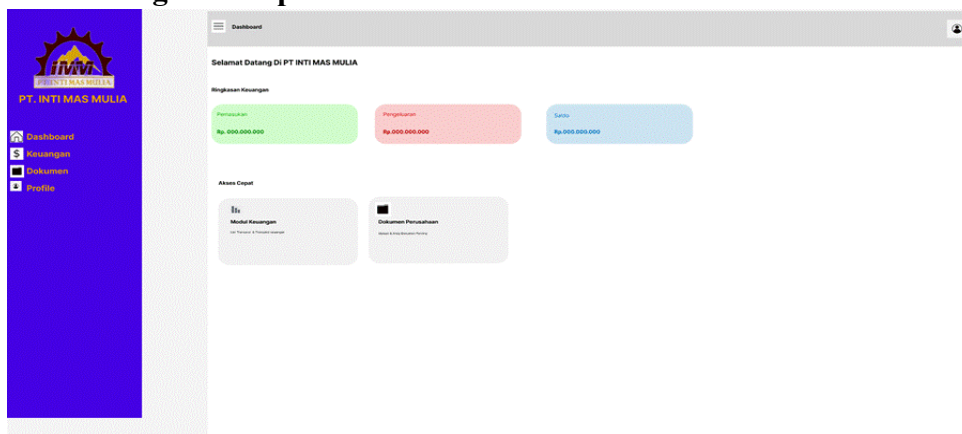
3. Perancangan *UI* Aplikasi Halaman *Register*



Gambar 3. 23UI Register

Gambar diatas merupakan contoh *UI* aplikasi halaman *register* yang dirancang untuk memfasilitasi pengguna baru dalam membuat akun baru. *UI* ini terdiri dari input nama panjang, nama panggilan, tanggal lahir, email, dan password yang sudah di lengkapi dengan ikon-ikon untuk memudahkan pemahaman bagi pengguna. Tombol Daftar berfungsi sebagai tindakan utama sedangkan tautan *login* disediakan bagi pengguna yang sudah memiliki akun. Desain ini dibuat sangat sederhana agar alur pendaftaran lebih mudah dipahami.

4. Perancangan *UI* Aplikasi Halaman *Dashboard* Utama

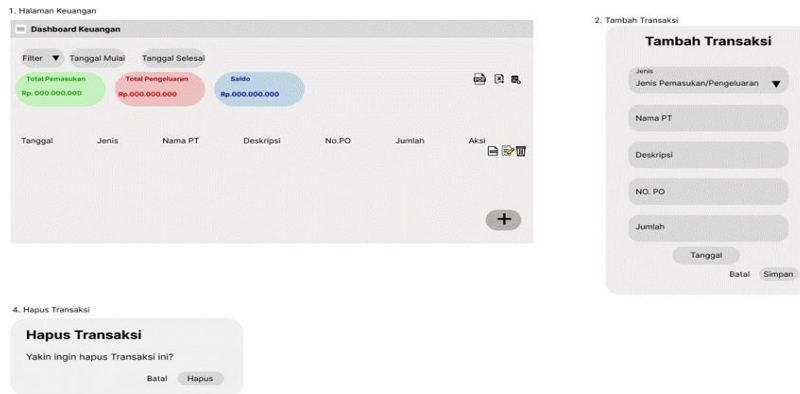


Gambar 3. 24 UI Dashboard Utama

Gambar diatas merupakan *UI dashboard* utama yang dirancang sebagai pusat informasi aplikasi yang menampilkan ringkasan data penting perusahaan. Pada bagian kiri terdapat gambar menu navigasi yang berisikan *dashboard*, keuangan, dokumen, dan profil. Bagian utama *dashboard* menampilkan informasi keuangan yang berupa pemasukan,

pengeluaran, dan saldo dalam bentuk berwarna agar pengguna dapat dengan mudah mengidentifikasi. Selain itu disediakan juga menu akses cepat untuk mempercepat penggunaan dalam mengakses fitur penting.

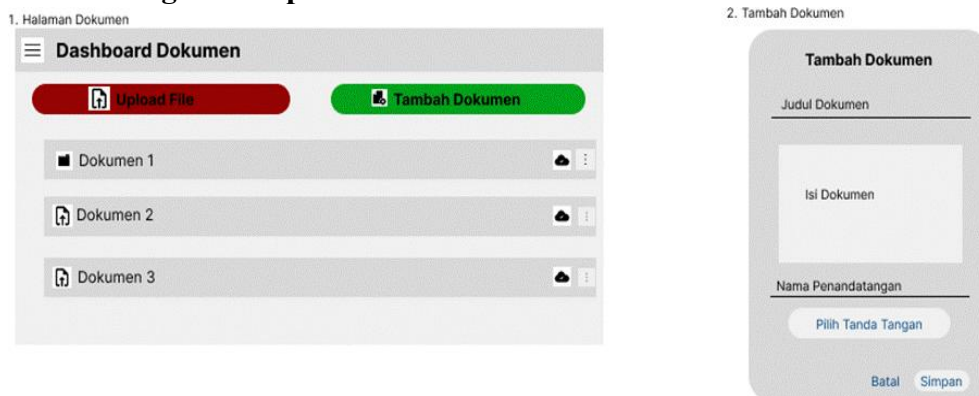
5. Perancangan *UI* Aplikasi Halaman Keuangan, Tambah Transaksi, Hapus Transaksi



Gambar 3. 25 *UI Halaman Keuangan, Tambah dan Hapus Transaksi*

Gambar *UI* diatas merupakan rancangan aplikasi keuangan untuk mencatat dan menampilkan data transaksi keuangan. Pada bagian utama, sistem menampilkan ringkasan berupa total pemasukan, total pengeluaran, dan saldo. Data transaksi juga menampilkan dalam bentuk tabel dengan kolom tanggal, jenis, nama pt, deskripsi, nomor PO, jumlah dan aksi. Rancangan tersebut memiliki fitur tambahan yaitu tambah transaksi, pengguna dapat mengisi data transaksi seperti jenis transaksi, nama pt, deskripsi, nomor po, dan jumlah. Selain itu terdapat fitur hapus transaksi dengan mengkonfirmasi terlebih dahulu agar tidak terjadi kesalahan pada penghapusan data.

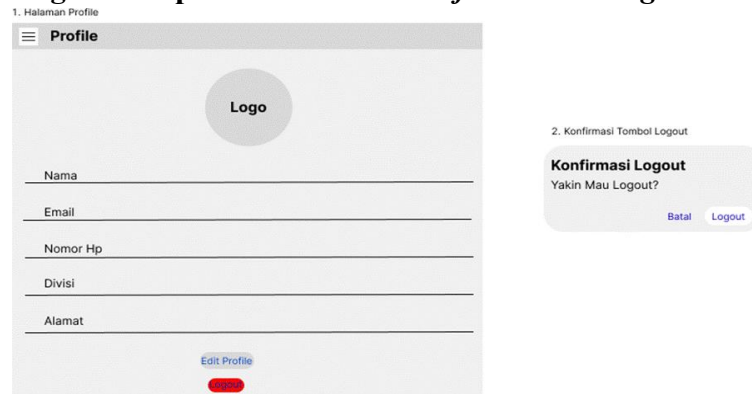
6. Perancangan *UI* Aplikasi Halaman Dokumen Beserta Tambah Dokumen



Gambar 3. 26 *UI Halaman dan Tambah Dokumen*

Gambar diatas merupakan rancangan *UI* aplikasi halaman dokumen beserta tambah dokumen. Halaman dokumen ini dirancang untuk memudahkan pengguna dapat mengelola arsip digital, pada halaman dokumen ini tersedia fitur upload file dan tambah dokumen yang memungkinkan pengguna mengunggah atau membuat dokumen baru. Daftar dokumen ditampilkan secara struktur sehingga mudah diakses kembali. Fitur tambah dokumen menyediakan form untuk menginput judul dokumen, isi dokumen, serta nama penandatanganan, ditambahkan juga opsi tambah tanda tangan digital. Tersedia tombol simpan untuk menyimpan dokumen dan tombol batal jika pengguna tidak ingin melanjutkan prosesnya.

7. Perancangan *UI* Aplikasi Halaman *Profile* Beserta *logout*



Gambar 3. 27 *UI Profile dan Logout*

Halaman profile dirancang untuk menampilkan informasi data diri pengguna yang meliputi nama, email, nomor hp, divisi, dan alamat. Tersedia tombol edit yang memungkinkan pengguna memperbarui data diri pengguna sesuai dengan kebutuhan. Selain itu, halaman ini juga menyediakan tombol *logout* untuk keluar dari aplikasi. Saat tombol *logout* ditekan sistem akan menampilkan konfirmasi *logout* dengan opsi batal dan *logout* agar tidak terjadi kesalahan dalam proses keluar aplikasi.

B. Perancangan *UI* Aplikasi Berbasis Mobile

1. Desain Halaman *Register*



Gambar 3. 28 *Register*

Gambar diatas merupakan contoh *UI* aplikasi halaman *register* yang dirancang untuk memfasilitasi pengguna baru dalam membuat akun baru. *UI* ini terdiri dari input nama

panjang, nama panggilan, tanggal lahir, email, dan password yang sudah di lengkapi dengan ikon-ikon untuk memudahkan pemahaman bagi pengguna. Tombol Daftar berfungsi sebagai tindakan utama, sedangkan tautan *login* disediakan bagi pengguna yang sudah memiliki akun. Desain ini buat sederhana dan responsif di semua ukuran layar hp sangat mudah dipahami oleh pengguna.

2. Desain Halaman *Login*



Gambar 3. 29 Aplikasi Mobile Halaman Login

Gambar diatas merupakan contoh gambar UI aplikasi halaman *login* yang dirancang untuk memfasilitasi pengguna dalam mengakses aplikasi dengan memasukan email dan password yang sudah terdaftar. Setelah itu sudah tersedia tombol *login* sebagai fungsi utama, serta opsi pada forget password untuk memulihkan akun apabila pengguna lupa dengan password yang terdaftar. Selain itu ada juga tombol *register* yang sudah disediakan untuk pengguna baru yang belum memiliki akun.

3. Desain Halaman *Forgot Password*



Gambar 3. 30 Lupa Password

Halaman *Forgot Password* dirancang ketika pengguna lupa kata sandi akun. Pada halaman ini, pengguna diminta untuk memasukan email yang sudah terdaftar untuk menerima

tautan reset password. Setelah email dimasukan, pengguna dapat menekan tombol kirim link reset sistem kemudian mengirimkan tautan email yang dapat digunakan untuk membuat kata sandi baru.

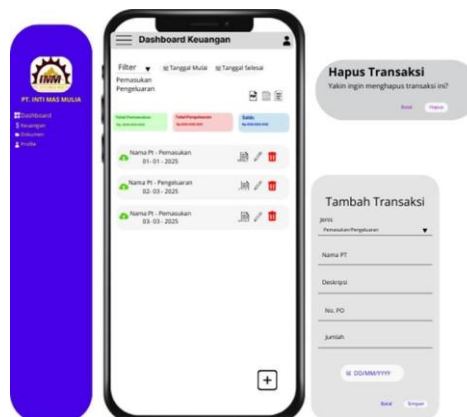
4. Desain Halaman *Dashboard* Utama



Gambar 3. 31 *Dashboard* Utama

Gambar diatas merupakan *UI dashboard* utama yang dirancang sebagai pusat informasi aplikasi yang menampilkan ringkasan data penting perusahaan. Pada bagian kiri terdapat gambar menu navigasi yang berisikan *dashboard*, keuangan, dokumen, dan profil. Bagian utama *dashboard* menampilkan informasi keuangan yang berupa pemasukan, pengeluaran, dan saldo dalam bentuk berwarna agar pengguna dapat dengan mudah mengidentifikasi. Selain itu disediakan juga menu akses cepat untuk mempercepat penggunaan dalam mengakses fitur penting

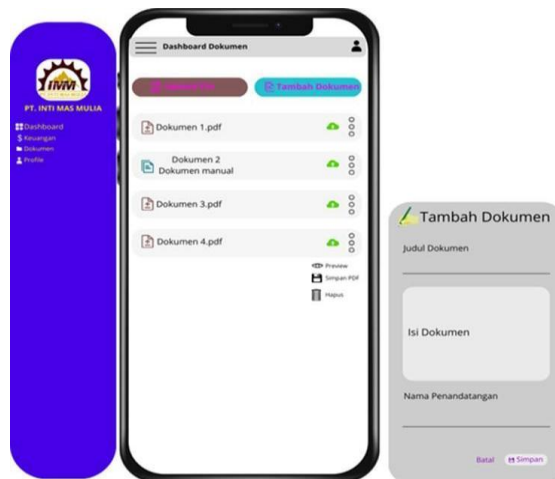
5. Desain Halaman *Dashboard* Keuangan Beserta Tambah Transaksi Dan Hapus Transaksi



Gambar 3. 32 *Dashboard* Keuangan, *Transaksi* Data dan *Hapus Transaksi*

Pada halaman utama ditampilkan daftar-daftar transaksi pemasukan dan pengeluaran yang dapat di filter berdasarkan tanggal dan jenis transaksi. Pengguna juga dapat menambahkan data transaksi baru melalui form input yang mencakup jenis transaksi, nama pt, deskripsi, nomor po, dan jumlah. Selain itu disediakan juga fitur untuk menghapus data transaksi yang dilengkapi dengan konfirmasi agar tidak terjadi kesalahan dalam penghapusan data transaksi.

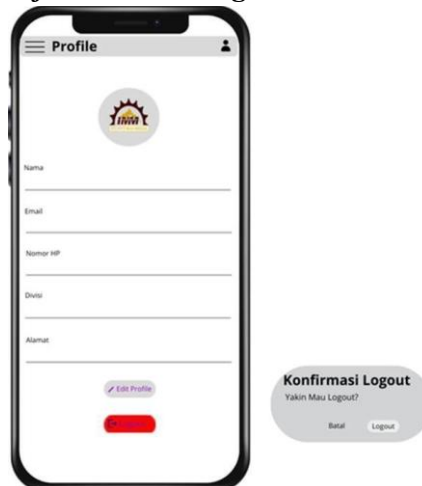
6. Desain Halaman *Dashboard* Dokumen Beserta Tambah Dokumen



Gambar 3. 33 Dashboard dan Tambah Dokumen

Halaman ini berfungsi untuk mengelola dokumen perusahaan, baik dengan cara mengupload dokumen serta menambahkan dokumen secara manual. Daftar dokumen ditampilkan dalam bentuk list, lengkap dengan opsi pemilihan preview, simpan pdf, dan hapus. Selain itu, tersedia juga fitur tambah dokumen yang memungkinkan pengguna mengisi judul dokumen, isi dokumen, dan nama penandatanganan. Desain ini dibuat agar pengguna dapat mengelola dokumen menjadi lebih rapi, mudah diakses dan mendukung efisiensi kerja dalam menyimpan dan mencari dokumen penting.

7. Desain Halaman *Profile* Beserta *Logout*



Gambar 3. 34 Halaman Profil dan Logout

Halaman *profile* dirancang untuk menampilkan informasi data diri pengguna yang meliputi nama, email, nomor hp, divisi, dan alamat. Tersedia tombol edit yang memungkinkan pengguna memperbarui data diri pengguna sesuai dengan kebutuhan. Selain itu, halaman ini juga menyediakan tombol *logout* untuk keluar dari aplikasi. Saat tombol *logout* ditekan sistem akan menampilkan konfirmasi *logout* dengan opsi batal dan *logout* agar tidak terjadi kesalahan dalam proses keluar aplikasi.

3.3 Pertimbangan Etika, Hukum, Keselamatan, dan Dampak Sosial

Dalam pengembangan aplikasi ini, aspek etika diperhatikan dengan menjaga kerahasiaan data pengguna dan mencegah penyalahgunaan informasi. Dari sisi hukum, sistem dikembangkan sesuai dengan peraturan yang berlaku serta menggunakan teknologi yang legal. Dari sisi keselamatan, sistem dilengkapi dengan mekanisme keamanan untuk melindungi data dari kehilangan maupun kebocoran. Dari sisi dampak sosial, aplikasi ini diharapkan dapat meningkatkan efisiensi kerja dan transparansi, meskipun pengguna perlu beradaptasi dengan penggunaan sistem digital.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Hasil Implementasi

Pada bab ini dijelaskan hasil implementasi dari rancang bangun aplikasi keuangan dan penyimpanan dokumen berbasis *web* dan *mobile* menggunakan *Flutter* dan *Firebase*. Sistem ini dibangun untuk mendukung kebutuhan PT Inti Mas Mulia dalam mengelola transaksi keuangan dan dokumen internal perusahaan secara terintegrasi. Rancangan yang telah dibahas pada bab sebelumnya digunakan untuk implementasi. Ini mencakup proses instalasi, konfigurasi layanan *Firebase*, *integrasi autentikasi*, dan penerapan penyimpanan dokumen berbasis *cloud* melalui *browser* dan *Chrome*.

4.1.1 Proses Implementasi

Proses implementasi dilakukan melalui beberapa tahap sebagai berikut:

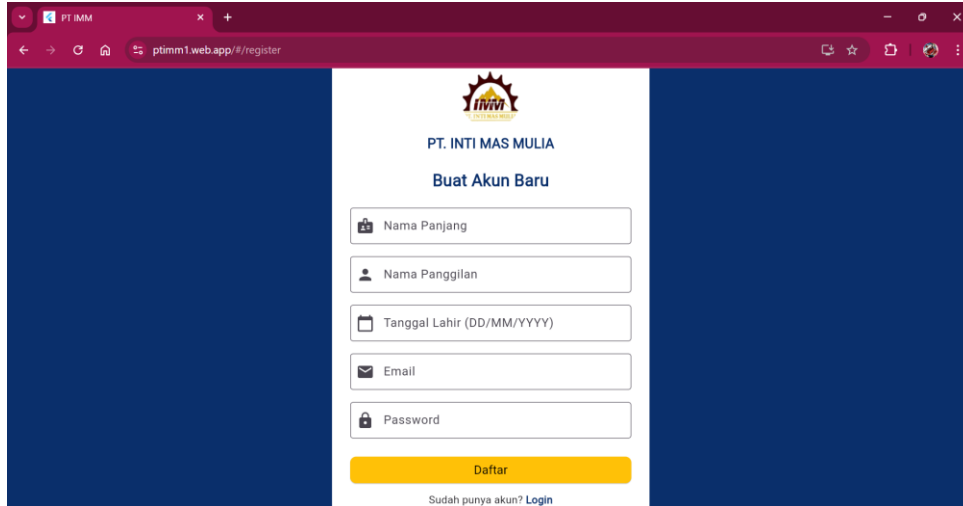
1. Instalasi dan konfigurasi environment
 - a. Instalasi *Flutter SDK* untuk pengembangan *web* dan *mobile*.
 - b. Visual Studio Code sebagai editor pengembangan.
 - c. Konfigurasi emulator android serta uji coba aplikasi pada perangkat fisik.
2. Konfigurasi *Firebase*
 - a. Integrasi *Firebase Authentication* untuk proses registrasi dan *login* pengguna.
 - b. integrasi *Cloud Firestore* Sebagai basis data utama untuk menyimpan data transaksi, metadata dokumen, dan data profil pengguna.
 - c. Konfigurasi keamanan melalui *Firebase Rules* untuk memastikan akses data hanya dapat dilakukan oleh pengguna yang berwenang.
3. Implementasi Penyimpanan Dokumen
 - a. Penyimpanan file tidak menggunakan *firebase storage*, melainkan memanfaatkan *browser/chrome cloud storage*.
 - b. File diunggah disimpan melalui mekanisme *storage* bawaan *browser* dan disinkronkan melalui *Chrome*.
 - c. Metadata file tetap disimpan di *Firestore*, berisi: nama file, deskripsi, ukuran, waktu unggahan, dan ID pemilik.
4. Implementasi Antarmuka *Web* dan *Mobile*
 - a. Antarmuka dibuat menggunakan *flutter* dengan pendekatan *one-codebase* untuk dua platform.
 - b. Penerapan widget responsive untuk menyesuaikan tampilan pada *mobile* dan *browser*.
 - c. Pemisahan modul keuangan dan dokumen agar mudah digunakan oleh pengguna
5. *Deployment* Sistem
 - a. Aplikasi *web* di-deploy menggunakan *Firebase Hosting*.
 - b. Aplikasi *mobile* diuji pada beberapa perangkat android untuk memastikan kompatibilitas.

4.1.2 Tampilan Hasil Implementasi

Berikut adalah tampilan antarmuka hasil implementasi aplikasi:

A. Tampilan Halaman *Web*

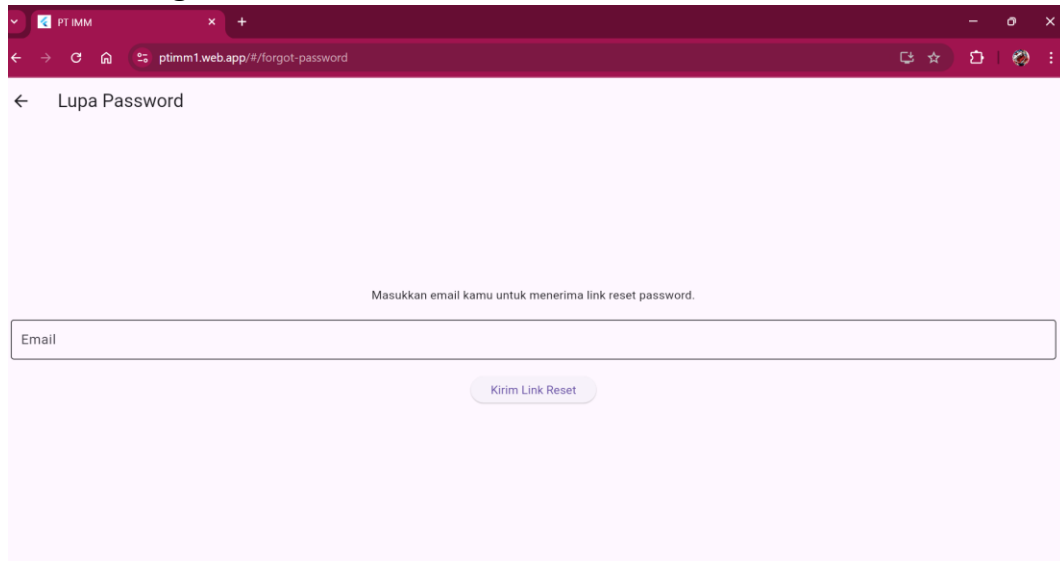
1. Halaman *Register*



The screenshot shows a web browser window with the URL `ptimm1.web.app/#/register`. The page features a dark blue sidebar on the left and a white main content area. At the top of the white area is the PT. INTI MAS MULIA logo and the text "PT. INTI MAS MULIA" and "Buat Akun Baru". Below this, there are five input fields: "Nama Panjang", "Nama Panggilan", "Tanggal Lahir (DD/MM/YYYY)", "Email", and "Password". Each field has a corresponding icon (ID card, person, calendar, envelope, and lock respectively). A yellow "Daftar" button is positioned below the "Password" field. At the bottom of the white area, there is a link that says "Sudah punya akun? Login".

Halaman *register* digunakan oleh pengguna untuk membuat akun baru pada aplikasi.

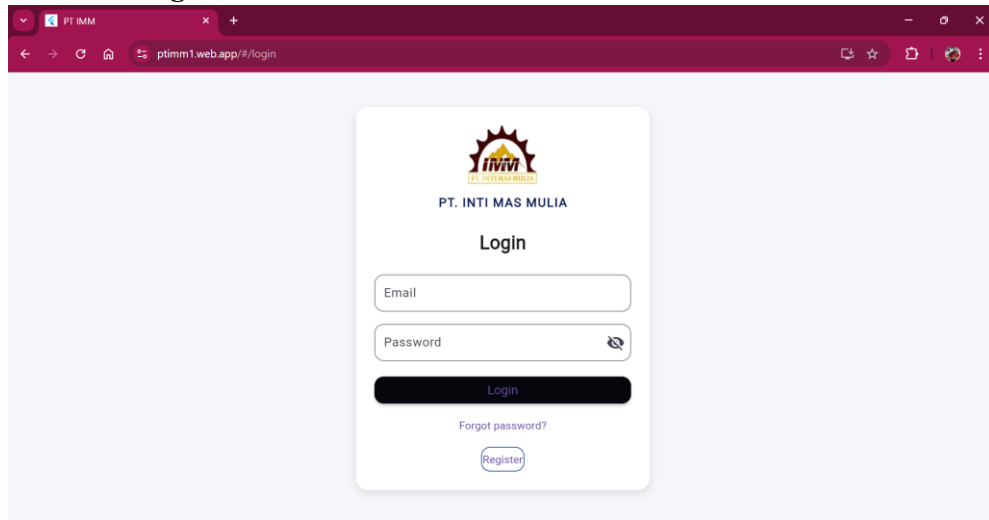
2. Halaman *Forgot Password*



The screenshot shows a web browser window with the URL `ptimm1.web.app/#/forgot-password`. The page has a light purple background. At the top left, there is a back arrow and the text "Lupa Password". In the center, there is a message: "Masukkan email kamu untuk menerima link reset password." Below this message is a single input field labeled "Email". At the bottom center, there is a button labeled "Kirim Link Reset".

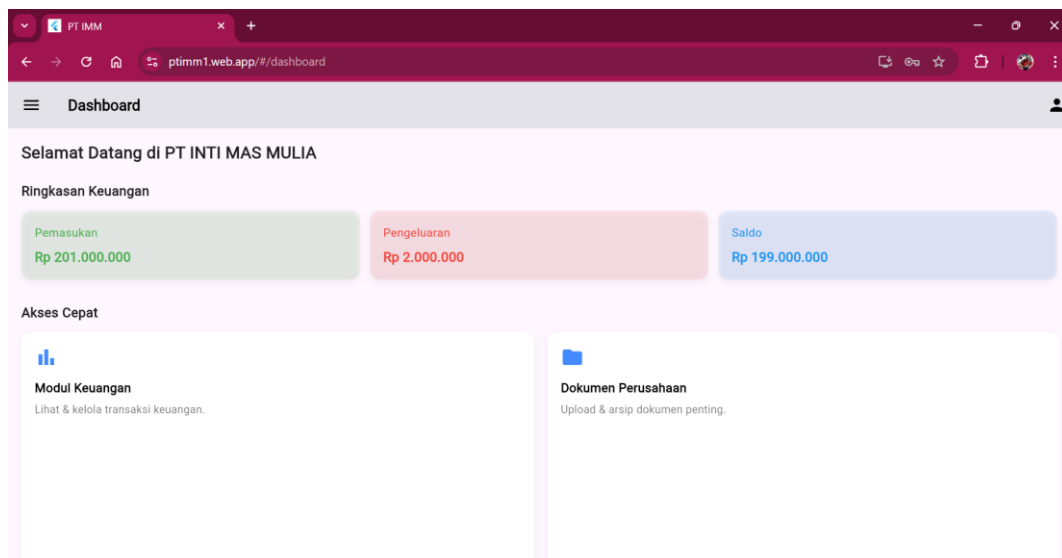
Halaman tersebut digunakan oleh pengguna ketiga pengguna lupa dengan kata sandi yang sudah dibuat.

3. Halaman *Login*



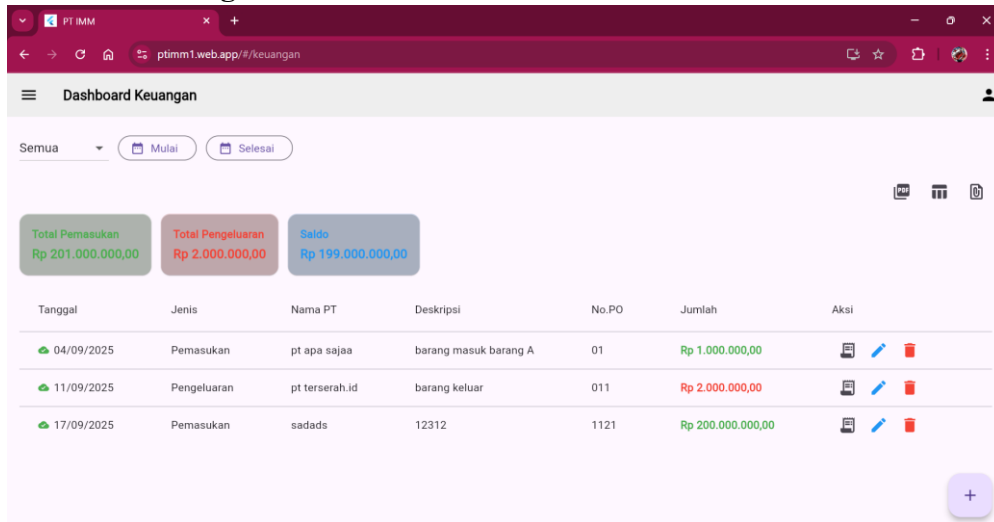
Halaman *login* digunakan oleh pengguna untuk masuk ke dalam aplikasi

4. Halaman Utama



Halaman utama ini berisikan fitur - fitur seperti akses cepat ke halaman keuangan dan kelemahan dokumen serta menampilkan total saldo keluar, masuk dan total keseluruhan saldo serta ada menu pop-up

5. Halaman Keuangan



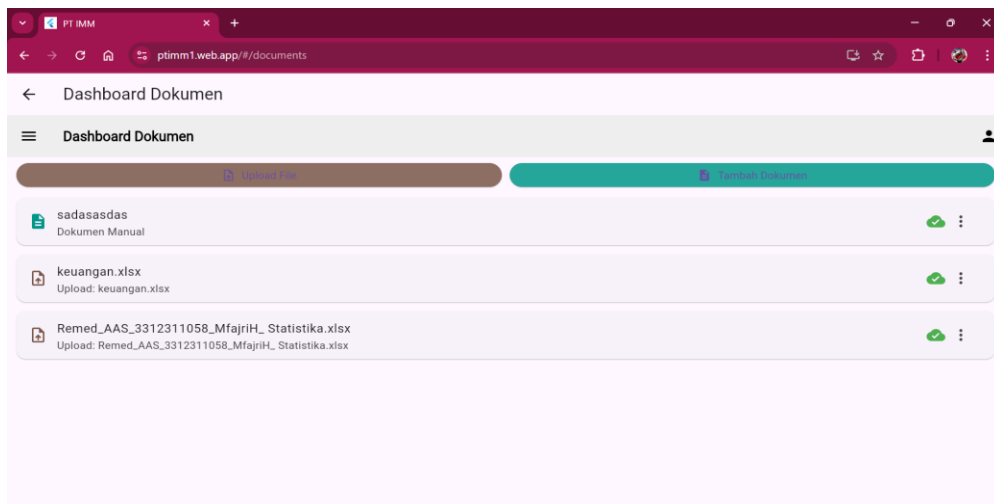
The screenshot displays the 'Dashboard Keuangan' interface. At the top, there are filters for 'Semua' (All), 'Mulai' (Start), and 'Selesai' (End). Below these are three summary cards: 'Total Pemasukan' (Total Revenue) at Rp 201.000.000,00, 'Total Pengeluaran' (Total Expenditure) at Rp 2.000.000,00, and 'Saldo' (Balance) at Rp 199.000.000,00. The main section is a table of transactions with columns for Tanggal (Date), Jenis (Type), Nama PT (Company Name), Deskripsi (Description), No. PO (PO Number), Jumlah (Amount), and Aksi (Action).

Tanggal	Jenis	Nama PT	Deskripsi	No. PO	Jumlah	Aksi
04/09/2025	Pemasukan	pt apa saja	barang masuk barang A	01	Rp 1.000.000,00	[Icon: Print, Edit, Delete]
11/09/2025	Pengeluaran	pt terserah.id	barang keluar	011	Rp 2.000.000,00	[Icon: Print, Edit, Delete]
17/09/2025	Pemasukan	sadads	12312	1121	Rp 200.000.000,00	[Icon: Print, Edit, Delete]

A '+ ' button is located at the bottom right of the table area.

Halaman keuangan ini digunakan pengguna untuk merekap laporan keuangan perusahaan, setiap laporan yang ditambahkan bisa dilihat, diedit dan dihapus, serta ada fitur export PDF, EXCEL, CSV.

6. Halaman Dokumen

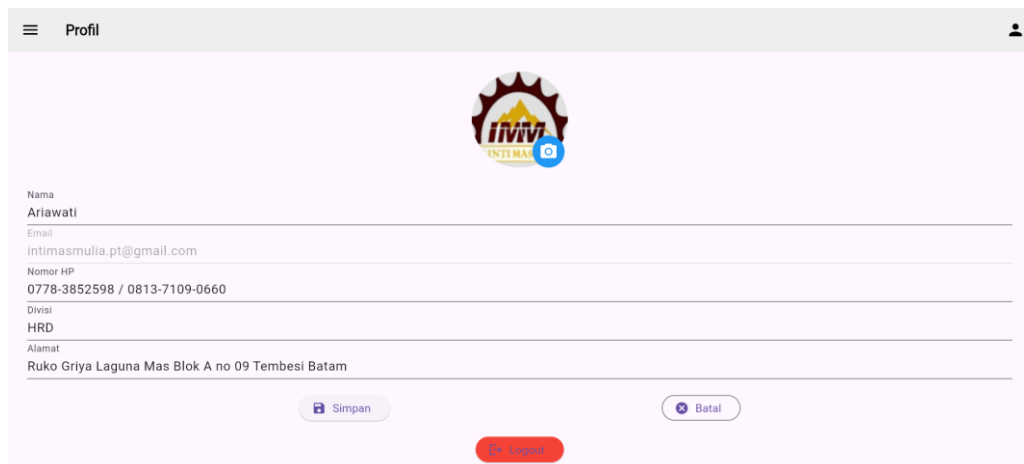


The screenshot shows the 'Dashboard Dokumen' interface. It features two tabs: 'Dokumen PDF' (selected) and 'Dokumen Dokumen'. Below the tabs is a list of documents, each with a document icon, a name, a description, and a status icon (green checkmark) with a three-dot menu.

Dokumen PDF	Dokumen Dokumen
sadasdasdas Dokumen Manual	
keuangan.xlsx Upload: keuangan.xlsx	
Remed_AAS_3312311058_MfajriH_Statistika.xlsx Upload: Remed_AAS_3312311058_MfajriH_Statistika.xlsx	

Halaman dokumen ini digunakan pengguna untuk menyimpan, mengupload serta menambahkan dokumen - dokumen penting perusahaan yang dapat di edit, hapus, dan juga disimpan kedalam bentuk PDF.

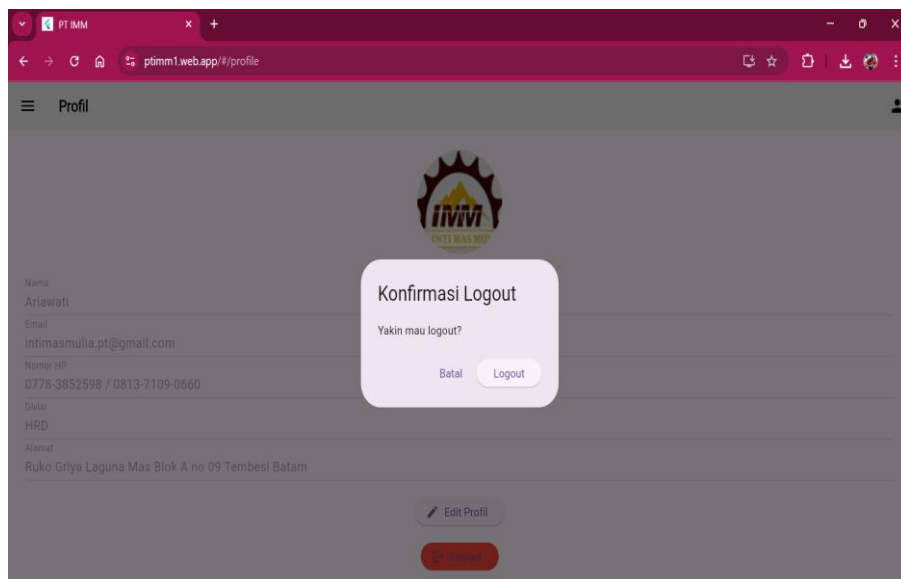
7. Halaman Profile



The screenshot shows a web application interface for a user profile. At the top, there is a header bar with a hamburger menu icon on the left, the word "Profil" in the center, and a user profile icon on the right. Below the header, the user's profile information is displayed. At the top of the profile section is a circular profile picture placeholder with a gear icon and a camera icon. Below this, the following fields are listed: "Nama" (Ariawati), "Email" (intimasmulia.pt@gmail.com), "Nomor HP" (0778-3852598 / 0813-7109-0660), "Divisi" (HRD), and "Alamat" (Ruko Griya Laguna Mas Blok A no 09 Tembesi Batam). At the bottom of the profile section, there are three buttons: "Simpan" (Save), "Batal" (Cancel), and "Edit Profil" (Edit Profile).

Halaman profile ini digunakan pengguna untuk menambahkan informasi data diri pengguna dengan menggunakan nama, email, nomor hp, divisi, serta alamat perusahaan.

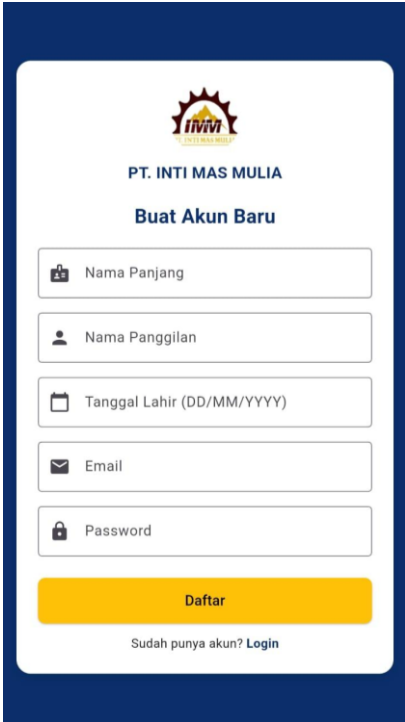
8. Logout



Ketika pengguna ingin mengeluarkan akun pengguna bisa mengklik tombol *logout* yang sudah ada di halaman *login*.

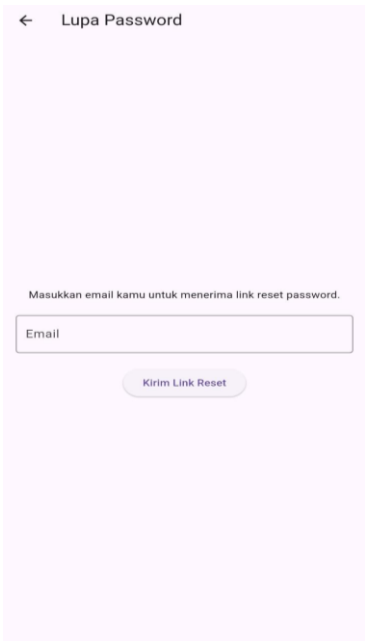
B. Tampilan Halaman Android

1. Halaman *Register*



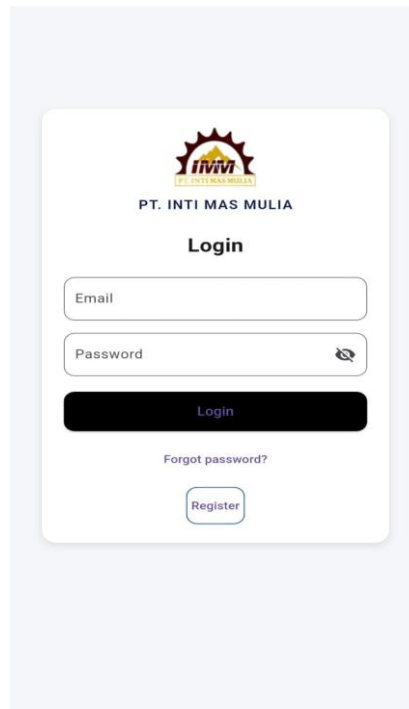
Pada Halaman Registrasi *user* dapat membuat akun pada halaman tersebut. *User* dapat mengisi nama panjang, nama panggilan, tanggal lahir, email, dan password. Setelah itu, *user* bisa mengklik daftar sistem akan memvalidasi data tersebut dan akan disimpan. Apabila *user* sudah memiliki akun *user* bisa mengklik *login* yang sudah ada dibawahnya sistem otomatis mengalihkan *user* pada halaman *login*.

2. Halaman *Forgot Password*



Pada Halaman *forgot password* ketika *user* lupa password *user* dapat melakukan verifikasi untuk mengatur ulang password tersebut. Caranya *user* dapat mengisi sebuah email yang sudah terdaftar pada akun yang lupa password tersebut kemudian sistem akan memberikan sebuah notifikasi email ke *user* setelah itu *user* bisa langsung mengklik link yang sudah dikirim ke email yang terdaftar, setelah itu *user* dapat mengganti password tersebut.

3. Halaman Login



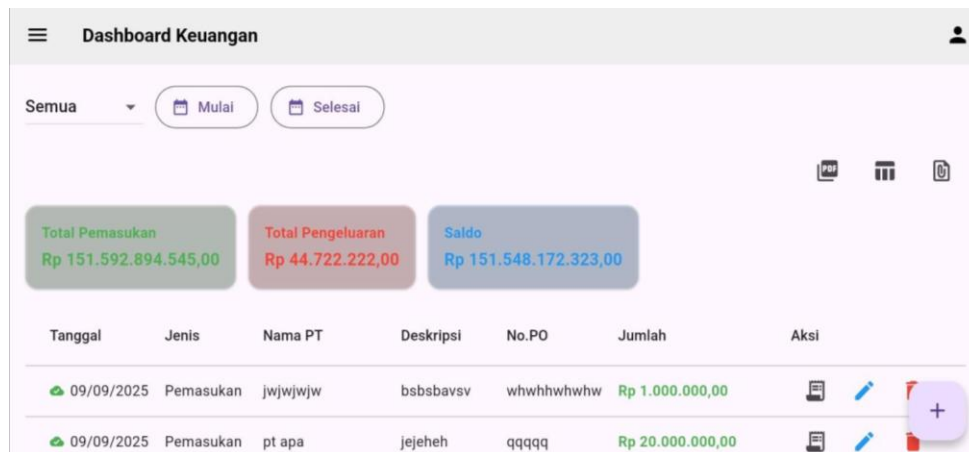
Halaman *Login user* dapat melakukan *login* dengan memasukkan email dan password yang sudah terdaftar. Pada halaman tersebut ketika *user* belum memiliki akun *user* dapat mengklik tombol *register* setelah itu sistem akan mengarahkan *user* pada halaman *register*, ketika *user* lupa dengan password yang sudah terdaftar *user* dapat mengklik tombol *forgot password* sistem akan mengarahkan *user* ke halaman *forgot password*.

4. Halaman Utama

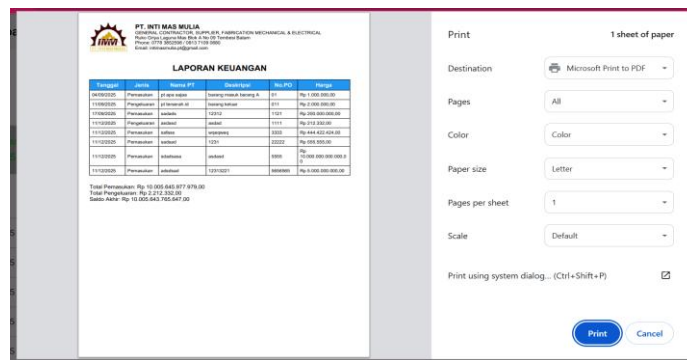


Pada halaman utama *user* dapat melihat sedikit ringkasan keuangan yang tertera di halaman utama yaitu tentang saldo keuangan dari pemasukan, pengeluaran, dan total saldo. Setelah itu *user* dapat melihat fitur akses cepat dimana fitur tersebut dapat mengarahkan langsung ke dalam halaman keuangan dan dokumen perusahaan. Tombol menu di samping berisikan fitur menu *dashboard*, keuangan, dokumen, dan profil. Disebelah kanan ada *icon profile user* bisa dapat langsung mengklik tombol *icon* tersebut dan tidak perlu untuk mengklik menu untuk beralih ke halaman *profile*.

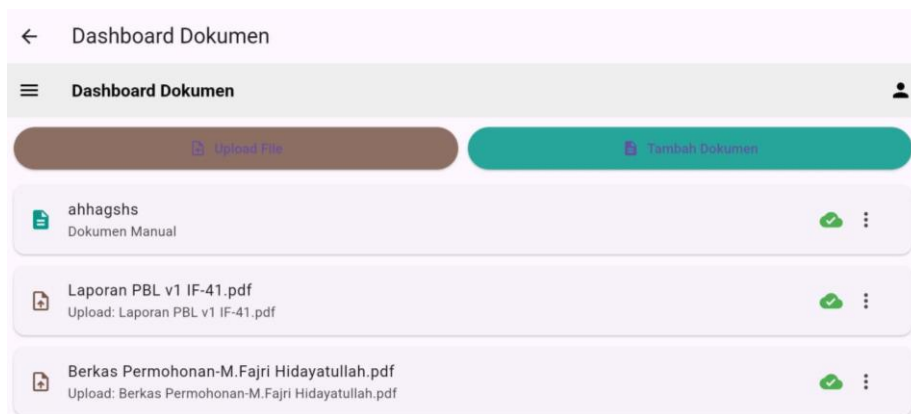
5. Halaman Keuangan



Halaman keuangan tersebut berisikan sebuah data - data keuangan, total pemasukan, pengeluaran, dan total saldo, fitur *export PDF*, excel, dan CSV, serta ada fitur lihat invoice, edit, hapus dan tambah pada tabel data keuangan. Ketika *user* ingin menambahkan data keuangan *user* bisa mengklik *icon* tambah yang ada dibawah, ketika *user* ingin melihat data keuangan per *invoice user* dapat mengklik salah satu data keuangan tersebut untuk melihat dengan cara mengklik *icon invoice*, ketika *user* ingin mengedit data tersebut *user* dapat mengklik *icon* pensil, ketika *user* ingin menghapus salah satu data keuangan tersebut *user* bisa mengklik *icon* hapus setelah itu sistem akan memberikan notifikasi kepada *user* “Yakin ingin menghapus transaksi ini?” *user* bisa mengklik “Iya” jika tidak *user* bisa memilih batal. Untuk *export PDF* *user* bisa mengklik *icon PDF* yang ada di halaman tersebut setelah itu sistem akan langsung menampilkan halaman berupa lembaran *PDF* yang bisa langsung diprint oleh *user* serta dapat langsung menyimpan hasil laporan keuangan tersebut. Berikut adalah halaman *export PDF*:



6. Halaman Dokumen



Pada halaman dokumen tersebut berisikan fitur *export file* serta tambah dokumen dan juga icon menu yang ada tabel dokumen. Jika *user* ingin menambahkan dokumen *user* dapat mengklik tombol tambah dokumen setelah itu sistem akan melampirkan selembar halaman tambah dokumen seperti berikut:

Tambah Dokumen

Judul Dokumen

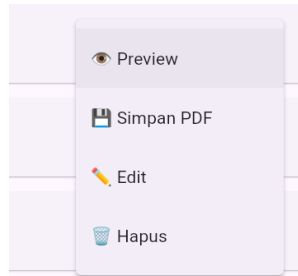
Isi Dokumen

Nama Penandatangan

Pilih Tanda Tangan

Batal Simpan

Setelah itu, *user* dapat mengisi judul dokumen, isi dokumen, nama penandatanganan, dan bisa menambahkan tanda tangan yang berupa gambar. Setelah selesai mengisi lembar tersebut *user* bisa langsung mengklik simpan, sistem akan mengkonfirmasi data dokumen terbaru dan akan menampilkan data tersebut kedalam tabel dokumen di halaman dokumen.

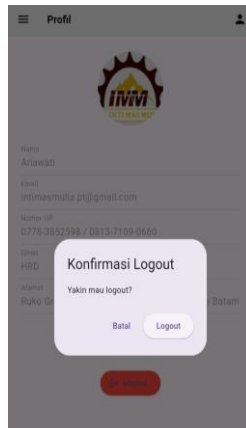


Jika *user* ingin melihat data dokumen yang ada di tabel dokumen tersebut *user* dapat mengklik icon menu yang sudah ada di tabel dokumen tersebut. *User* juga dapat menyimpan data tersebut data tersebut otomatis ke download, *user* juga bisa mengedit serta menghapus dokumen.

7. Halaman Profile



Halaman profile tersebut *user* dapat melihat nama, email, nomor hp, divisi, serta alamat. *user* dapat mengedit profile tersebut dan juga *user* bisa menambahkan foto profil *user*. Setelah itu, *user* dapat menyimpan editan tersebut otomatis sistem akan memperbarui data profile *user* ke dalam *firebase*. Setelah semuanya selesai *user* bisa melakukan *logout* untuk mengakhiri aktivitas yang ada di dalam aplikasi tersebut, *user* bisa mengklik tombol *logout* yang sudah ada di bawah setelah mengklik tombol tersebut akan muncul notifikasi konfirmasi *logout* seperti berikut:



4.2 Pengujian *Black-Box Testing*

Pengujian pada sistem ini dilakukan menggunakan *black-box testing* dengan fokus pada kesesuaian antara input dan output yang dihasilkan sistem. Pengujian ini bertujuan untuk memastikan seluruh fungsi fitur berjalan sesuai dengan kebutuhan tanpa melihat struktur kode pemrograman.

A. Pengujian Fitur *Register*

Tabel 4.1 Pengujian Fitur Register

No	Skenario	Input	Output	Hasil
1	<i>Register</i> benar	Semua data harus diisi dengan lengkap	Berhasil terdaftar	Berhasil
2	<i>User</i> langsung masuk	<i>User</i> diarahkan halaman utama	Berhasil mengalihkan <i>user</i>	Berhasil

Dari hasil pengujian Black Box Testing tersebut, fitur registrasi telah memenuhi kebutuhan fungsional sistem. Pengujian *User Acceptance Testing (UAT)* dilakukan sebagai tahap lanjutan untuk memastikan bahwa operasi pengelolaan pengguna dapat dijalankan secara optimal dan sesuai.

B. Pengujian Fitur *Login*

Tabel 4.2 Pengujian Fitur Login

No	Skenario	Input	Output	Hasil
1	<i>Login</i> benar	Username dan password valid	Berhasil <i>Login</i>	Berhasil
2	Password salah	Password salah	<i>Login</i> gagal	Berhasil
3	Field kosong	Kosong	Validasi muncul	Berhasil

Dari hasil pengujian *Black-Box* testing tersebut, fitur *login* telah memenuhi kebutuhan fungsional sistem. Pengujian *User Acceptance Testing (UAT)* dilakukan sebagai tahap lanjutan untuk memastikan bahwa operasi pengelolaan pengguna dapat dijalankan secara optimal dan sesuai.

C. Pengujian Tambah Keuangan

Tabel 4.3 Pengujian Tambah Keuangan

No	Skenario	Input	Output	Hasil
1	Input keuangan	Data keuangan lengkap	Input Tersimpan	Berhasil
2	Lihat invoice keuangan	Data keuangan terlihat seperti laporan	Data keuangan berbentuk halaman print	Berhasil
3	Edit keuangan	Data diperbarui	Data berubah	Berhasil
4	Hapus keuangan		Data Terhapus	Berhasil
5	Melihat seluruh laporan keuangan	<i>Export PDF</i>	Keluar halaman laporan berbentuk print	Berhasil

Dari hasil pengujian *Black-box* Testing tersebut. Fitur tambah keuangan telah memenuhi kebutuhan fungsional sistem. Pengujian *User Acceptance Testing (UAT)* dilakukan sebagai tahap lanjutan untuk memastikan bahwa operasi pengelolaan pengguna dapat dijalankan secara optimal dan sesuai.

D. Pengujian Tambah Dokumen

Tabel 4.4 Pengujian Tambah Dokumen

No	Skenario	Input	Output	Hasil
1	Input dokumen	Data dokumen lengkap	Inputan tersimpan	Berhasil
2	Lihat dokumen	Dokumen terlihat seperti laporan	Dokumen berbentuk halaman print	Berhasil
3	Dokumen diedit	<i>User</i> bisa mengedit dokumen tersebut	Sistem mengeluarkan form edit untuk mengedit dokumen	Berhasil

4	Simpan <i>PDF</i>	User bisa menyimpan dokumen dalam bentuk PDF	Dokumen Tersimpan	Berhasil
5	Hapus dokumen		Dokumen terhapus	Berhasil

Dari hasil pengujian *Black-Box* Testing tersebut. Fitur Tambah dokumen telah memenuhi kebutuhan fungsional sistem. Pengujian *User Acceptance Testing (UAT)* dilakukan sebagai tahap lanjutan untuk memastikan bahwa operasi pengelolaan pengguna dapat dijalankan secara optimal dan sesuai.

E. Pengujian Upload Dokumen

Tabel 4.5 Pengujian Upload Dokumen

No	Skenario	Input	Output	Hasil
1	Upload Dokumen	File dokumen di ambil dari File Explorer	Dokumen Terupload dan muncul di halaman daftar dokumen	Berhasil
2	Lihat dokumen	Dokumen terlihat seperti laporan	Dokumen berbentuk halaman print	Berhasil
3	Simpan <i>PDF</i>	Dokumen tersimpan	Dokumen tersimpan dalam bentuk <i>PDF</i>	Berhasil
4	Hapus dokumen		Dokumen terhapus	Berhasil

Dari hasil pengujian *Black-Box* Testing tersebut. Fitur Upload dokumen telah memenuhi kebutuhan fungsional sistem. Pengujian *User Acceptance Testing (UAT)* dilakukan sebagai tahap lanjutan untuk memastikan bahwa operasi pengelolaan pengguna dapat dijalankan secara optimal dan sesuai.

4.3 Hasil Persentase Pengujian

Untuk mengetahui tingkat keberhasilan fungsi—fungsi yang telah dikembangkan dilakukan evaluasi sistem. Pengujian ini dilakukan dengan membandingkan hasil yang diharapkan dengan hasil yang sebenarnya. Jika hasilnya sesuai dengan harapan, pengujian tersebut dikategorikan, dan jika tidak sesuai, pengujian tersebut dianggap tidak valid. Berdasarkan hasil ini, perhitungan persentase keberhasilan dilakukan untuk mengetahui seberapa baik sistem secara keseluruhan berfungsi.

Tabel 4.6 Hasil Persentase Pengujian

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Aktual	Persentase
1	<i>Register</i>	Akun berhasil dibuat	Sesuai	100%
2	<i>Login</i> pengguna	Berhasil masuk ke sistem	Sesuai	100%
3	Tambah keuangan	Data – Data keuangan berhasil di tambahkan ke daftar halaman keuangan	Sesuai	100%
4	Tambah dokumen	Data – Data dokumen berhasil ditambahkan ke daftar halaman dokumen	Sesuai	100%
5	Upload dokumen	Dokumen yang di upload melalui File Explorer berhasil ditambahkan ke daftar halaman dokumen	Sesuai	100%

Dari hasil di atas dapat diketahui bahwa sebagian besar skenario pengujian telah berjalan sesuai dengan hasil yang diharapkan. Sehingga semua fitur dinyatakan Berpersentase 100%, fitur – fitur utama seperti registrasi, *login*, pengelolaan data keuangan, serta pengelolaan dan unggah dokumen telah berjalan baik tanpa kendala. Dengan demikian, pengujian lanjutan tetap diperlukan untuk memastikan kestabilan sistem dalam berbagai kondisi pengguna.

BAB 5

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada proyek akhir ini, dapat disimpulkan bahwa tujuan pengembangan aplikasi keuangan dan penyimpanan dokumen berbasis *web* dan *mobile* di PT. Inti Mas Mulia telah tercapai. Aplikasi yang dibangun mampu mengintegrasikan pencatatan transaksi keuangan serta pengelolaan dokumen perusahaan dalam satu sistem yang terpusat dan mudah digunakan. Hasil utama dari proyek akhir ini menunjukkan bahwa aplikasi berhasil diimplementasikan menggunakan *framework flutter* sebagai Solusi lintas platform, sehingga dapat dijalankan pada perangkat *web* dan *mobile* dengan tampilan yang konsisten. Pemanfaatan *firebase* sebagai *backend* juga berhasil mendukung proses autentikasi pengguna, penyimpanan data transaksi keuangan, serta mengelola metadata dokumen secara aman dan *real-time*.

Fitur – Fitur utama seperti *login*, pencatatan keuangan, pengelolaan dokumen, dan pembuatan laporan telah berjalan sesuai dengan kebutuhan pengguna. Pengujian menggunakan *Black-Box*, seluruh fungsi sistem dapat beroperasi dengan baik tanpa ditemukan kesalahan yang signifikan. Temuan penting dari proses pengembangan menunjukkan bahwa digitalisasi pencatatan keuangan dan penyimpanan dokumen dapat meningkatkan efisiensi kerja, mengurangi resiko kesalahan pencatatan, serta mempermudah akses data bagi pengguna. Dengan demikian, aplikasi ini dapat dijadikan sebagai Solusi yang efektif untuk mendukung pengelolaan keuangan dan dokumen di PT. Inti Mas Mulia.

5.2 Saran

Berdasarkan hasil perancangan, implementasi, dan pengujian aplikasi keuangan dan penyimpanan dokumen berbasis *web* dan *mobile* di PT. Inti Mas Mulia, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembang dan penerapan sistem di masa mendatang.

Pertama, dari sisi pengembangan sistem, aplikasi ini masih dapat ditingkatkan dengan menambahkan fitur lanjutan seperti pengelolaan hak akses pengguna berdasarkan peran (*role-based access control*), integrasi dengan sistem akuntansi atau pembayaran digital, serta peningkatan keamanan data melalui penerapan enkripsi tambahan pada penyimpanan dokumen. Selain itu, pengembangan fitur analisis dan visualisasi laporan keuangan yang lebih detail juga dapat membantu manajemen dalam pengambilan Keputusan strategis.

Kedua, dari sisi implementasi di lapangan, disarankan agar perusahaan menyiapkan infrastruktur pendukung yang memadai, trauma koneksi internet yang stabil untuk mendukung sinkronisasi data secara *real-time*. Selain itu, perlu dilakukan pelatihan kepada pengguna agar seluruh fitur aplikasi dapat dimanfaatkan secara optimal.

DAFTAR PUSTAKA

- (Alfonso & Setiawati, 2024). Aplikasi manajemen uang kas kelas berbasis *mobile* dengan *framework flutter*. *Journal of Computer Science and Information Technology (JCSIT)*, 2(1).
- (Anjilna, 2025).Efektifitas digitalisasi terhadap pengelolaan dan penyimpanan arsip dalam pengaturan protokol notaris.
- (Apisca et al., 2024). Aplikasi Pencatatan Keuangan Berbasis *Website* Dengan Metode *Rapid Application Development* Pada PT Samsriwi Adi Megah. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 6(1), 212–219.
- (Astuti & Hermawan, n.d.). Pemanfaatan *firebase realtime database* pada aplikasi pembelajaran agama islam menggunakan *framework flutter*
- (Diana & Putra, 2021). *InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan*.
- (Fafions Osama Effendy et al., 2025). Catatan Keuangan UMKM Berbasis *Mobile* Menggunakan *Flutter*. *Repeater : Publikasi Teknik Informatika Dan Jaringan*, 3(2), 75–88.
- (Fahmi Aziz & Sanjaya, 2023). 3520 <https://ejournal.sidyanusa.org/index.php/jkdn> Aplikasi Kas Berbasis *Flutter* untuk Meningkatkan Efisiensi Pencatatan Transaksi Keuangan (Vol. 3, Issue 1).
- (Lendo & Mapaly, 2023). Rancang Bangun Aplikasi Perpustakaan Digital Berbasis *Mobile* Menggunakan *Framework Flutter*.
- (Ramdan et al., 2021).Pengelolaan keuangan pribadi yang interaktif berbasis android.
- (Renaldy & Pakereng, 2023). Aplikasi Arsip Surat *Mobile* Menggunakan *Framework Flutter* (Studi Kasus PT. BPD Maluku, Malut).
- (Soepandi et al., 2023). Rancang Bangun Aplikasi Pengelolaan Keuangan Berbasis Android Menggunakan *Framework React Native*. In *Informatics Journal* (Vol. 8, Issue 3).
- (Sudarsana, 2024).Analisis digitalisasi arsip di dinas arsip dan perpustakaan kabupaten karawang.

LAMPIRAN

A. Lampiran Link Proyek

- c. Aplikasi : <https://ptimm1.web.app> -
- d. ZIP: <https://drive.google.com/drive/folders/1Ba5eKcNl10Zs7uCJbbo2haX9lw71PcJ?usp=sharing>

B. Lampiran Link Dokumen Pengujian UAT

<https://drive.google.com/drive/u/0/folders/1Odgum5IVtywr2ZgqzpxcSMxSqoeWJxbV>

C. Lampiran Link Figma

<https://www.figma.com/design/YkQBXY6cCWtXgMIo0yK2D7/DESAIN-PT.IMM?node-id=4-2&t=vc7xjzPILMGsW4Rs-0>

D. Lampiran Pengujian Dokumentasi

<https://drive.google.com/drive/u/0/folders/1n-Z1LpaUcWHTNKRuMNjiRxMcP4lCNfms>