

## BAB 3

### METODE PENELITIAN

Penelitian ini menggunakan metode Agile dengan pendekatan Kanban untuk mengatur alur kerja secara visual, efisien, dan terstruktur. Teknik ini diterapkan di semua tahap Agile mulai dari *Plan* hingga *Launch*, guna memantau perkembangan tugas, mengelola prioritas fitur, serta mendukung prinsip continuous delivery dalam Agile.

Berikut adalah penerapan metode Agile dengan pendekatan Kanban:

#### 1. *Plan*

Tahap *Plan* merupakan langkah awal untuk memahami kebutuhan pengguna dan menetapkan ruang lingkup aplikasi. Tahap *Plan* meliputi:

- a. Pengumpulan data literasi keuangan mahasiswa dari SNLIK 2024, dan perancangan awal fitur.
- b. Analisis permasalahan dan penentuan fitur utama.
- c. Identifikasi kebutuhan pengguna dan penentuan fitur akhir berdasarkan hasil analisis.

#### 2. *Design*

Tahap *Design* merupakan proses perancangan sistem berdasarkan kebutuhan yang telah dikumpulkan. Tahap *Design* meliputi:

- a. Mendesain UI/UX dan menyusun arsitektur sistem.
- b. Mendesain antarmuka menggunakan Figma, dan merancang struktur basis data dengan Supabase.
- c. UI/UX selesai dalam bentuk wireframe, dan struktur database selesai disusun.

#### 3. *Develop*

Tahap *Develop* merupakan proses pengembangan sistem berdasarkan desain yang telah dibuat. Tahap *Develop* meliputi:

- a. Menyusun dan merancang fitur login, register, kategori, transaksi, pos anggaran, laporan, dan download.

- b. Implementasi dan pengembangan masing-masing fitur.
- c. Seluruh fitur berhasil diimplementasikan.

#### 4. *Test*

Tahap *Test* merupakan proses pengujian sistem yang telah dikembangkan, pengujian dilakukan dengan memastikan fungsionalitas dan pengalaman pengguna. Tahap *Test* meliputi:

- a. Menyusun skenario pengujian, rencana uji UX, dan kuesioner evaluasi pengguna.
- b. Melakukan pengujian fungsional dan pengalaman pengguna, serta perbaikan berdasarkan umpan balik.
- c. Semua fitur telah lulus pengujian, dan perbaikan akhir berdasarkan evaluasi telah dilakukan.

#### 5. *Deploy*

Tahap *Deploy* merupakan proses implementasi sistem pada lingkungan pengguna nyata setelah sistem dinyatakan stabil. Tahap *Deploy* meliputi:

- a. Mempersiapkan sistem untuk implementasi.
- b. Implementasi sistem ke lingkungan pengguna.
- c. Sistem berhasil diterapkan di lingkungan pengguna.

#### 6. *Review*

Tahap *Review* merupakan proses evaluasi terhadap umpan balik dan saran pengguna akhir. Tahap *Review* meliputi:

- a. Menganalisis hasil implementasi pada penerapan ke lingkungan pengguna.
- b. Mengevaluasi kekurangan dan kelebihan aplikasi.
- c. Menyusun daftar masukan perbaikan dari umpan balik dan saran pengguna.

#### 7. *Launch*

- a. Tahap *Launch* merupakan proses distribusi akhir sistem kepada pengguna secara luas. Tahap *Launch* meliputi:
- b. Persiapan akhir untuk distribusi aplikasi.
- c. Finalisasi aplikasi berdasarkan hasil review.

- d. Sistem diluncurkan dan siap digunakan oleh pengguna akhir.

### 3.1 BAHAN DAN ALAT PENELITIAN

Bahan penelitian yang digunakan dalam pengembangan sistem manajemen keuangan mahasiswa berbasis mobile meliputi:

1. Informasi Literasi Keuangan:

Referensi mengenai indeks literasi keuangan mahasiswa dari Survei Nasional Literasi dan Inklusi Keuangan (SNLIK) 2024 oleh Otoritas Jasa Keuangan (OJK).

2. Desain UI/UX aplikasi:

Desain antarmuka pengguna yang dikembangkan berdasarkan studi literatur, preferensi pengguna, serta prinsip desain yang memudahkan mahasiswa dalam mencatat dan memantau keuangan mereka.

3. Data Uji Coba Pengguna:

Data simulasi berupa akun pengguna yang mencerminkan profil mahasiswa dengan berbagai kondisi keuangan, yang digunakan untuk menguji fungsionalitas sistem.

Alat yang digunakan dalam pengembangan sistem ini meliputi:

1. Perangkat Keras (*Hardware*):

Laptop/Komputer dengan spesifikasi:

- a. Prosesor: Chip Apple M1.
- b. RAM: 8 GB.
- c. Penyimpanan: 256 GB SSD.
- d. Perangkat *Mobile* (*Smartphone*): Digunakan untuk menguji fungsionalitas sistem.

2. Perangkat Lunak (*Software*):

- a. Sistem Operasi: macOS.
- b. *Framework* dan Bahasa Pemrograman: Flutter (versi terbaru) sebagai *framework* utama untuk pengembangan aplikasi *mobile*. Dart 3.102.0 sebagai bahasa pemrograman utama.
- c. *Database*: Supabase sebagai sistem manajemen basis data (DBMS).

- d. Alat Pengembangan:
  - a) Figma.
  - b) Visual Studio Code.
  - c) Git.
  - d) Xcode.

### 3.2 JALAN PENELITIAN

Penelitian ini menggunakan metode Agile dengan pendekatan kanban, dalam pengembangan aplikasi manajemen keuangan mahasiswa berbasis *mobile*. Berikut proses pengembangannya:

1. Identifikasi Kebutuhan Pengguna
  - a. Informasi Survei Nasional Literasi dan Inklusi Keuangan (SNLIK) 2024 yang dirilis oleh Otoritas Jasa Keuangan (OJK) tahun 2024.
  - b. Menganalisis permasalahan mahasiswa dalam mengatur keuangan dan menentukan fitur aplikasi yang relevan.
2. Perancangan Sistem
  - a. Menyusun desain UI/UX berdasarkan hasil analisis kebutuhan pengguna.
  - b. Merancang arsitektur sistem dan basis data menggunakan Flutter dan Supabase.
3. Pengembangan Aplikasi (Kanban)
  - a. Visualisasi tugas-tugas ke dalam papan Kanban (To Do, Doing, dan Done).
  - b. Setiap fitur utama seperti login, register, kategori, tambah transaksi, pos anggaran, dan laporan keuangan dimasukkan sebagai kartu tugas
  - c. Pengerjaan dilakukan satu per satu, memprioritaskan fitur inti terlebih dahulu.
  - d. Setelah fitur selesai, dilakukan pengujian internal dan evaluasi pengguna, lalu diterapkan perbaikan berdasarkan umpan balik.
  - e. Perubahan dan penyempurnaan dilakukan secara berkelanjutan (continuous delivery) hingga seluruh fitur terpenuhi.

4. Pengujian dan Evaluasi
  - a. Melakukan pengujian fungsional untuk memastikan setiap fitur bekerja sesuai spesifikasi.
  - b. Melakukan pengujian pengalaman pengguna akhir untuk menilai kemudahan penggunaan aplikasi.
  - c. Mengumpulkan umpan balik dari pengguna untuk perbaikan dan penyempurnaan sistem.
5. Penyempurnaan dan Implementasi Akhir
  - a. Melakukan perbaikan dan optimasi berdasarkan hasil evaluasi dan pengujian.
  - b. Mempersiapkan sistem untuk tahap implementasi akhir dan distribusi kepada pengguna.

### 3.3 PERANCANGAN SISTEM

Perancangan sistem adalah proses menentukan bagaimana suatu sistem informasi akan dirancang untuk memenuhi kebutuhan fungsional dan non-fungsional yang telah diidentifikasi pada tahap analisis sistem, termasuk arsitektur, komponen, modul, antarmuka, dan data yang diperlukan (Nirsal et al., 2025).

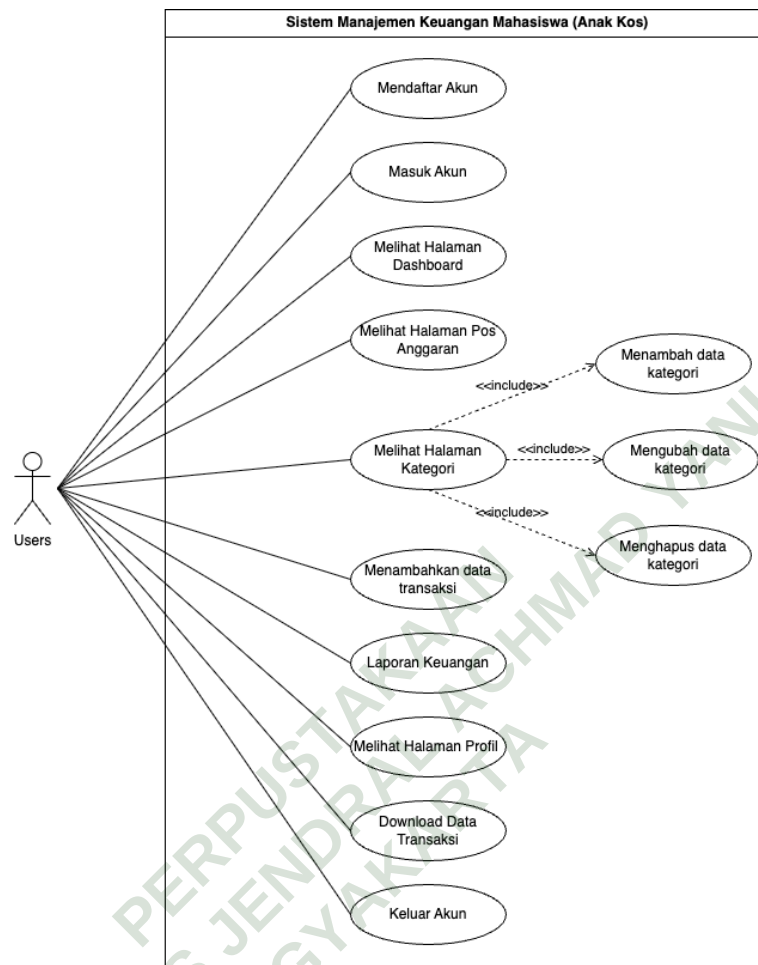
Perancangan sistem adalah strategi untuk memecahkan masalah saat ini untuk mendapatkan solusi terbaik dan mencapai tujuan (Rusmawan, 2019).

#### 3.3.1 Unified Modeling Language (UML)

UML merupakan bahasa pemodelan untuk menggambarkan, memvisualisasikan, dan mendokumentasikan sistem perangkat lunak. UML menyediakan berbagai jenis diagram yang menjelaskan aspek statis dan dinamis dari suatu sistem (Nirsal et al., 2025).

##### 1. Use Case Diagram

*Use case diagram* adalah salah satu variasi dari berbagai bentuk dan jenis diagram UML yang menggambarkan atau menjelaskan hubungan antara sistem dan aktor (Ahmad et al., 2022). *Use case diagram* pada Sistem Manajemen Keuangan Mahasiswa dapat dilihat pada Gambar 3.1.



**Gambar 3.1** Use Case Diagram

Penjelasan tentang kegiatan aktor pada sistem dapat dilihat pada Tabel 3.1. Aktor user merupakan Mahasiswa.

**Tabel 3.1** Penjelasan Use Case Diagram

| Aktor | Use Case                  | Deskripsi                                                       |
|-------|---------------------------|-----------------------------------------------------------------|
| User  | Mendaftar Akun            | Aktivitas yang menggambarkan proses pendaftaran akun            |
|       | Masuk Akun                | Aktivitas yang menggambarkan proses masuk/login ke dalam system |
|       | Melihat Halaman Dashboard | Aktivitas yang menggambarkan proses melihat halaman dashboard   |

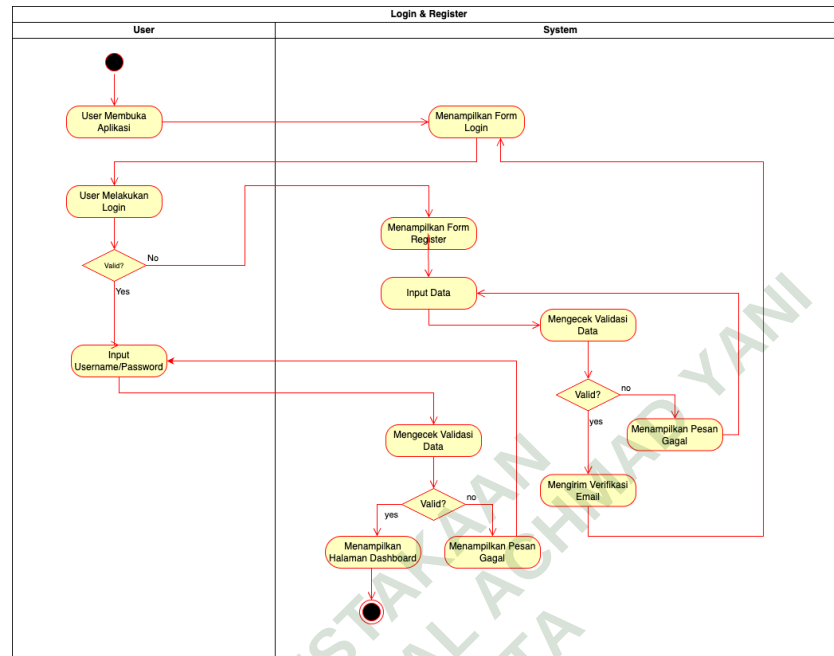
|  |                              |                                                                  |
|--|------------------------------|------------------------------------------------------------------|
|  | Melihat Halaman Pos Anggaran | Aktivitas yang menggambarkan proses melihat halaman pos anggaran |
|  | Menambahkan Data Kategori    | Aktivitas yang menggambarkan proses menambahkan data kategori    |
|  | Mengubah Data Kategori       | Aktivitas yang menggambarkan proses mengubah data kategori       |
|  | Menghapus Data Kategori      | Aktivitas yang menggambarkan proses menghapus data kategori      |
|  | Menambahkan Data Transaksi   | Aktivitas yang menggambarkan proses menambahkan data transaksi   |
|  | Laporan Keuangan             | Aktivitas yang menggambarkan proses melihat laporan keuangan     |
|  | Melihat Halaman Profil       | Aktivitas yang menggambarkan proses melihat halaman profil       |
|  | Download Data Transaksi      | Aktivitas yang menggambarkan proses download data transaksi      |
|  | Keluar Akun                  | Aktivitas yang menggambarkan proses keluar akun                  |

## 2. Activity Diagram

*Activity Diagram* adalah diagram yang menggambarkan alur kerja berbagai aktivitas user atau sistem, orang yang melakukannya, dan aliran berurutan dari aktivitas ini (Destriana et al., 2022).

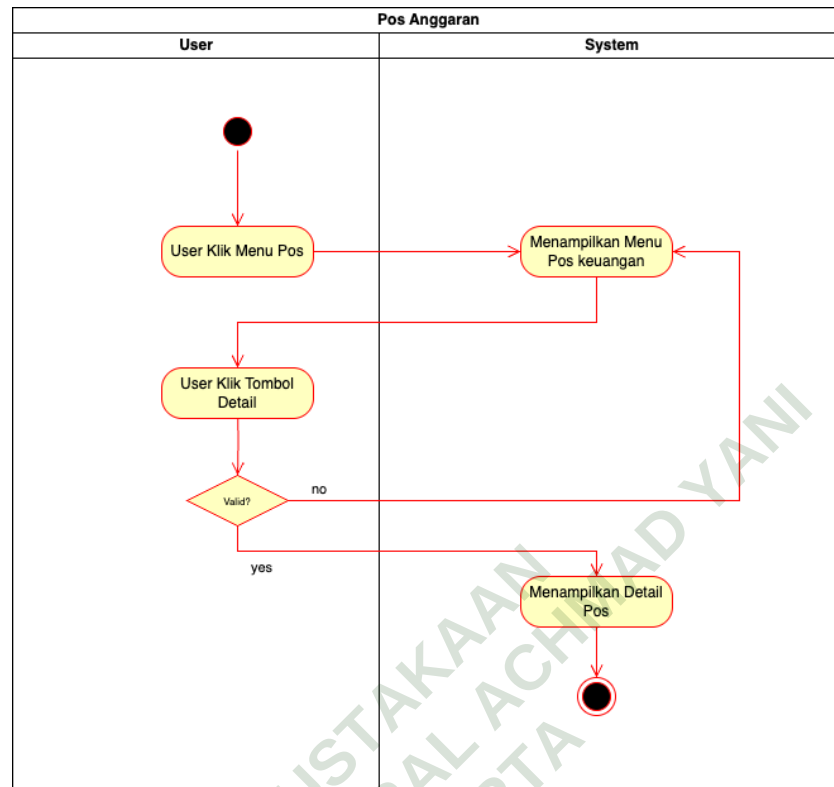
Gambar 3.2 menjelaskan *Activity Diagram login dan Register* yang dapat diakses oleh Mahasiswa sebagai *User*. *User* memulai dengan mendaftarkan akunnya, dan sistem mengirim email konfirmasi kepada pengguna agar akunnya dapat digunakan untuk masuk ke sistem. Jika data yang dimasukkan tidak sesuai, akan kembali ke *form* registrasi. Dari akun yang

didaftarkan, pengguna dapat masuk ke sistem dengan menggunakan *email* dan *password*.



**Gambar 3.2** Activity Diagram Login & Register

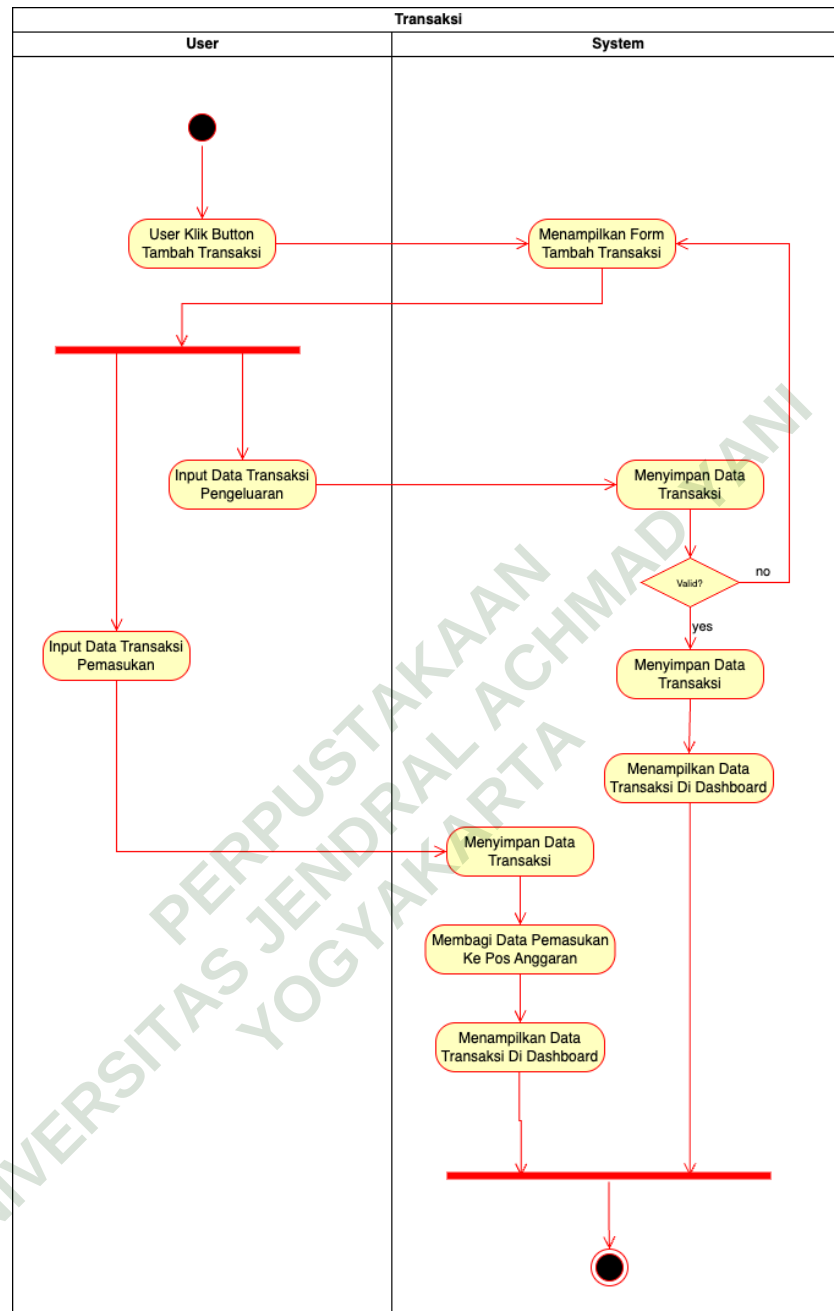
Pada Gambar 3.3 menjelaskan Activity Diagram Pos Anggaran yang dapat diakses oleh Mahasiswa sebagai *User* untuk melihat informasi dana dan alokasi anggaran mereka.



**Gambar 3.3** Activity Diagram Pos Anggaran

Pada Gambar 3.4 menjelaskan *Activity Diagram* Kategori, yang dapat dilakukan pengguna untuk mengelola data kategori, yang mencakup aktivitas menambahkan, mengubah, dan menghapus kategori. Proses diawali ketika pengguna memilih menu kategori, kemudian sistem menampilkan daftar kategori. Saat pengguna ingin menambah data, sistem menampilkan form tambah data, pengguna mengisi data dan menekan tombol simpan, kemudian sistem akan menyimpan data tersebut. Untuk mengubah kategori, sistem menampilkan form ubah data, pengguna memasukkan perubahan, dan setelah mengklik simpan, sistem menyimpan perubahan tersebut. Jika pengguna ingin menghapus kategori, sistem menampilkan dialog konfirmasi, dan setelah pengguna menekan tombol hapus, sistem menghapus data kategori.

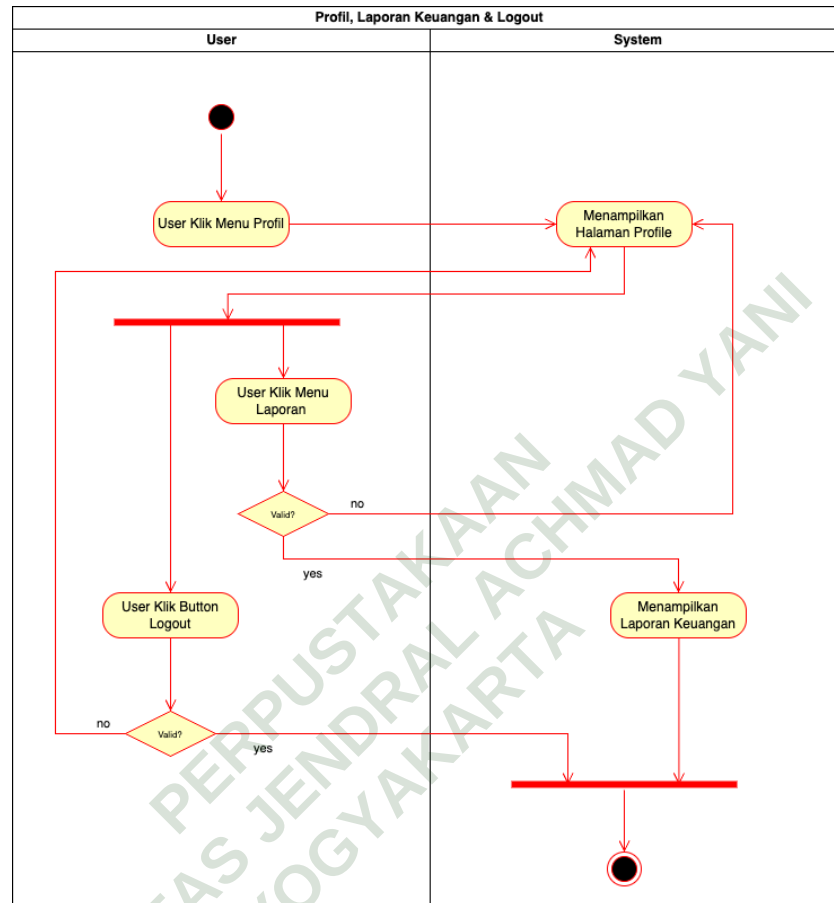




**Gambar 3.5** Activity Diagram Transaksi

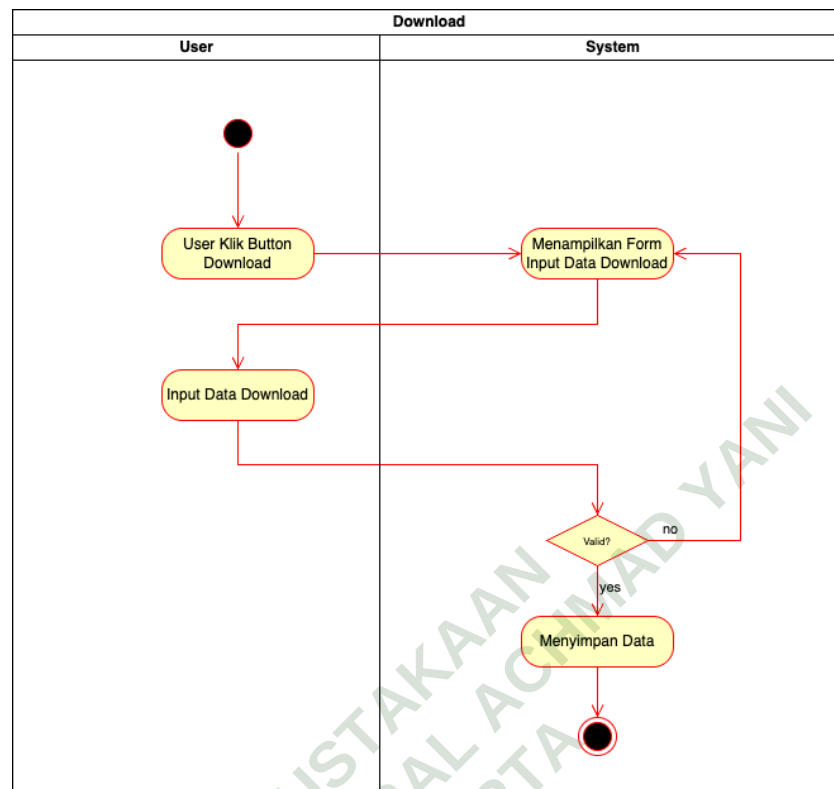
Gambar 3.6 menjelaskan *Activity Diagram* profil, laporan keuangan, dan *logout*. Jika user klik menu profil maka sistem akan menampilkan halaman profil. Jika pengguna memilih menu laporan dan input yang diberikan benar, sistem akan menampilkan laporan keuangan. Setelah itu, ketika pengguna

menekan tombol logout dan prosesnya berhasil, sistem akan mengakhiri sesi dan kembali ke *login*.



**Gambar 3.6** Activity Diagram Profil, Laporan Keuangan & Logout

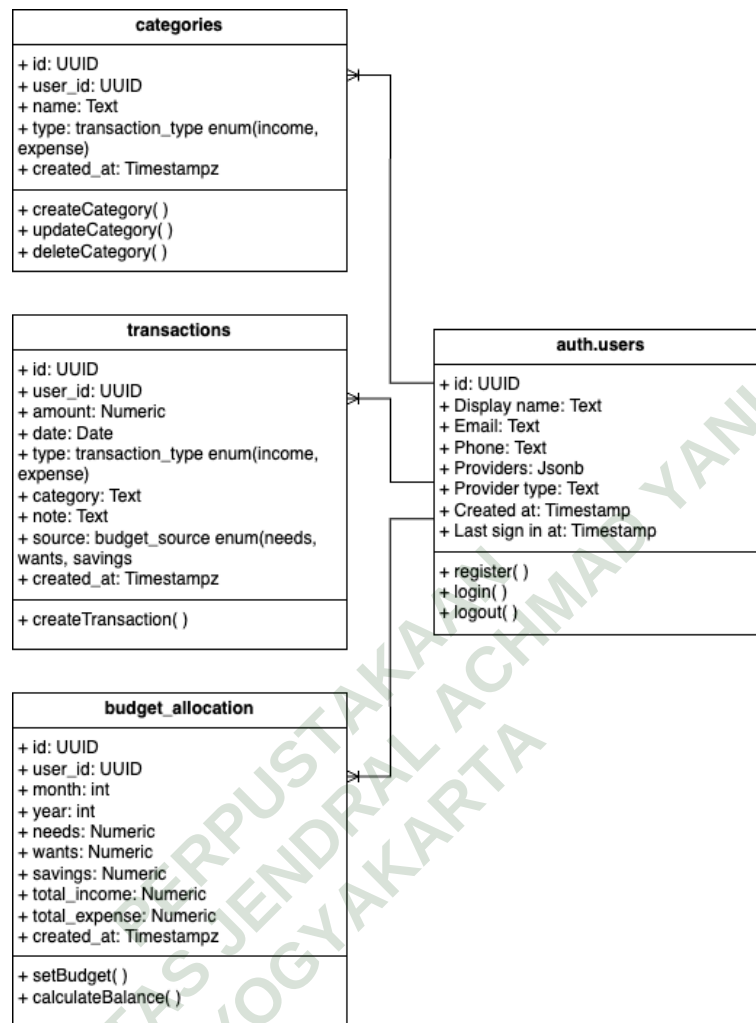
Pada Gambar 3.7 menjelaskan Activity Diagram download oleh user. Jika user klik tombol *download* maka sistem menampilkan *form download*. Selanjutnya, user mengisi form tersebut. Kemudian sistem akan memeriksa data jika data sesuai maka data akan disimpan, jika tidak sesuai maka akan kembali ke *form download*.



**Gambar 3.7** Activity Diagram Download

### 3. Class Diagram

*Class Diagram* adalah diagram yang menunjukkan struktur statis sistem dan menunjukkan *class-class* dan hubungan logikanya (Cahyani et al., 2020). Class diagram pada istem ini dapat dilihat pada Gambar 3.8.



**Gambar 3.8** Class Diagram

### 3.3.2 Rancangan Database

Rancangan *Database* berupa struktur tabel untuk membedakan tabel satu sama lain. Setiap tabel memiliki *primary key* (PK), yang juga berfungsi sebagai *foreign key* (FK) untuk tabel lain. Berikut struktur tabel yang terdapat pada *database*.

#### 1. Tabel *auth.users*

Tabel *auth.users* merupakan tabel bawaan dari Supabase yang secara otomatis disediakan untuk menyimpan data pengguna yang terdaftar melalui sistem autentikasi Supabase, seperti email, nomor telepon, waktu pembuatan akun, dan informasi login pengguna. Berikut struktur tabelnya dapat dilihat pada Tabel 3.2.

**Tabel 3.2** Tabel *auth.users*

| auth.users |                 |           |            |
|------------|-----------------|-----------|------------|
| No         | Atribut         | Tipe Data | Keterangan |
| 1          | Id              | UUID      | PK         |
| 2          | Display name    | Text      |            |
| 3          | Email           | Text      |            |
| 4          | Phone           | Text      |            |
| 5          | Providers       | Jsonb     |            |
| 6          | Provider type   | Text      |            |
| 7          | Created at      | Timestamp |            |
| 8          | Last sign in at | Timestamp |            |

## 2. Tabel *categories*

Tabel *categories* berisi daftar kategori transaksi yang dimiliki masing-masing pengguna. Atribut *user\_id* berfungsi sebagai *foreign key* (FK), yang mengarah ke tabel *auth.users*. Setiap kategori memiliki nama dan tipe yang terdiri dari enum *income* dan *expense*. Karena relasi dengan tabel *auth.users* bersifat *many-to-one*, maka satu pengguna dapat memiliki banyak kategori. Tabel *categories* dapat dilihat pada Tabel 3.3.

**Tabel 3.3** Tabel *categories*

| categories |            |                                           |            |
|------------|------------|-------------------------------------------|------------|
| No         | Atribut    | Tipe Data                                 | Keterangan |
| 1          | Id         | UUID                                      | PK         |
| 2          | User_id    | UUID                                      | FK         |
| 3          | Name       | Text                                      |            |
| 4          | Type       | Transaction_type<br>enum(income, expense) |            |
| 5          | Created_at | Timestampz                                |            |

## 3. Tabel *transactions*

Tabel *transactions* adalah tabel utama yang mencatat semua aktivitas keuangan pengguna. Atribut *user\_id* berfungsi sebagai *foreign key* (FK) yang

mengarah ke tabel *auth.users*, menunjukkan pemilik transaksi. Setiap transaksi menyimpan jumlah dana, tanggal, tipe berupa enum *income* atau *expense*, kategori, catatan, serta sumber dana yang berupa enum *needs*, *wants*, atau *savings*. Pada transaksi pemasukan dana akan otomatis dibagi ke dalam tiga sumber dana dengan metode 50/30/20, yaitu 50% untuk *needs*, 30% untuk *wants*, dan 20% untuk *savings*. Sedangkan untuk transaksi pengeluaran, pengguna memilih sendiri sumber dananya tanpa pembagian. Tabel *transactions* dapat dilihat pada Tabel 3.4.

**Tabel 3.4** Tabel transactions

| transactions |               |                                              |            |
|--------------|---------------|----------------------------------------------|------------|
| No           | Atribut       | Tipe Data                                    | Keterangan |
| 1            | Id            | UUID                                         | PK         |
| 2            | User_id       | UUID                                         | FK         |
| 3            | Category_id   | UUID                                         | FK         |
| 4            | Allocation_id | UUID                                         | FK         |
| 3            | Amount        | Numeric                                      |            |
| 4            | Date          | date                                         |            |
| 5            | Type          | Transaction_type<br>enum(income, expense)    |            |
| 6            | Category      | Text                                         |            |
| 7            | Note          | Text                                         |            |
| 8            | Source        | Budget_source enum(needs,<br>wants, savings) |            |
| 9            | Created_at    | Timestampz                                   |            |

#### 4. Tabel *budget\_allocation*

Tabel *budget\_allocation* menyimpan data alokasi anggaran bulanan setiap pengguna. Atribut *user\_id* berfungsi sebagai *foreign key* (FK) yang mengarah ke tabel *auth.users*. Setiap data mencakup bulan, tahun, serta nilai anggaran untuk *needs*, *wants*, dan *savings*, yang dihitung berdasarkan transaksi pemasukan. Tabel *budget\_allocation* dapat dilihat pada Tabel 3.5.

**Tabel 3.5** Tabel budget\_allocation

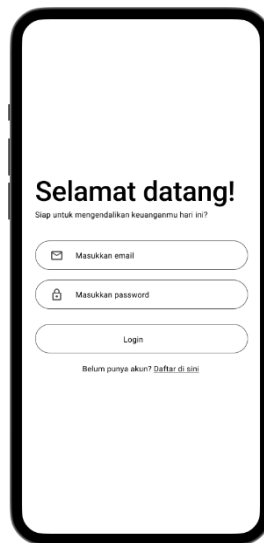
| Budget_allocation |               |            |            |
|-------------------|---------------|------------|------------|
| No                | Atribut       | Tipe Data  | Keterangan |
| 1                 | Id            | UUID       | PK         |
| 2                 | User_id       | UUID       | FK         |
| 3                 | Month         | int        |            |
| 4                 | Year          | int        |            |
| 5                 | Needs         | Numeric    |            |
| 6                 | Wants         | Numeric    |            |
| 7                 | Savings       | Numeric    |            |
| 8                 | Total_income  | Numeric    |            |
| 9                 | Total_expense | Numeric    |            |
| 10                | Created_at    | Timestampz |            |

### 3.4 PERANCANGAN *INTERFACE*

Perancangan *interface* adalah proses desain antarmuka pengguna untuk perangkat lunak yang berfokus pada tampilan dan interaksi. Tujuannya adalah untuk membuat interaksi pengguna efisien dan mudah dipahami dengan tampilan antarmuka yang memungkinkan akses mudah ke fungsi perangkat lunak. Pemilihan skema warna, desain tombol, pengaturan tata letak, dan elemen visual lainnya adalah bagian dari proses ini (Umar et al., 2024).

#### 3.4.1 Desain Halaman *Login*

Pada halaman login terdapat *form input email* dan *password*, dan *button login*, serta terdapat navigasi ke halaman *register*. Desain halaman *login* dapat dilihat pada Gambar 3.9.



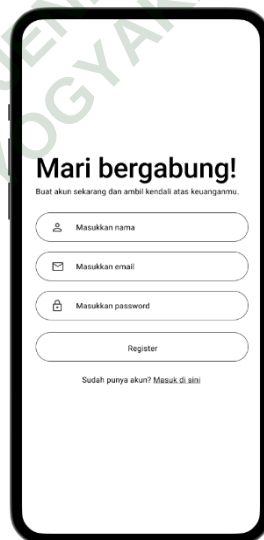
**Selamat datang!**  
Siap untuk mengendalikan keuanganmu hari ini?

Belum punya akun? [Daftar di sini](#)

**Gambar 3.9** Desain halaman *login*

### 3.4.2 Desain Halaman *Register*

Desain halaman *register* berisi formulir untuk *input* data nama, email, dan password, serta navigasi ke halaman *login*. Desain halaman *register* dapat dilihat pada Gambar 3.10.



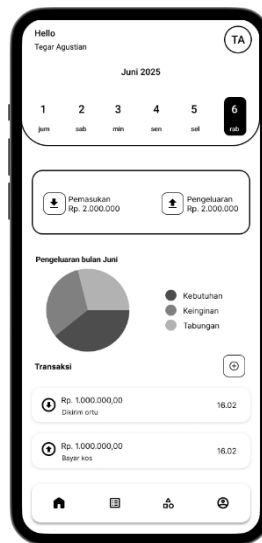
**Mari bergabung!**  
Buat akun sekarang dan ambil kendali atas keuanganmu.

Sudah punya akun? [Masuk di sini](#)

**Gambar 3.10** Desain halaman *register*

### 3.4.3 Desain Halaman *Dashboard*

Desain halaman *dashboard* menampilkan informasi total pemasukan dan pengeluaran harian, diagram lingkaran sumber pengeluaran, serta list transaksi harian. Desain halaman *dashboard* dapat dilihat pada Gambar 3.11.



**Gambar 3.11** Desain halaman *dashboard*

#### 3.4.4 Desain Halaman Pos Anggaran

Gambar 3.12 menampilkan desain halaman pos anggaran dengan informasi total dana yang tersedia dari setiap sumber anggaran, serta total dana keseluruhan.



**Gambar 3.12** Desain halaman pos anggaran

#### 3.4.5 Desain Halaman Kategori

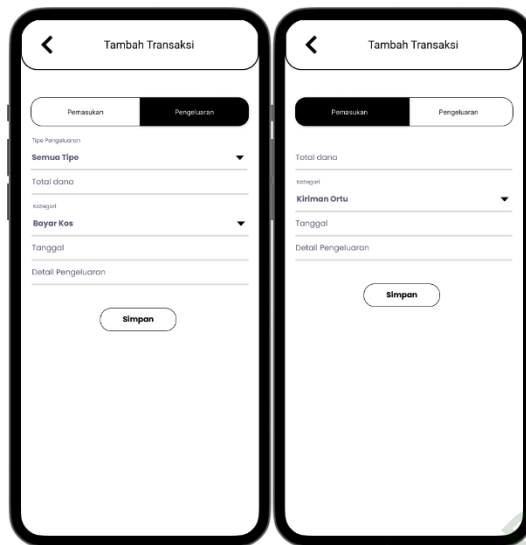
Gambar 3.13 menampilkan desain halaman kategori yang berisi list kategori, *form* tambah data kategori, *form* edit kategori, serta dialog konfirmasi hapus data kategori.



**Gambar 3.13** Desain halaman kategori

### 3.4.6 Desain Halaman Tambah Transaksi

Gambar 3.14 menampilkan desain halaman tambah transaksi pemasukan dan pengeluaran.



**Gambar 3.14** Desain halaman tambah transaksi

### 3.4.7 Desain Halaman Profil

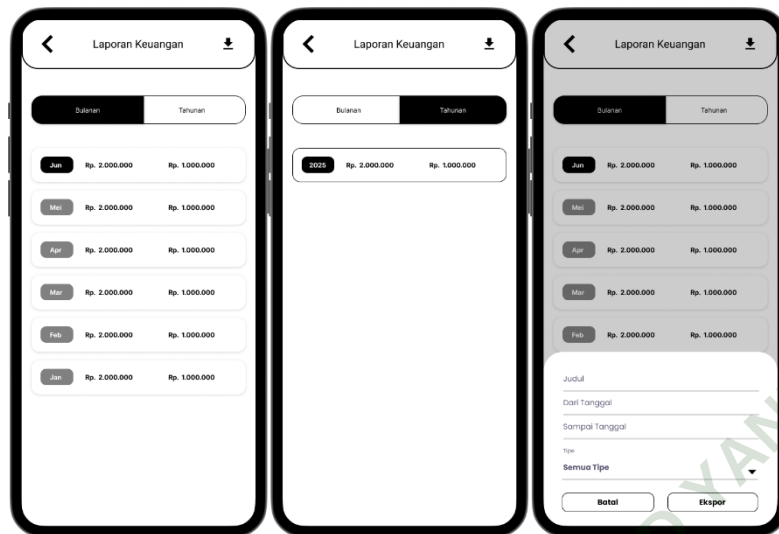
Gambar 3.15 menampilkan desain halaman profil dengan informasi *user*, navigasi ke halaman laporan keuangan, serta *button logout* untuk keluar dari sistem.



**Gambar 3.15** Desain halaman profil

### 3.4.8 Desain Halaman Laporan Keuangan

Pada halaman laporan keuangan dapat melihat data laporan keuangan bulanan dan tahunan, serta dapat mengunduh data transaksi sesuai filter tanggal. Desain halaman laporan keuangan dapat dilihat pada Gambar 3.16.



Gambar 3.16 Desain laporan keuangan