

**DPPL-XX**

## **DESKRIPSI PERANCANGAN PERANGKAT LUNAK**

### **Finance Management Application**

untuk:

Finance Management Application

Dipersiapkan oleh :

Putri Khairamulya Ramadhini - 1301220176

Marsha Sukma Ningsih - 1301223024

Luthfiah Nazla Hidayat - 1301223363

Raisa Lukman Hakim - 1301223301

Necha Agmeyla Tama - 1301223412

Muhammad Amer Hadafi Utomo – 1301223096

**Program Studi Teknologi Informasi**

**Fakultas Informatika**

**Jl. Telekomunikasi 1, Dayeuhkolot Bandung**

Telkom University	Prodi S1- Teknologi Informasi Universitas Telkom	Nomor Dokumen		Halaman
		<b>DPPL-xx &lt;Kelompok E&gt;</b>		<#/>/<jml #
		Revisi	<nomor revisi>	Tgl: <06 Januari 2023>

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	
<b>G</b>	

INDEX	-	A	B	C	D	E	F	G
TGL								

Ditulis oleh							
Diperiksa oleh							
Disetujui oleh							

## Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

## Daftar Isi

Daftar Isi.....	4
1. Pendahuluan.....	6
1.1 Tujuan Penulisan Dokumen.....	7
1.2 Lingkup Masalah.....	7
1.3 Definisi dan Istilah.....	7
1.4 Referensi.....	8
2 Perancangan Global.....	9
2.1 Rencana Lingkungan Implementasi.....	9
2.2 Deskripsi Arsitektur Perangkat Lunak.....	9
3 Perancangan Rinci.....	11
3.1 Realisasi Use Case.....	11
3. 1.1 Use Case #1 <SIGN IN>.....	11
3.1.1.1 Use Case Scenario #1 “ACCOUNT SIGN IN”.....	11
3.1.1.2 UI Design dan Deskripsi Objek UI #1 <SIGN IN>.....	13
3.1.1.3 Identifikasi Objek dan Tipe Kelas #1 <SIGN IN>.....	14
3.1.1.4 Sequence Diagram #1 <SIGN IN>.....	14
3.1.2 Use Case #2 <LOGIN>.....	15
3.1.2.1 Use Case Scenario #2 “Login Account”.....	15
3.1.2.2 UI Design dan Deskripsi Objek UI #2 <LOG IN>.....	16
3.1.2.3 Identifikasi Object Baru & Tipe Kelas #2 <LOGIN>.....	17
3.1.2.4 Sequence Diagram #2 <LOGIN>.....	17
3.1.3 Use Case #3 <Record Pendapatan>.....	17
3.1.3.1 Use Case Scenario #3 “Record Pendapatan”.....	18
3.1.3.2 UI Design dan Deskripsi Objek UI #3 <Record Pendapatan>.....	19
3.1.3.3 Identifikasi Object Baru & Tipe Kelas #3 <Record Pendapatan>.....	20
3.1.3.4 Sequence Diagram #3 <Record Pendapatan>.....	20
3.1.4 Use Case #4 <Perencanaan Rancangan Anggaran>.....	21
3.1.4.1 Use Case Scenario #4 “Perencanaan Rancangan Anggaran”.....	21
3.1.4.2 UI Design dan Deskripsi Objek UI #4 <Perencanaan Rancangan Anggaran>.....	22
3.1.4.3 Identifikasi Object Baru & Tipe Kelas #4 <Perencanaan Rancangan Keuangan>.....	23
3.1.4.4 Sequence Diagram #4 <Perencanaan Rancangan Keuangan>.....	23
3.1.5 Use Case #5 <Identifikasi Pengeluaran>.....	24
3.1.5.1 Use Case Scenario #5 “Identifikasi Pengeluaran”.....	24
3.1.5.2 UI Design dan Deskripsi Objek UI #5 <Identifikasi Pengeluaran>.....	25
3.1.5.3 Identifikasi Object Baru & Tipe Kelas #5 <Identifikasi Pengeluaran>.....	26
3.1.5.4 Sequence Diagram #5 <Identifikasi Pengeluaran>.....	27

3.1.6 Use Case #6 <Laporan Keuangan>.....	27
3.1.6.1 Use Case Scenario #6 “Laporan Keuangan” .....	27
3.1.6.2 UI Design dan Deskripsi Objek UI #6 <Laporan Keuangan>.....	28
3.1.6.3 Identifikasi Object Baru & Tipe Kelas #6 <Laporan Keuangan>.....	29
3.1.6.4 Sequence Diagram #6 <Laporan Keuangan>.....	29
3.1.7 Use Case #7 <Peringatan Penggunaan>.....	30
3.1.7.1 Use Case Scenario #7 “Peringatan Penggunaan”.....	30
3.1.7.2 UI Design dan Deskripsi Objek UI #7 <Peringatan Penggunaan>.....	31
3.1.7.3 Identifikasi Object Baru & Tipe Kelas #7 <Peringatan Penggunaan>.....	31
3.1.7.4 Sequence Diagram #7 <Peringatan Penggunaan>.....	32
4. Diagram Kelas Keseluruhan.....	33
4. 1 Perancangan Detail Kelas.....	34
4.2 Perancangan Algoritma dan/atau Query.....	34
5. Matriks Kerunutan (Requirement Traceability Matrix).....	39

# **1. Pendahuluan**

## **1.1 Tujuan Penulisan Dokumen**

Dokumen Software Requirements Specification (SRS) ini mencakup spesifikasi perangkat lunak untuk Aplikasi Manajemen Keuangan Pribadi dengan Nomor Rilis 1.0. Perangkat lunak ini akan membantu mengatur dan melacak keuangan penggunanya, tujuannya adalah untuk membantu pengguna mengelola pengeluaran, menentukan anggaran dan mengidentifikasi pengeluaran penggunanya. Perangkat lunak pengatur keuangan ini digunakan untuk mencatat semua transaksi untuk mempermudah pengelolaan keuangan. Perangkat membantu pengguna untuk merencanakan keuangan jangka panjang dan potensi penghematan uang dan membantu membuat keputusan keuangan yang lebih baik. Produk ini mencakup fitur-fitur utama seperti Manajemen Akun Pengguna, Pencatatan Transaksi Keuangan, Perencanaan dan Pemantauan Anggaran, Pengidentifikasi Pengeluaran, Pelaporan Keuangan, dan Peringatan Anggaran. Aplikasi ini dirancang untuk beroperasi pada berbagai platform termasuk Windows, macOS, dan Android, memastikan ketersediaan kepada berbagai jenis pengguna. Dengan ketergantungan pada API dari bank dan lembaga keuangan eksternal, aplikasi ini menyediakan fungsionalitas yang luas untuk membantu individu, seperti mahasiswa, pemilik bisnis, dan keluarga, mengelola keuangan pribadi mereka dengan lebih efektif.

## **1.2 Lingkup Masalah**

Dalam dokumen Spesifikasi Persyaratan Perangkat Lunak (SKPL) ini, terdapat serangkaian konvensi penulisan yang diatur untuk memastikan keterbacaan, struktur yang jelas, dan pemahaman yang optimal. Beberapa standar atau konvensi pengetikan yang diterapkan dalam SKPL ini adalah:

1. Prioritas Kebutuhan: Setiap kebutuhan pengguna dalam perangkat lunak dibagi menjadi tiga tingkatan: primer, sekunder, dan tersier. Hal ini memberikan pemahaman yang jelas mengenai urgensi dan pentingnya setiap kebutuhan.
2. Penomoran Kebutuhan: Setiap kebutuhan diberikan nomor referensi untuk memudahkan identifikasi dan organisasi. Penomoran yang berbeda diberikan untuk kebutuhan utama dan kebutuhan rincian, memfasilitasi pemahaman struktur kebutuhan secara hierarkis.
3. Deskripsi Kebutuhan: Setiap kebutuhan dilengkapi dengan deskripsi kepentingan yang jelas dan komprehensif. Hal ini membantu dalam pemahaman mengapa suatu kebutuhan diperlukan, serta tingkat keperluannya. Deskripsi yang rinci memastikan pemahaman yang mendalam tentang setiap elemen kebutuhan.
4. Highlight atau Warna: Tingkatan kebutuhan diberikan warna yang berbeda sesuai dengan prioritasnya. Pemberian warna juga diterapkan pada sisa saldo dan total pengeluaran untuk memberikan penekanan visual terhadap informasi yang krusial. Hal ini memudahkan pembaca dalam mengidentifikasi dan memahami urgensi setiap elemen.
5. Font: Font yang digunakan dalam dokumen ini adalah Arial. Pemilihan font yang konsisten membantu dalam memberikan kejelasan dan keterbacaan dokumen.
6. Perincian Kebutuhan: Setiap kebutuhan diperinci dengan detail yang mencakup aspek-aspek tertentu. Hal ini meningkatkan kejelasan dan menghindari interpretasi ganda terhadap kebutuhan tersebut.
7. Header dan Footer: Setiap halaman dokumen mencakup header yang mencantumkan informasi penting seperti nama proyek, nomor revisi, dan nama penulis proyek. Footer mencantumkan tanggal revisi, memberikan konteks waktu terhadap perkembangan dokumen.

## **1.3 Definisi dan Istilah**

Perangkat lunak yang dispesifikasi dalam dokumen SKPL ini adalah Aplikasi Manajemen Keuangan Pribadi. Tujuan utama dari perangkat lunak ini adalah membantu pengguna dalam mengelola keuangan pribadi mereka dengan lebih efektif. Dengan fokus pada pencatatan transaksi keuangan, perencanaan anggaran, dan identifikasi pola pengeluaran, aplikasi ini bertujuan untuk memberikan alat yang kuat bagi pengguna agar dapat membuat keputusan keuangan yang lebih baik.

Manfaat yang relevan dari perangkat lunak ini mencakup:

1. Pencatatan Transaksi: Mempermudah pengguna dalam mencatat semua transaksi keuangan mereka, baik pendapatan maupun pengeluaran, untuk menciptakan pemahaman yang lebih baik tentang situasi keuangan mereka.
2. Perencanaan Anggaran: Memungkinkan pengguna untuk menetapkan dan memantau anggaran keuangan, memberikan visibilitas terhadap alokasi dana dan membantu menghindari pengeluaran yang tidak terduga.

3. Identifikasi Pengeluaran: Membantu pengguna dalam mengenali dan mengelompokkan jenis pengeluaran, memberikan wawasan yang lebih baik tentang pola pengeluaran dan memungkinkan untuk membuat perubahan yang sesuai.

Aplikasi ini juga sejalan dengan tujuan perusahaan atau strategi bisnis dengan menyediakan solusi yang dapat meningkatkan kesejahteraan finansial pengguna. Dengan memberdayakan individu untuk mengelola keuangan pribadi mereka, perusahaan dapat mencapai tujuan strategis seperti peningkatan kepuasan pelanggan, membangun reputasi sebagai penyedia solusi finansial yang andal, dan berkontribusi pada perkembangan keuangan yang berkelanjutan.

Sasaran aplikasi ini melibatkan berbagai kelompok pengguna, termasuk mahasiswa, pemilik bisnis, dan keluarga. Dengan menyediakan fitur-fitur yang relevan untuk setiap kelompok ini, perangkat lunak ini diharapkan dapat menjadi alat yang berguna dan bermanfaat dalam konteks keuangan pribadi mereka. Dengan memahami visi dan lingkup dokumen, pengguna dapat lebih baik mengimplementasikan dan memanfaatkan fungsionalitas yang disediakan oleh aplikasi ini sesuai dengan kebutuhan dan tujuan mereka.

#### **1.4 Referensi**

Hasan, S., Elpisah, E., Sabtohadi, J., Nurwahidah, M., Abdullah, A., & Fachrurazi, F. (2022). Manajemen keuangan. Penerbit Widina. Musthafa, H., & SE, M. (2017). Manajemen keuangan. Penerbit Andi.

## 2 Perancangan Global

### 2.1 Rencana Lingkungan Implementasi

1. Arsitektur Umum:
  - 1.1 Model View Controller (MVC): Menggunakan pola arsitektur MVC untuk memisahkan logika aplikasi, presentasi data, dan interaksi pengguna.
  - 1.2 Client-Server Model: Aplikasi memiliki model client-server, dengan peran "Client" diambil oleh aplikasi yang dijalankan oleh pengguna dan "Server" yang mengelola data dan logika di belakang layar.
2. Komponen Utama:
  - 2.1 User Interface (UI): Terdiri dari berbagai objek UI seperti tombol, textbox, dan RTF box di setiap layar (Home Page, Page Registrasi, Page Transaksi).
  - 2.2 Controller: Berperan dalam mengatur alur kontrol, validasi input, dan memproses logika bisnis. Dapat mencakup objek seperti EmailValidator untuk memastikan format email yang benar.
  - 2.3 Entity: Mewakili entitas atau data dalam aplikasi, seperti UserAccount yang menyimpan informasi tentang pengguna yang terdaftar.
3. Alur Kerja:
  - 3.1. Pengguna berinteraksi dengan aplikasi melalui UI.
  - 3.2. Controller mengelola alur kerja, memvalidasi data, dan berkomunikasi dengan entitas (Entity).
  - 3.3. Registrasi User melibatkan pengisian formulir pada Page Registrasi, dan transaksi dilakukan pada Page Transaksi.
  - 3.4. Proses registrasi melibatkan validasi dan pengiriman verifikasi melalui email.
  - 3.5. Page Transaksi memungkinkan pengguna melakukan operasi terkait transaksi dan menyajikan informasi transaksi.
4. Tipe Kelas:
  - Boundary (Interface): UserInterface mewakili interaksi antara pengguna dan sistem.
  - Entity (Database): UserAccount menyimpan informasi pengguna.
  - Controller: EmailValidator bertindak sebagai kontrol validasi.

Perangkat lunak Manajemen Keuangan Berbasis Mobile ini dirancang untuk membantu pengguna mengelola dan melacak keuangan mereka, dengan fokus pada pengeluaran, penetapan anggaran, dan identifikasi pola pengeluaran. Aplikasi ini mencakup fitur-fitur utama seperti Manajemen Akun Pengguna, Pencatatan Transaksi Keuangan, Perencanaan dan Pemantauan Anggaran, Pengidentifikasi Pengeluaran, Pelaporan Keuangan, dan Peringatan Anggaran. Dengan operasionalitas pada berbagai platform termasuk Windows, macOS, dan Android, serta ketergantungan pada API dari bank dan lembaga keuangan eksternal, aplikasi ini memberikan fungsionalitas yang luas untuk membantu individu mengelola keuangan pribadi mereka. Rancangan Lingkungan Implementasinya mencakup aspek UI/UX, antarmuka perangkat keras, antarmuka perangkat lunak, dan fitur sistem cerdas, dengan penekanan pada keamanan dan privasi data melalui enkripsi dan kontrol akses yang ketat. Sebagai solusi komprehensif, aplikasi ini menawarkan pengalaman pengguna yang efektif dan mendukung keputusan keuangan yang cerdas. Untuk mendukung perangkat lunak ini, alat-alat pendukung seperti infrastruktur big data, integrasi API layanan keuangan eksternal, dan mekanisme pemeliharaan model menjadi bagian integral dalam memberikan dasar yang kuat untuk pengembangan dan pemeliharaan aplikasi ini.

### 2.2 Deskripsi Arsitektur Perangkat Lunak

#### Isinya komponen diagram

No	Nama Komponen	Keterangan
----	---------------	------------

1.	Manajemen Akun Pengguna	Fitur ini memungkinkan individu untuk membuat akun dan masuk ke dalam sistem untuk dapat menggunakan fitur-fitur dari perangkat lunak.
	Pencatatan Transaksi Keuangan	Pengguna dapat mencatat semua transaksi keuangan mereka, seperti pendapatan dan pengeluaran, guna melacak semua aktivitas keuangan yang terjadi.
	Perencanaan dan Pemantauan Anggaran	Perangkat lunak ini dilengkapi dengan alat-alat yang memungkinkan individu untuk membuat dan memantau anggaran keuangan mereka. Pengguna dapat menetapkan tujuan anggaran dan memonitor kemajuannya.
	Pengidentifikasi Pengeluaran	Fitur ini membantu pengguna dalam mengelompokkan dan mengidentifikasi berbagai jenis pengeluaran, sehingga memudahkan untuk memahami pola pengeluaran.
	Pelaporan Keuangan	Pengguna dapat menghasilkan laporan keuangan yang memberikan wawasan mendalam mengenai aktivitas keuangan mereka dan bagaimana kinerja anggaran mereka.
	Peringatan Anggaran	Perangkat lunak ini memiliki kemampuan untuk memberi notifikasi kepada pengguna ketika mereka mendekati atau melebihi batas anggaran atau tujuan keuangan yang telah ditetapkan.

### 3 Perancangan Rinci

#### 3.1 Realisasi Use Case

Berisi TABEL USE CASE sebagai berikut :

No	Nama Use Case	Deskripsi Use Case
1	Sign In	Pengguna yang belum memiliki akun dapat sign in terlebih dahulu, dengan memasukkan identitas sesuai yang diinstruksikan di halaman sign-in
2	Log in	Pengguna yang telah terdaftar dapat log in ke aplikasi dengan mengautentikasi identitas mereka menggunakan nama pengguna dan kata sandi yang benar.
3	Record Pendapatan	Pengguna dapat mencatat dan memantau pendapatan mereka dengan menambahkan entri transaksi pendapatan ke dalam aplikasi.
4	Pembuatan & pemanfaatan rencana anggaran keuangan	Pengguna dapat membuat, mengelola, dan memanfaatkan rencana anggaran keuangan untuk merencanakan dan memonitor pengeluaran mereka.
5	Identifikasi Pengeluaran	Pengguna dapat mengidentifikasi dan mencatat pengeluaran mereka dalam kategori yang ditentukan untuk memahami pola pengeluaran.
6	Laporan Keuangan	Pengguna dapat mengakses laporan keuangan yang menyajikan ringkasan transaksi, perkembangan anggaran, dan analisis keuangan lainnya.
7	Peringatan Pengguna	Aplikasi memberikan peringatan kepada pengguna terkait batas anggaran tertentu atau pemberitahuan lain yang relevan dengan keuangan mereka.

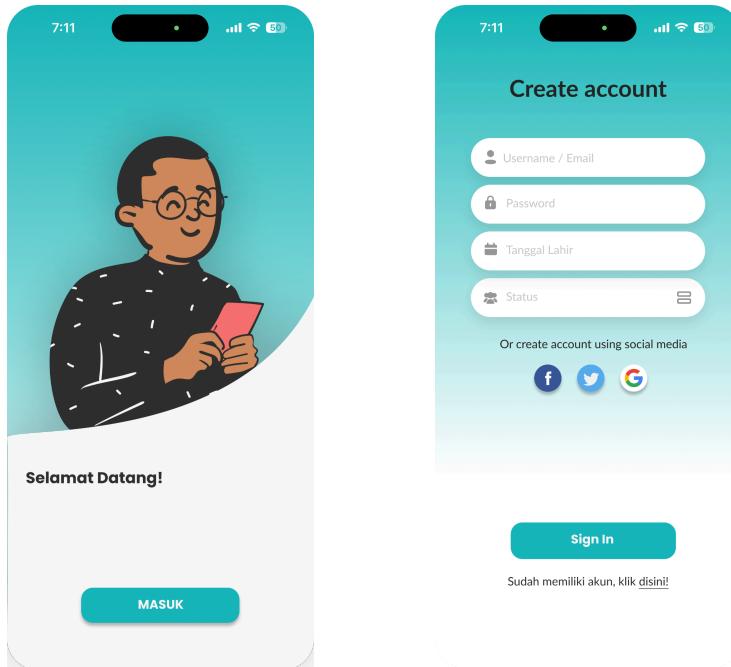
#### 3. 1.1 Use Case #1 <SIGN IN>

##### 3.1.1.1 Use Case Scenario #1 “ACCOUNT SIGN IN”

Nama Use Case	Sign In
Deskripsi	User sign in terlebih dahulu untuk membuat akun pada aplikasi
Aktor	1. User
Precondition	- User belum memiliki akun. - User Memiliki akses internet
Postcondition	Aplikasi akan membuat akun untuk user.

Normal Flow of event	<ol style="list-style-type: none"> <li>1. User membuka aplikasi             <ul style="list-style-type: none"> <li>- User memilih registrasi sign in</li> <li>- Aplikasi akan menampilkan halaman registrasi.</li> </ul> </li> <li>2. User memasukkan username, password, email pengguna, tanggal lahir, status individu (pelajar, keluarga muda, dll).             <ul style="list-style-type: none"> <li>- Sistem akan mengecek informasi yang diberikan oleh user.</li> <li>- Jika informasi yang diberikan oleh user itu valid maka sistem akan membuat profile user baru.</li> </ul> </li> <li>3. User akan mendapatkan verifikasi melalui email yang telah diinputkan             <ul style="list-style-type: none"> <li>- Email yang didapatkan user berisi link konfirmasi.</li> </ul> </li> <li>4. User akan verifikasi melalui link Profile user akan teraktivasi</li> </ol>
Alternative Flow of event	<ol style="list-style-type: none"> <li>1. Jika user memasukkan email yang telah memiliki akun, aplikasi akan menampilkan informasi             <ul style="list-style-type: none"> <li>- Aplikasi akan menampilkan "email tersebut sudah registrasi akun"</li> </ul> </li> </ol>
	<ol style="list-style-type: none"> <li>2. User akan submit ulang informasi</li> <li>3. Hal ini akan berulang sampai user memberikan informasi data yang valid</li> </ol>

### 3.1.1.2 UI Design dan Deskripsi Objek UI #1 <SIGN IN>



A0001

C0001

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
A0001	Welcome Page	Welcome Page menampilkan satu pilihan yaitu "masuk"
C0001	Sign In Page	Mengecek validitas data yang diinput oleh user

A001 : Page Welcome Page

Id_Objek	JENIS	LABEL*	Keterangan**
Button 1	Button	MASUK	Jika diklik, akan muncul halaman Sign In

C0001 : Page SIGN IN

Id_Objek	JENIS	LABEL*	Keterangan**
RTF 1	RTF BOX		Mengisi Username/Email yang akan didaftarkan
RTF 2	RTF BOX		Mengisi/membuat password
RTF 3	RTF BOX		Mengisi tanggal lahir
RTF 4	RTF BOX		Mengisi status (menikah, pelajar, dll)

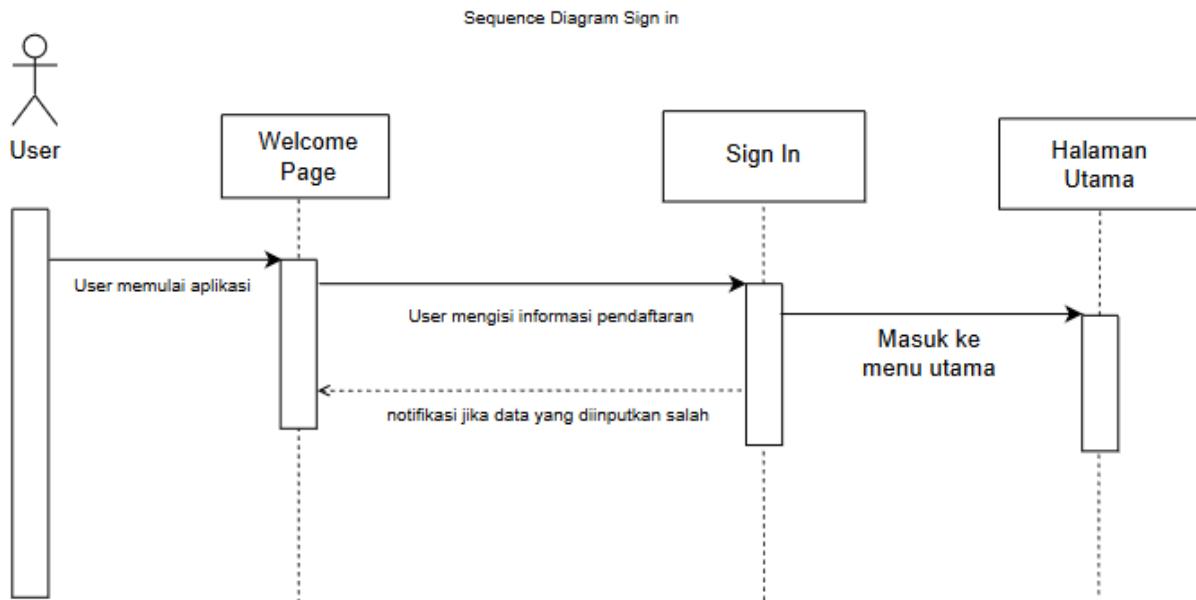
<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
Button 1	Button	Sign In	Jika memilih itu, maka sistem akan meng confirm Sign In
Button 2	Button	disini!	Jika memilih itu, akan mengaktifkan Use case Login dan beralih ke log in
Button 3	Button	facebook	Jika mengklik itu, akan diarahkan sign in dengan menggunakan Facebook
Button 4	Button	Google	Jika mengklik itu, akan diarahkan sign in dengan menggunakan Google
Button 5	Button	Twitter	Jika mengklik itu, akan diarahkan sign in dengan menggunakan Twitter

### 3.1.1.3 Identifikasi Objek dan Tipe Kelas #1 <SIGN IN>

TABEL OBJECT PERANCANGAN

<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1.	halaman register	Boundary
2.	Tombol “Register”	Controller
3.	Tombol “Login”	Controller
4.	Tombol “disini!”	Controller
5.	Tombol “facebook/twitter/google”	Controller
6.	Data register	Entity

### 3.1.1.4 Sequence Diagram #1 <SIGN IN>

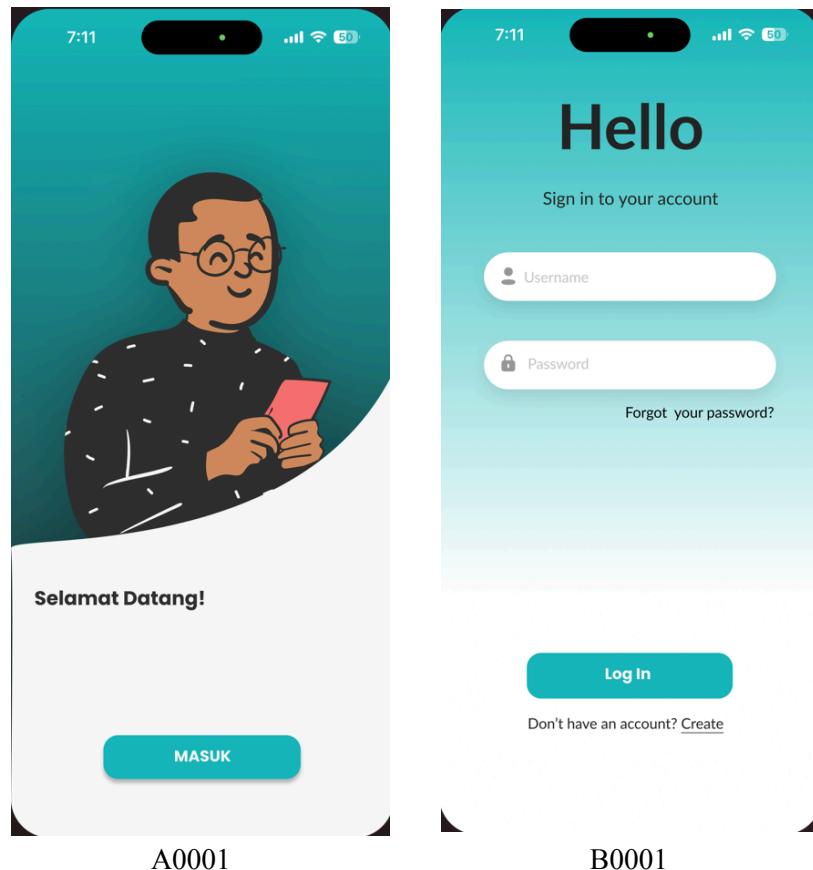


### 3.1.2 Use Case #2 <LOGIN>

#### 3.1.2.1 Use Case Scenario #2 “Login Account”

Nama Use Case	Login
Deskripsi	User login untuk dapat mengakses aplikasi
Aktor	1. User
Precondition	User telah memiliki akun. User memiliki akses internet.
Postcondition	User berhasil masuk ke akunnya di aplikasi.
Normal Flow of event	<ol style="list-style-type: none"><li>1. User membuka aplikasi<ul style="list-style-type: none"><li>- User memilih log in</li><li>- Aplikasi akan menampilkan halaman login.</li></ul></li><li>2. User memasukkan username dan password</li><li>3. Aplikasi akan memverifikasi user agar user dapat menggunakan aplikasi.</li><li>4. User dapat mengakses fitur-fitur pada aplikasi</li></ol>
Alternative Flow of event	<ol style="list-style-type: none"><li>1. Jika user memasukkan username dan password yang salah, aplikasi akan menampilkan informasi<ul style="list-style-type: none"><li>- Aplikasi akan menampilkan “username atau password yang diinputkan salah”</li></ul></li><li>2. User akan submit ulang informasi</li><li>3. Hal ini akan berulang sampai user memberikan informasi data yang valid</li></ol>

### 3.1.2.2 UI Design dan Deskripsi Objek UI #2 <LOG IN>



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
A0001	Welcome Page	Welcome Page menampilkan satu pilihan yaitu "masuk"
B0001	Page Login	Mengecek validitas dari Username dan Password yang akan dimasukkan

B0001 : Page LOGIN

ID_Objek	JENIS	LABEL*	Keterangan**
Button1	Button	Log In	Jika diklik akan mengkonfirmasi Login
Buuton 2	Button	Create	Jika diklik akan mengaktifkan Use Case Sign In
Button 3	Button	Forgot your password?	Jika diklik akan diarahkan untuk mengganti password, menyambungkan dengan email
RTF 1	RTF Box		Isi Username yang terdaftar
RTF 2	RTF BOX		Isi Password yang terdaftar

A0001 : page Home Page

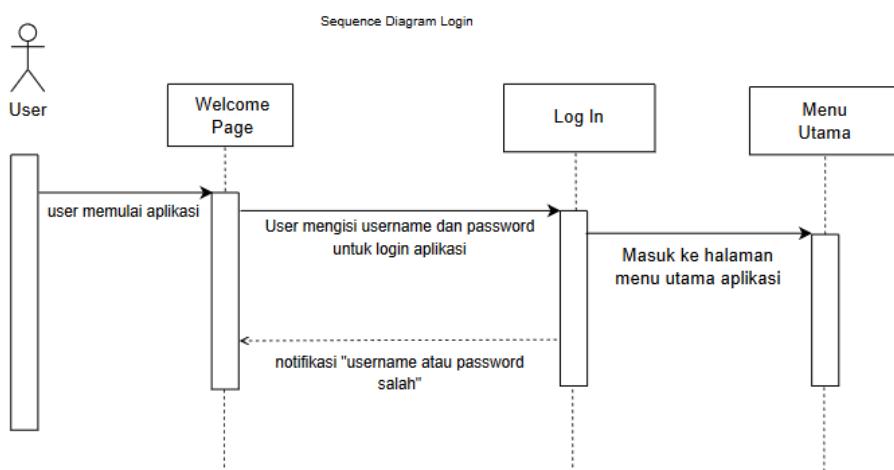
<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
Button 1	Button	Masuk	Jika di klik, akan muncul halaman Sign In

### 3.1.2.3 Identifikasi Object Baru & Tipe Kelas #2 <LOGIN>

TABEL OBJECT PERANCANGAN

<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1.	Halaman login	Boundary
2.	Tombol “login”	Controller
3.	Tombol “Forgot your password?”	Controller
4.	Tombol “Create”	Controller
5.	Data login	Entity

### 3.1.2.4 Sequence Diagram #2 <LOGIN>

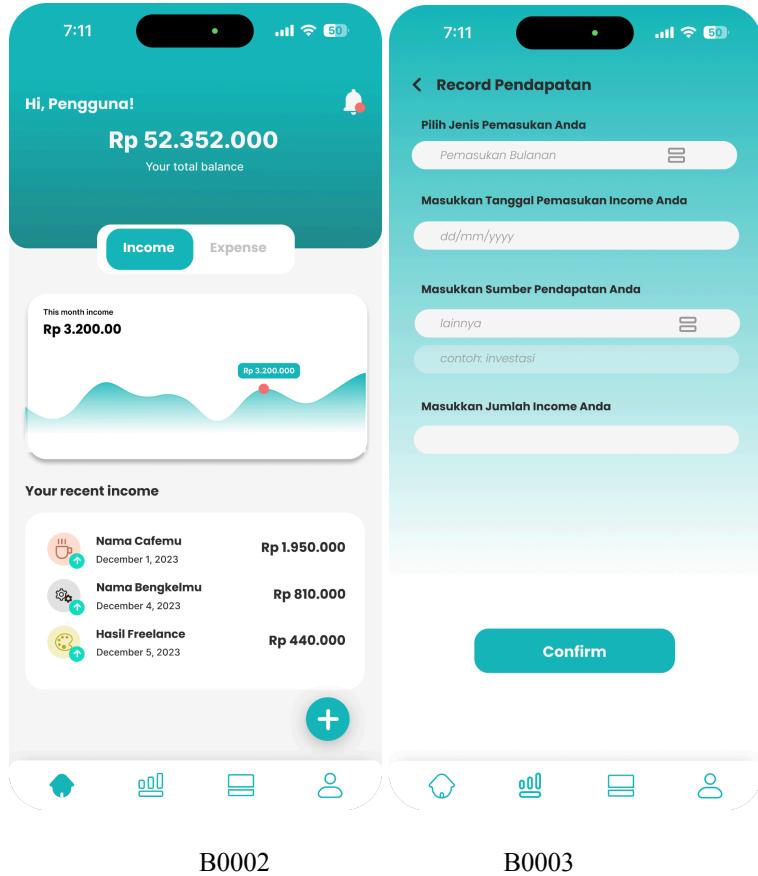


### 3.1.3 Use Case #3 <Record Pendapatan>

### 3.1.3.1 Use Case Scenario #3 “Record Pendapatan”

Nama Use Case	Record Pendapatan
Deskripsi	User dapat mencatat pendapatan mereka dalam aplikasi.
Aktor	1. User
Precondition	User sudah login ke akun nya.
Postcondition	Pendapatan keuangan tersimpan dalam aplikasi.
Normal Flow of event	<ol style="list-style-type: none"><li>1. User masuk ke dalam aplikasi.</li><li>2. User memilih menu "Record Pendapatan."</li><li>3. User mengisi informasi tentang pendapatan (jumlah, sumber, tanggal, dsb.).</li><li>4. Sistem menyimpan catatan pendapatan tersebut.</li></ol>
Alternative Flow of event	

### 3.1.3.2 UI Design dan Deskripsi Objek UI #3 <Record Pendapatan>



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
B0002	Page Menu Record Pendapatan	Layar Utama untuk menampilkan pendapatan terbaru dan statistik..
B0003	Page Record Pendapatan	Layar Utama untuk merekam pendapatan user dalam aplikasi

B0002 : Page Menu Record Pendapatan

ID_Objek	JENIS	LABEL*	Keterangan**
Button1	Button	Menu Pendapatan	jika di klik, akan membuka menu pendapatan dan akan menampilkan pendapatan terbaru dan statistik..
Button 2	Button	Menambahkan pendapatan	Jika diklik akan membuka page record pendapatan.

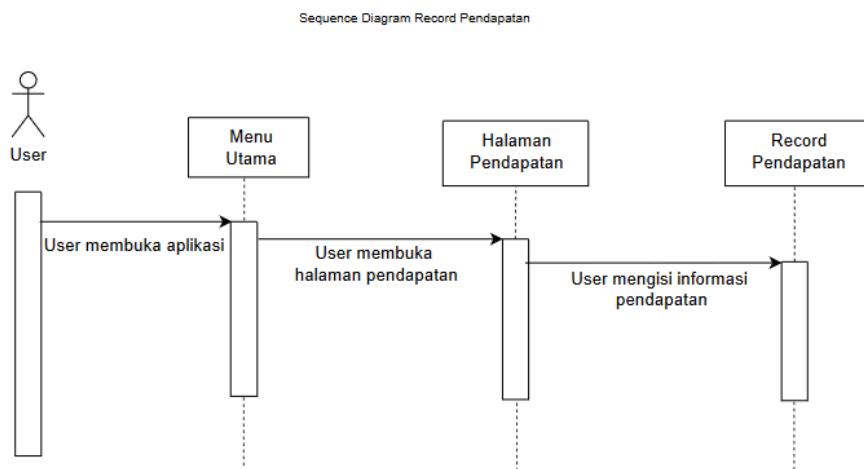
<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
Button1	Button	jenis pemasukan	jika di klik, akan membuka pilihan jenis pemasukan yakni mingguan, bulanan, tahunan
Button 2	Button	tanggal pemasukkan	Jika diklik akan membuka kalender untuk memilih tanggal.
Button 3	Button	sumber pendapatan	Jika diklik akan memilih opsi untuk memilih sumber pendapatan
RTF 1	RTF Box	jumlah pendapatan pengguna	digunakan untuk mengisi informasi jumlah pendapatan
RTF 2	RTF BOX		Isi Password yang terdaftar

### 3.1.3.3 Identifikasi Object Baru & Tipe Kelas #3 <Record Pendapatan>

TABEL OBJECT PERANCANGAN

<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1.	Form Record Pendapatan	Boundary
2.	Tombol "Create"	Controller
3.	Tombol "<"	Controller
4.	Tombol "confirm"	Controller
5.	Input Jenis Pemasukan	Entity
6.	Input Tanggal Pemasukan	Entity
7.	Input Sumber Pendapatan	Entity
8.	Input Jumlah Pemasukan	Entity

### 3.1.3.4 Sequence Diagram #3 <Record Pendapatan>



### **3.1.4 Use Case #4 <Perencanaan Rancangan Anggaran>**

#### **3.1.4.1 Use Case Scenario #4 "Perencanaan Rancangan Anggaran"**

Nama Use Case	Pembuatan & Pemanfaatan Rencana Anggaran Keuangan
Deskripsi	Aplikasi akan membuat rencana anggaran keuangan dari jumlah pendapatan yang telah diinputkan
Aktor	1. User 2. Aplikasi
Precondition	User memilih jenis rencana anggaran keuangan
Postcondition	Aplikasi menampilkan jenis rencana anggaran keuangan sesuai pilihan user
Normal Flow of event	<ol style="list-style-type: none"><li>1. User masuk ke dalam aplikasi.</li><li>2. User memilih menu "Rencana Anggaran Keuangan."</li><li>3. User memilih kategori anggaran</li><li>4. Aplikasi membuat rencana anggaran dengan menentukan anggaran untuk setiap kategori pengeluaran.</li><li>5. User dapat mengikuti rencana anggaran ini dalam manajemen keuangan mereka.</li></ol>
Alternative Flow of event	

### 3.1.4.2 UI Design dan Deskripsi Objek UI #4 <Perencanaan Rancangan Anggaran>



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
B0043	Page Perencanaan Rancangan Anggaran	Layar Utama untuk memilih dan akan menampilkan tabel rencana anggaran.

B0004 : Page Perencanaan Rancangan Anggaran

Id_Objek	JENIS	LABEL*	Keterangan**
Icon	Icon Back		Jika di klik, akan kembali kepada use case identifikasi pengeluaran bagian pertama (E0001)
Button1	Button	jenis Rencana Anggaran	jika di klik, akan membuka pilihan jenis pemasukan yakni mingguan, bulanan, tahunan
Button2	Button	RencanaAnggaran.docs	Jika di klik, akan mendownload rincian rencana anggaran dalam bentuk Docs.
Button3	Button	RencanaAnggaran.pdf	Jika di klik, akan mendownload rincian rencana anggaran dalam bentuk PDF.

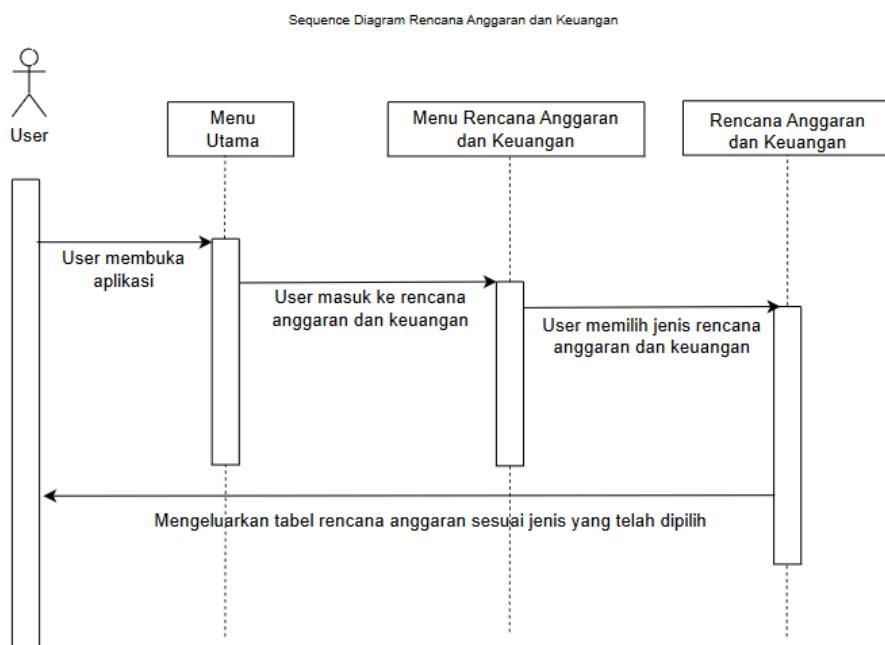
<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
Button4	Button	RencanaAnggaran.xls	Jika di klik, akan mendownload rincian rencana anggaran dalam bentuk Excel.
RTF 1	RTF Box	Total Saldo	Menampilkan informasi saldo saat ini.
RTF 2	RTF Box	Rencana Keuangan	Menampilkan rencana keuangan secara umum

### 3.1.4.3 Identifikasi Object Baru & Tipe Kelas #4 <Perencanaan Rancangan Keuangan>

TABEL OBJECT PERANCANGAN

<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1.	Form Record Pendapatan	Boundary
2.	Tombol "Create"	Controller
3.	Tombol "confirm"	Controller
4.	Input Jenis Rancangan Keuangan	Controller
5.	total saldo	Controller
6.	Rincian Anggaran keuangan	Entity

### 3.1.4.4 Sequence Diagram #4 <Perencanaan Rancangan Keuangan>

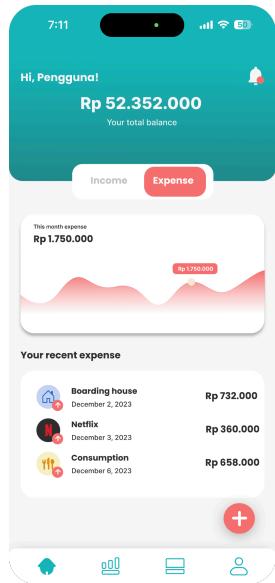


### **3.1.5 Use Case #5 <Identifikasi Pengeluaran>**

#### **3.1.5.1 Use Case Scenario #5 “Identifikasi Pengeluaran”**

Nama Use Case	Identifikasi Pengeluaran
Deskripsi	Aplikasi akan menyimpan jumlah pengeluaran yang telah diinputkan
Aktor	1. User
Precondition	User akan memasukkan setiap pengeluaran
Postcondition	Pengeluaran user berhasil dicatat dalam sistem.
Normal Flow of event	<ol style="list-style-type: none"><li>1. User masuk ke dalam aplikasi.</li><li>2. User memilih menu "Identifikasi Pengeluaran."</li><li>3. User mencatat pengeluaran yang baru, termasuk jumlah, tujuan pengeluaran, dan tanggal pengeluaran.</li><li>4. Aplikasi menyimpan catatan pengeluaran tersebut.</li></ol>
Alternative Flow of event	

### 3.1.5.2 UI Design dan Deskripsi Objek UI #5 <Identifikasi Pengeluaran>



E0001



E0002



E0003

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
E0001	Page Menu Identifikasi Pengeluaran	Layar Utama untuk menampilkan pendapatan terbaru dan statistik..
E0002	Page Identifikasi Pengeluaran	Menampilkan halaman identifikasi pengeluaran bagian pertama.
E0003	Page Identifikasi Pengeluaran	Menampilkan halaman identifikasi pengeluaran bagian kedua atau selanjutnya

E0001 : Page Menu Identifikasi Pengeluaran

Id_Objek	JENIS	LABEL*	Keterangan**
Button1	Button	Menu Pengeluaran	jika di klik, akan membuka menu pengeluaran dan akan menampilkan pengeluaran terbaru dan statistik..
Button 2	Button	Menambahkan pengeluaran	Jika diklik akan membuka page identifikasi pengeluaran.

E0002 : Page Identifikasi Pengeluaran

Id_Objek	JENIS	LABEL*	Keterangan**
Icon	Icon Back		Jika di klik, akan kembali kepada menu homepage

<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
Button 1	Button	gambar notifikasi	Jika di klik notifikasi, akan muncul laporan peringatan pengguna
RTF 1	RTF BOX	Masukkan Tanggal Pengeluaran Anda	Menginputkan tanggal pengeluaran anda, (dalam format dd/mm//yy )
RTF 2	RTF BOX	Masukkan Tujuan pengeluaran Anda	Menginputkan tujuan pengeluaran anda (ex : makan, kebutuhan sekolah)
RTF 3	RTF BOX	Masukkan Jumlah Pengeluaran Anda	Menginputkan jumlah pengeluaran anda (Rp. xx.xxx.xxx)
Icon	Icon Next		Jika di klik, akan berganti selanjutnya ke page identifikasi pengeluaran bagian 2 (E0002)

E0003 : Page Identifikasi Pengeluaran

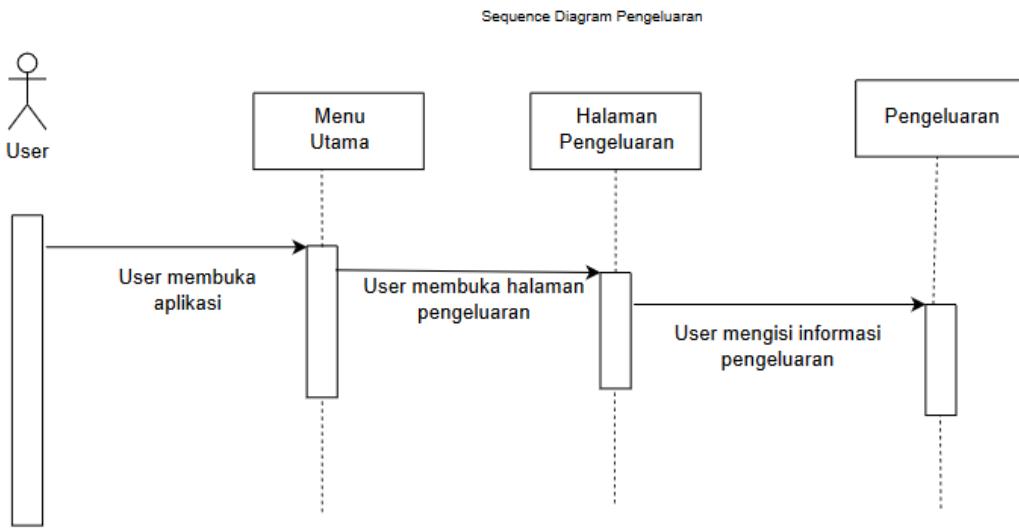
<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
Icon	Icon Back		Jika di klik, akan kembali kepada use case identifikasi pengeluaran bagian pertama (E0001)
Button 1	Button	gambar notifikasi	Jika di klik notifikasi, akan muncul laporan peringatan pengguna
RTF 1	RTF BOX		Menampilkan semua hasil inputan dari yang halaman sebelumnya diminta, beserta Total Pengeluaran dan Sisa Saldo anda.

### 3.1.5.3 Identifikasi Object Baru & Tipe Kelas #5 <Identifikasi Pengeluaran>

TABEL OBJECT PERANCANGAN

<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1.	Input “tanggal pengeluaran”	Entity
2.	Input “tujuan pengeluaran”	Entity
3.	Input “jumlah pengeluaran”	Entity
4.	Tombol “Icon Next”	Controller
5.	Tombol “Icon Back”	Controller
6.	Tombol “Notifikasi”	Boundary
7.	Tampilan hasil inputan identifikasi pengeluaran	Boundary

### 3.1.5.4 Sequence Diagram #5 <Identifikasi Pengeluaran>



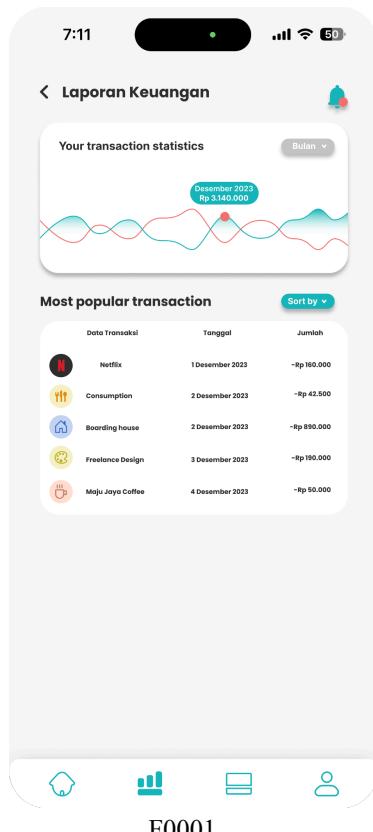
### 3.1.6 Use Case #6 <Laporan Keuangan>

#### 3.1.6.1 Use Case Scenario #6 “Laporan Keuangan”

Nama Use Case	Laporan Keuangan
Deskripsi	Aplikasi akan membuat laporan keuangan dari jumlah pendapatan dan pengeluaran yang telah diinputkan
Aktor	1. Aplikasi
Precondition	User memilih jenis laporan keuangan
Postcondition	Aplikasi menampilkan jenis laporan keuangan sesuai dengan pilihan user

Normal Flow of event	<ol style="list-style-type: none"> <li>1. User masuk ke dalam aplikasi.</li> <li>2. User memilih menu "Laporan Keuangan."</li> <li>3. Aplikasi menampilkan laporan keuangan yang berisi informasi tentang pendapatan dan pengeluaran user.</li> </ol>
Alternative Flow of event	

### 3.1.6.2 UI Design dan Deskripsi Objek UI #6 <Laporan Keuangan>



F0001

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
F0001	Page Laporan Keuangan	Menampilkan halaman laporan keuangan

F0001 : Page Laporan Keuangan

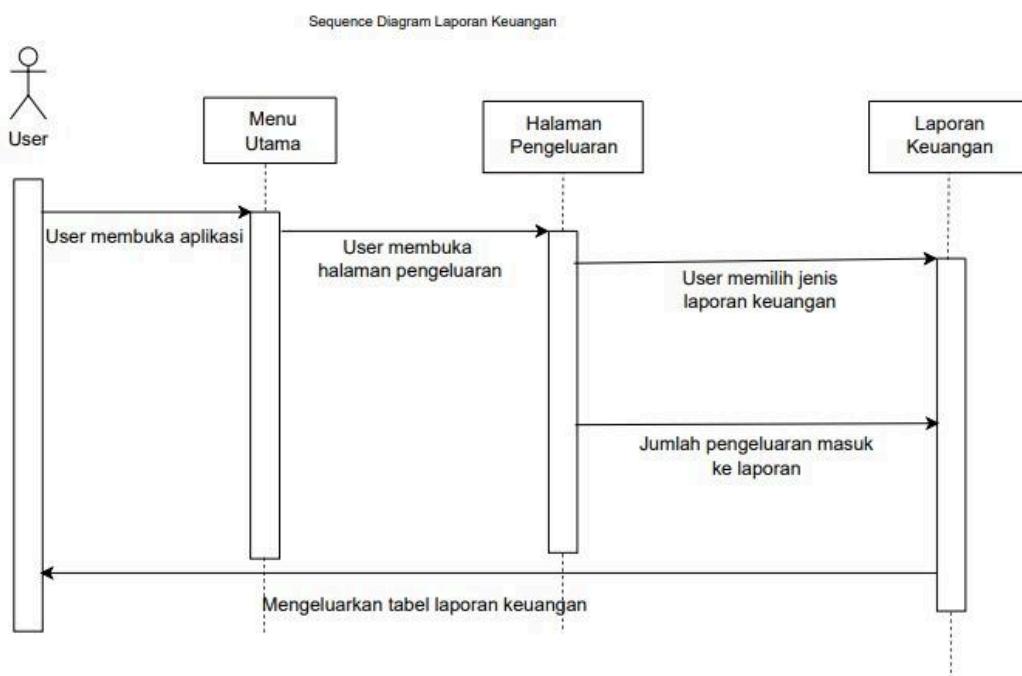
<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
Icon	Icon Back		Jika diklik, akan kembali kepada beranda
Button 1	Button	Jenis statistik transaksi	Jika diklik, akan membuka pilihan jenis statistik transaksi yakni mingguan, bulanan, tahunan
Button 2	Button	Pengurutan transaksi	Jika diklik, akan membuka pilihan jenis pengurutan transaksi yakni
RTF 1	RTF Box	Statistik Transaksi	Menampilkan grafik statistik dari transaksi pengguna
RTF 2	RTF Box	Most Popular Transaction	Menampilkan daftar transaksi yang paling sering pengguna lakukan

### 3.1.6.3 Identifikasi Object Baru & Tipe Kelas #6 <Laporan Keuangan>

TABEL OBJECT PERANCANGAN

<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1.	Tombol “Icon Back”	Controller
2.	Tombol Jenis Transaksi	Controller
3.	Tombol “Sort By”	Controller
4.	Transaction statistics	Boundary
5.	Popular Transaction	Boundary

### 3.1.6.4 Sequence Diagram #6 <Laporan Keuangan>



### **3.1.7 Use Case #7 <Peringatan Penggunaan>**

#### **3.1.7.1 Use Case Scenario #7 “Peringatan Penggunaan”**

Nama Use Case	Peringatan Penggunaan
Deskripsi	Aplikasi memberikan peringatan kepada user tentang pengeluaran yang mendekati batas anggaran.
Aktor	1. Aplikasi
Precondition	User sudah membuat rencana anggaran dalam aplikasi.
Postcondition	User menerima peringatan jika pengeluaran mendekati atau melebihi batas anggaran yang telah ditetapkan.
Normal Flow of event	<ol style="list-style-type: none"><li>1. Aplikasi secara rutin memeriksa pengeluaran user.</li><li>2. Jika pengeluaran mendekati atau melebihi batas anggaran yang telah ditetapkan, aplikasi mengirimkan peringatan kepada user.</li></ol>
Alternative Flow of event	

### 3.1.7.2 UI Design dan Deskripsi Objek UI #7 <Peringatan Penggunaan>



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
G0001	Page Peringatan Penggunaan	Menampilkan halaman peringatan penggunaan

G0001 : Page Identifikasi Pengeluaran

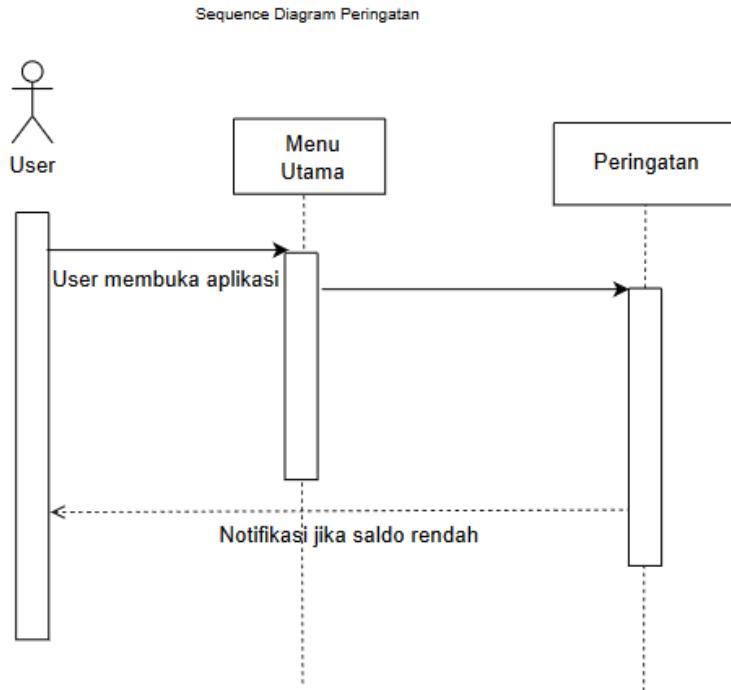
Id_Objek	JENIS	LABEL*	Keterangan**
Button 1	Button	Close	Jika diklik, akan menutup peringatan
RTF 1	RTF		Menampilkan peringatan pembayaran tagihan
RTF 2	RTF		Menampilkan peringatan sisa saldo minim

### 3.1.7.3 Identifikasi Object Baru & Tipe Kelas #7 <Peringatan Penggunaan>

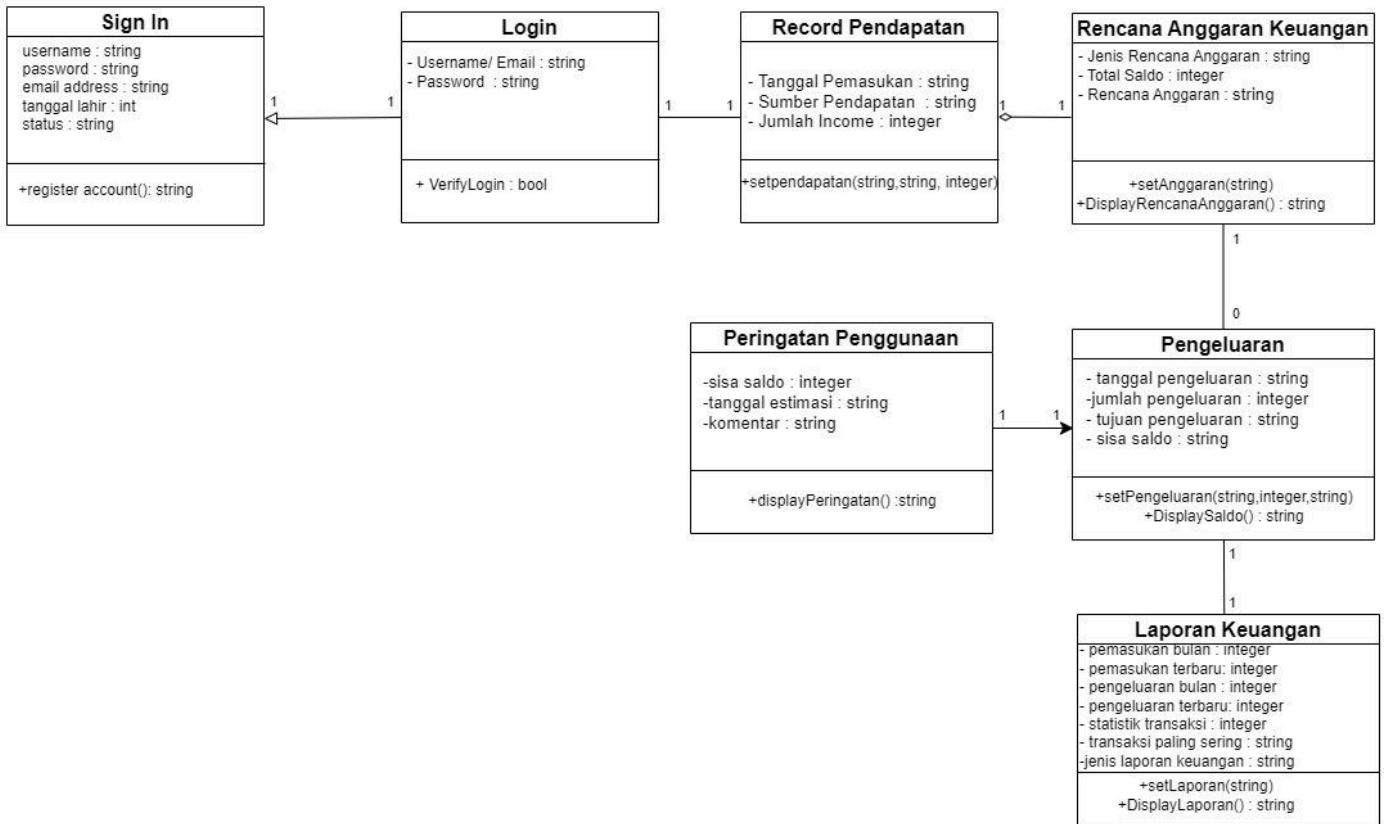
TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1.	Tombol "Close"	Controller

### 3.1.7.4 Sequence Diagram #7 <Peringatan Penggunaan>



#### 4. Diagram Kelas Keseluruhan



#### **4. 1 Perancangan Detail Kelas**

**TABEL KELAS :**

<b>ID Kelas</b>	<b>Nama Kelas Perancangan</b>	<b>Atribute (visibility)</b>	<b>Method / Operation</b>
	Sign In	Username/Email : String Password: String Tanggal lahir: Integer Status: String	register account(): string
	Log In	Username/Email: String Password: string	VerifyLogin: bool
	Rencana Anggaran Keuangan	Jenis rencana anggaran: String Total Saldo: Integer Rencana Anggaran: string	setAnggaran(string) DisplayRencanaAnggaran() : string
	Identifikasi Pengeluaran	Tanggal Pengeluaran: String Tujuan Pengeluaran: String Jumlah Pengeluaran: Integer Riwayat Pengeluaran: String	setPengeluaran(string,integer,string) DisplaySaldo() : string
	Record Pendapatan	Jenis Pengeluaran: String Tanggal Pemasukan: String Sumber Pendapatan: String Jumlah Income: Integer	setPendapatan(string,string,integer)
	Peringatan Pengguna	Sisa Saldo: Integer Tanggal Estimasi: String	displayPeringatan() :string
	Laporan Keuangan	Pemasukan Bulan : integer Pemasukan Terbaru: integer Pengeluaran Bulan : integer Pengeluaran Terbaru: integer Statistik Transaksi : integer Transaksi paling sering : string Jenis Laporan Keuangan :string	setLaporan(string) DisplayLaporan() : string

#### **4.2 Perancangan Algoritma dan/atau Query**

##### **Algoritma #1**

*Nama Kelas : Sign In*  
*Nama Operasi : doSignIn*

```
struct userData{
    string username;
    string password;
    string emailAddress;
    string birthDate;
    string status;
}

void doSignIn() {
```

```

displayRegistrationPage();
getUserInput();
if (validateInformation()) {
    createNewUserProfile();
} else {
    displayInvalidInformationMessage();
    getUserInput();
}
};


```

**Query :**

```

//membuat tabel pengguna
CREATE TABLE Users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    birth_date DATE NOT NULL,
    status VARCHAR(50) NOT NULL
);

//membuat akun
INSERT INTO Users (username, password, email, birth_date, status)
VALUES ('kiki', 'yta123', 'kikiaja@gmail.com', '2006-01-01', 'pelajar');


```

**Algoritma #2**

*Nama Kelas : Log In  
 Nama Operasi : doLogIn*

```

struct userData{
    string username;
    string password;
    string emailAddress;
    string birthDate;
    string status;
}

void doLogIn() {
    displayLoginPage();
    getUserInput();
    if(validateUser()) {
        grantAccess();
    } else {
        displayInvalidCredentialsMessage();
        getUserInput();
    }
}


```

```
    }  
};
```

**Query :**

```
// mengecek apakah data user sudah terdaftar  
SELECT * FROM Users  
WHERE username = 'kiki' AND password = 'ytta123';
```

**Algoritma #3**

*Nama Kelas : Record Pendapatan*  
*Nama Operasi : recordIncome*

```
struct RecordPendapatan {  
    int jumlah;  
    string sumber;  
    string tanggal;  
}  
  
void recordIncome() {  
    displayRecordIncomePage();  
    getUserInput();  
    saveIncomeRecord();  
};
```

**Query :**

```
// tabel untuk kelas record pendapatan  
CREATE TABLE Income (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    amount INT NOT NULL,  
    source VARCHAR(255) NOT NULL,  
    date DATE NOT NULL  
);  
  
// Menginputkan pemasukan  
INSERT INTO Income (amount, source, date)  
VALUES (500, 'Saham', '2024-01-07');
```

**Algoritma #4**

*Nama Kelas : Rencana Anggaran dan Keuangan*  
*Nama Operasi : budgetPlan*

```
struct RencanaAnggaranKeuangan {
```

```

    string jenisRencana;
    string tanggal;
    int jumlahPengeluaran;
    string jenisPengeluaran;
    int sisaSaldo;
}
void budgetPlan() {
    displayBudgetPlanPage();
    getUserInput();
    saveBudgetPlan();
    displayBudgetPlanReport();
}

```

**Query :**

```

//tabel Perencanaan anggaran Keuangan
CREATE TABLE BudgetPlan (
    id INT AUTO_INCREMENT PRIMARY KEY,
    jenis_rencana VARCHAR(255) NOT NULL,
    tanggal DATE NOT NULL,
    jumlah_pengeluaran INT NOT NULL,
    jenis_pengeluaran VARCHAR(255) NOT NULL,
    sisa_saldo INT NOT NULL
);

-- Insert Budget Plan
INSERT INTO BudgetPlan (jenis_rencana)
VALUES ('Mingguan');

```

**Algoritma #5**

Nama Kelas : Identifikasi Pengeluaran  
 Nama Operasi : identifySpending

```

struct IdentifikasiPengeluaran {
    string tanggalPengeluaran;
    int jumlahPengeluaran;
    string tujuanPengeluaran;
    int sisaSaldo;
}

void identifySpending() {
    displayIdentifySpendingPage();
    getUserInput();
    saveSpendingRecord();
}

```

```

Query :
//tabel pengeluaran user
CREATE TABLE pengeluaran (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tanggal_pengeluaran DATE NOT NULL,
    jumlah_pengeluaran INT NOT NULL,
    tujuan_pengeluaran VARCHAR(255) NOT NULL,
);

//memasukkan pengeluaran
INSERT INTO pengeluaran(tanggal_pengeluaran, jumlah_pengeluaran, tujuan_pengeluaran)
VALUES ('2024-01-07',5000000, 'Hiburan');

```

### **Algoritma #6**

*Nama Kelas : Laporan Keuangan  
 Nama Operasi : financialReport*

```

string LaporanKeuangan {
    string jenisLaporan;
    string tanggal;
    int jumlahPengeluaran;
    string jenisPengeluaran;
    int sisaSaldo;
}

void financialReport() {
    displayFinancialReportPage();
    getUserInput();
    displayFinancialReport();
}
};

```

### **Query :**

```

// tabel laporan keuangan
CREATE TABLE FinancialReport (
    id INT AUTO_INCREMENT PRIMARY KEY,
    jenis_laporan VARCHAR(255) NOT NULL,
    tanggal DATE NOT NULL,
    jumlah_pengeluaran INT NOT NULL,
    jenis_pengeluaran VARCHAR(255) NOT NULL,
    sisa_saldo INT NOT NULL

```

```

);
//user memilih jenis laporan keuangan
INSERT INTO FinancialReport (jenis_laporan)
VALUES ('Mingguan');

```

**Algoritma #7**

*Nama Kelas : Peringatan Penggunaan  
 Nama Operasi : warningMessage*

```

struct PeringatanPenggunaan {
    int sisaSaldo;
    string tanggalEstimasi;
    string komentar;
}

void warningMessage() {
    if(checkSpendingStatus()){
        displayWarningMessage();
        sendNotification();
    }
};

```

## 5. Matriks Kerunutan (Requirement Traceability Matrix)

Kode FR	Nama Functional Requirement	Nama Use Case
FR-01	Register	Sign in
FR-02	Login	Login
FR-03	Mengisi dan menambahkan pemasukan	Record pendapatan
FR-04	Memilih dan menampilkan, dan mengunduh Rancangan Anggaran Keuangan	Perencanaan Rancangan Keuangan
FR-05	Memasukkan dan Mengidentifikasi pengeluaran user	Identifikasi Pengeluaran
FR-06	Memberikan peringatan saat kelebihan dalam pengeluaran pengeluaran diluar rancangan anggaran	Peringatan penggunaan
FR-07	menampilkan laporan keuangan dan rewayat keuangan user	Laporan Keuangan

