

BAB 3

METODOLOGI PENELITIAN

3.1 Studi Literatur

Studi Literatur, salah satu metode untuk mendapatkan referensi dari penelitian sebelumnya. Dalam melakukan penelitian ini, pengumpulan bahan pustaka menjadi fokus utama. Beberapa jurnal yang relevan telah dijadikan referensi dalam pengembangan aplikasi pada penelitian ini. Peneliti melakukan studi literatur dengan mengumpulkan informasi dari penelitian sebelumnya serta mempelajari berbagai literatur dan dokumen seperti buku, jurnal, dan teori-teori lain yang mendukung konsep aplikasi manajemen keuangan pada perangkat bergerak berbasis Android.

3.2 Pengumpulan Data

Pengumpulan data merupakan proses di mana peneliti mencari informasi yang relevan di lapangan untuk menjawab pertanyaan penelitian. Kualitas data sangat dipengaruhi oleh cara pengumpulan dan kualifikasi orang yang melakukan pengumpulan data. Dalam penelitian ini, peneliti akan mencari dan mengumpulkan data serta informasi terkait melalui survei. Survei dilakukan untuk mendapatkan tanggapan dan respons langsung dari mahasiswa yang menyewa tempat tinggal terkait dengan pengeluaran rutin mereka. Peneliti bertujuan untuk mengumpulkan data tentang pola pengeluaran rutin, kebutuhan keuangan, serta preferensi terkait fitur-fitur yang diinginkan dalam aplikasi manajemen keuangan berbasis Android yang sedang dikembangkan.

3.3 Metode Pengembangan Aplikasi

Metode pengembangan aplikasi manajemen keuangan personal berbasis Android menggunakan pendekatan *Waterfall* dipilih karena keunggulannya ketika seluruh kebutuhan sistem telah ditetapkan dengan jelas dan lengkap. Hal ini memastikan bahwa perancangan dan pengembangan aplikasi dapat berjalan dengan lancar dan efisien. Selain kemudahan dalam penjelasan, pendekatan *Waterfall* juga memberikan kerangka kerja yang terstruktur dan terorganisir untuk mengelola proyek pengembangan perangkat lunak. Tahapan dan langkah-langkah dalam

pengembangan sistem pada penelitian ini dapat diuraikan sebagai berikut.

3.3.1 Analisis

Pada tahap analisis, dilakukan pengumpulan data yang diperlukan untuk menganalisis kebutuhan pengguna secara lebih mendalam. Proses ini menggunakan metode survei untuk memperoleh pemahaman yang komprehensif tentang kebutuhan pengguna serta lingkungan sistem yang akan dikembangkan. Data yang terkumpul dari survei tersebut akan menjadi dasar untuk merumuskan persyaratan yang jelas dan spesifik untuk pengembangan aplikasi. Selain itu, pengumpulan data juga membantu dalam mengidentifikasi masalah atau tantangan yang mungkin dihadapi selama proses pengembangan aplikasi, sehingga solusi yang tepat dapat dirancang dan diimplementasikan. Dengan demikian, pengumpulan data memainkan peran penting dalam memastikan keberhasilan implementasi metode *waterfall* dalam pengembangan aplikasi.

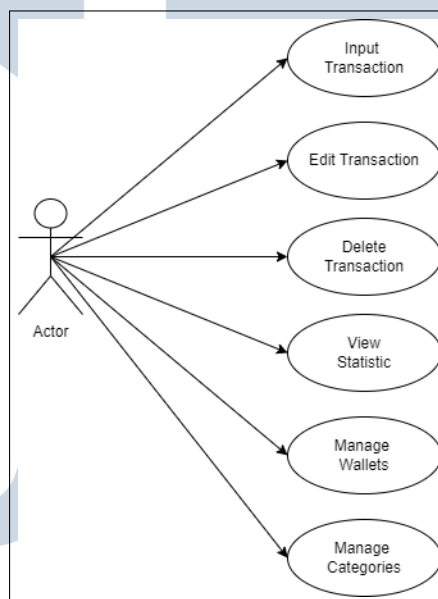


3.3.2 Perancangan Aplikasi

Pada tahap ini dilakukan pembuatan diagram UML yang memvisualisasikan alur kerja sistem secara keseluruhan. Selain itu, juga dilakukan perancangan antarmuka pengguna aplikasi, yang mencakup desain tampilan, navigasi, dan interaksi pengguna agar dapat memastikan pengalaman pengguna yang optimal.

A Use Case Diagram

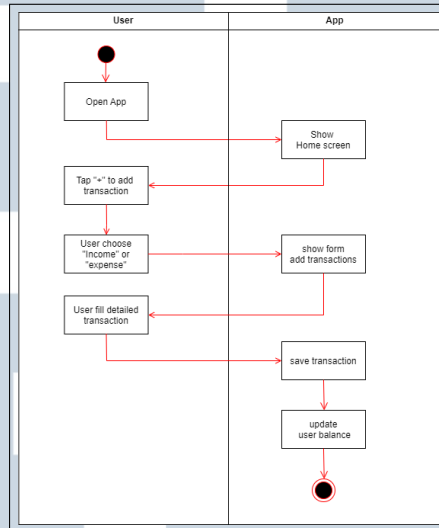
Diagram use case menggambarkan sistem dari perspektif penggunanya, menekankan pada fungsi-fungsi yang sistem tawarkan, bukan pada urutan peristiwa. Komponen utamanya meliputi aktor (pemain), use case (fungsi yang dilakukan), dan relasi yang menghubungkan keduanya. Berikut merupakan diagram use case untuk aplikasi yang akan dibuat. Use case diagram untuk aplikasi manajemen keuangan yang sedang dikembangkan dapat dilihat pada Gambar 3.1.



Gambar 3.1. Use Case Diagram

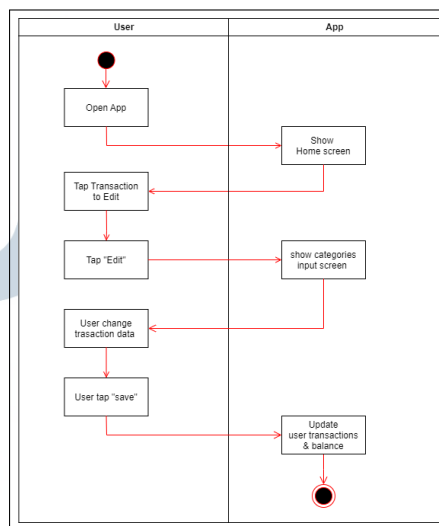
B Activity Diagram

Activity diagram pada Gambar 3.2 menunjukkan alur proses yang terjadi ketika seorang pengguna menambahkan transaksi baru ke dalam aplikasi.



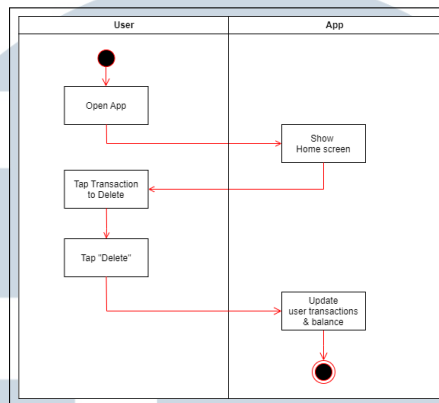
Gambar 3.2. Activity Diagram Tambah Transaksi

Selanjutnya, pada Gambar 3.3 ditunjukkan activity diagram yang menggambarkan proses pengguna mengubah transaksi.



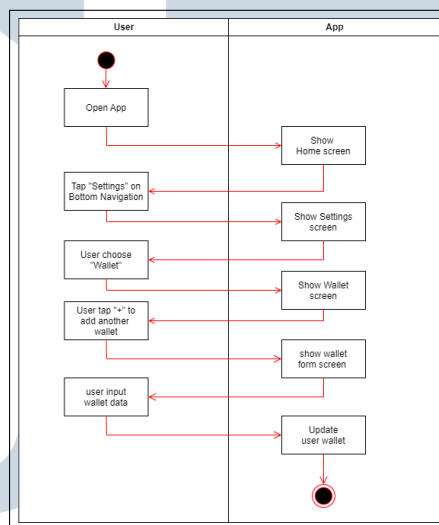
Gambar 3.3. Activity Diagram Edit Transaksi

Selanjutnya, pada Gambar 3.4 ditunjukkan activity diagram yang menggambarkan proses pengguna menghapus transaksi.



Gambar 3.4. Activity Diagram Delete Transaksi

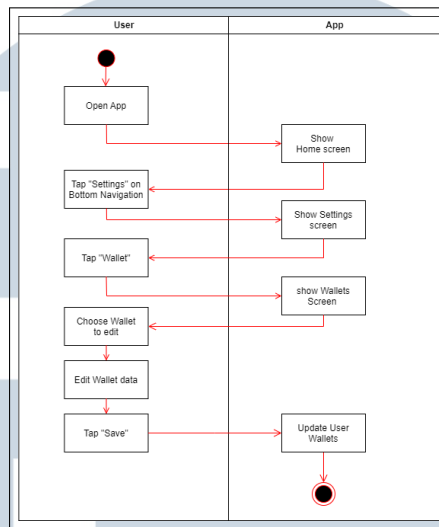
Selanjutnya, pada Gambar 3.5 ditunjukkan activity diagram yang menggambarkan proses pengguna menambahkan dompet baru ke dalam aplikasi.



Gambar 3.5. Activity Diagram Add Dompet

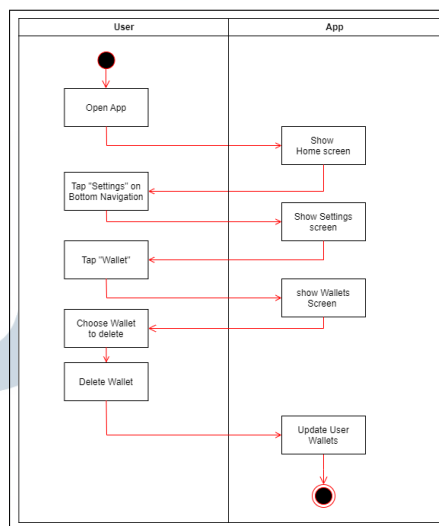
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Selanjutnya, pada Gambar 3.6 ditunjukkan activity diagram yang menggambarkan proses pengguna mengubah dompet.



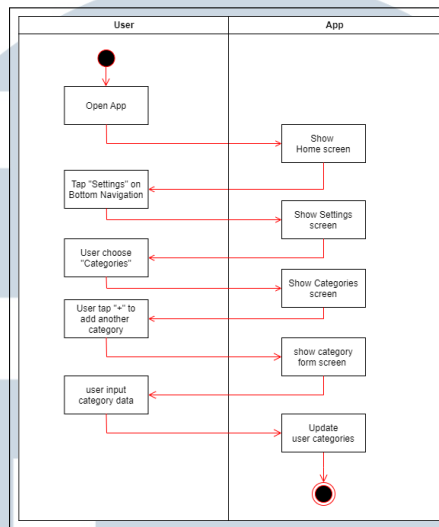
Gambar 3.6. Activity Diagram Edit Dompet

Selanjutnya, pada Gambar 3.7 ditunjukkan activity diagram yang menggambarkan proses pengguna menghapus dompet.



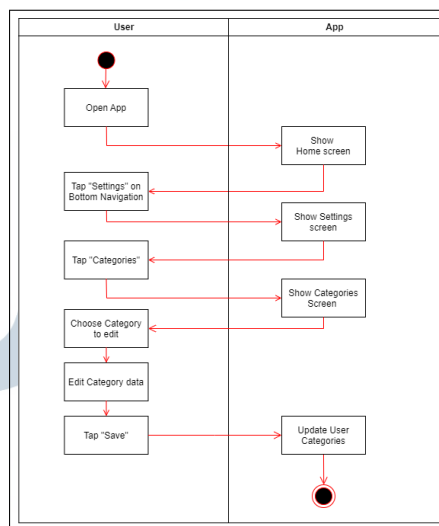
Gambar 3.7. Activity Diagram Delete Dompet

Selanjutnya, pada Gambar 3.8 ditunjukkan activity diagram yang menggambarkan proses pengguna menambahkan kategori baru ke dalam aplikasi.



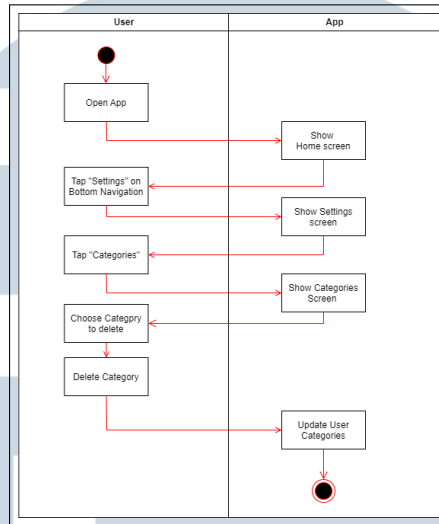
Gambar 3.8. Activity Diagram Add Kategori

Selanjutnya, pada Gambar 3.9 ditunjukkan activity diagram yang menggambarkan proses pengguna mengubah kategori.



Gambar 3.9. Activity Diagram Edit Kategori

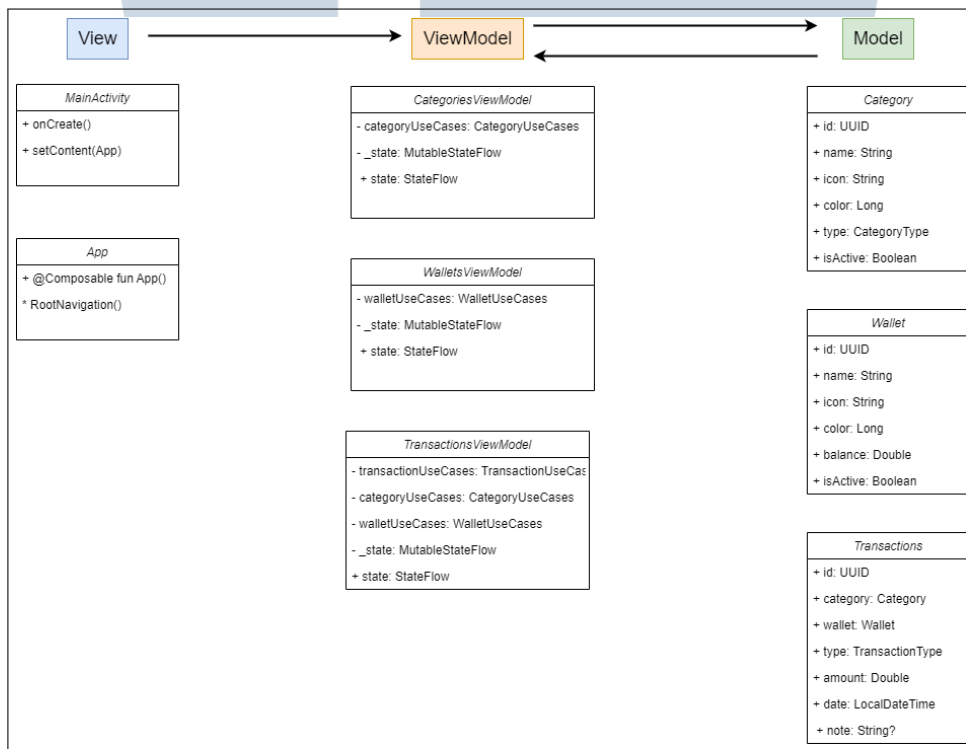
Selanjutnya, pada Gambar 3.10 ditunjukkan activity diagram yang menggambarkan proses pengguna menghapus kategori.



Gambar 3.10. Activity Diagram Delete Kategori

C Class Diagram

Pada Gambar 3.11 merupakan class diagram yang menggambarkan implementasi arsitektur *MVVM* secara garis besar. Diagram ini terdiri dari tiga komponen utama: *View*, *ViewModel*, dan *Model*. *Model* mewakili data dan logika bisnis, *View* menangani tampilan pengguna, sementara *ViewModel* bertindak sebagai perantara yang menghubungkan *Model* dan *View*, menyediakan data yang diperlukan dan mengelola interaksi antara keduanya. Dengan memisahkan peran-peran ini, *MVVM* memfasilitasi pengembangan yang terstruktur, memungkinkan pengujian yang lebih baik, serta memisahkan logika bisnis dari antarmuka pengguna.

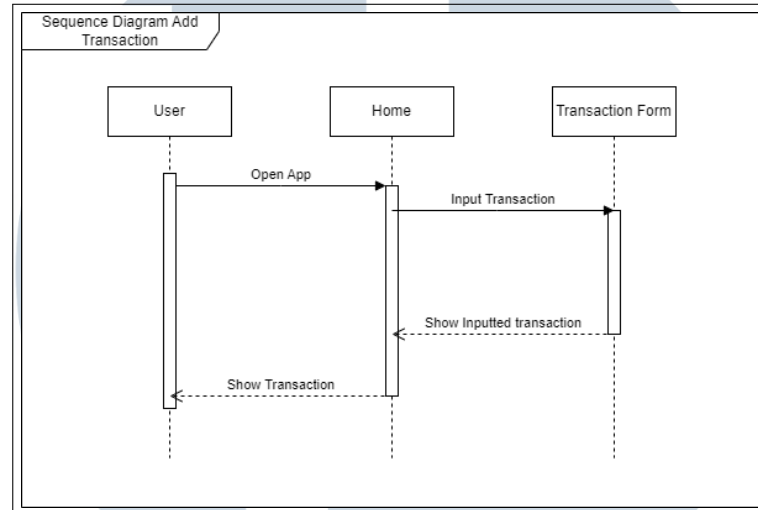


Gambar 3.11. *Class Diagram* MVVM

UNIVERSITAS
MULTIMEDIA
NUSANTARA

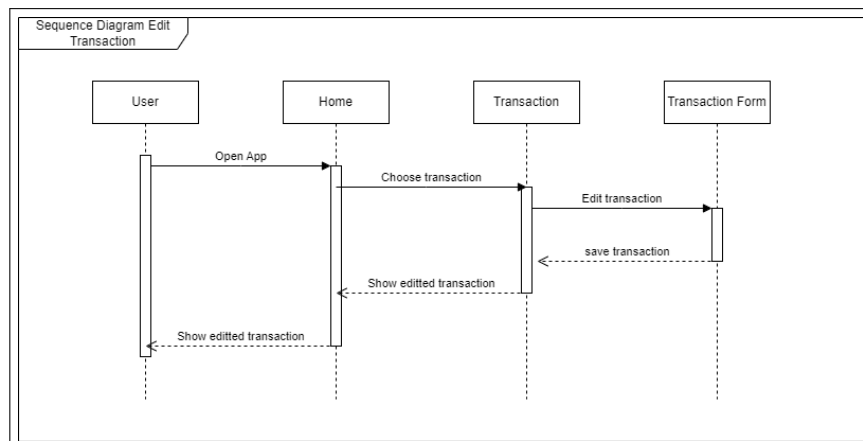
D Sequence Diagram

Pada Gambar 3.12 merupakan squence diagram yang menunjukkan interaksi antar objek pada aktivitas *add* transaksi.



Gambar 3.12. *Sequence Diagram Add* Transaksi

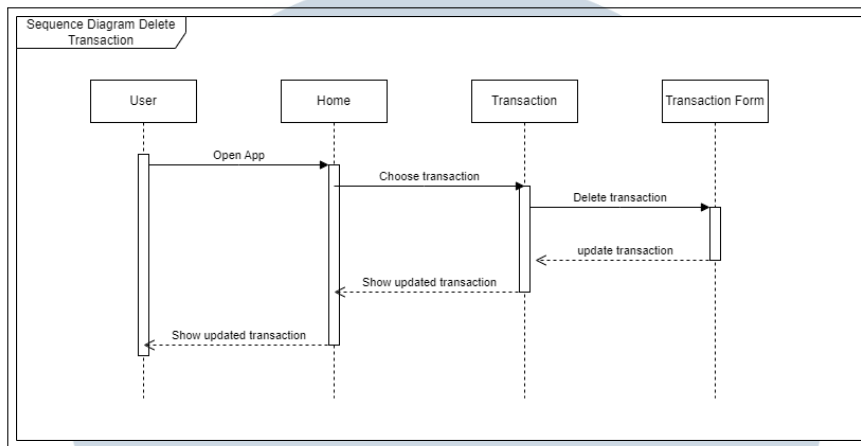
Pada Gambar 3.13 merupakan squence diagram yang menunjukkan interaksi antar objek pada aktivitas *edit* transaksi.



Gambar 3.13. *Sequence Diagram Edit* Transaksi

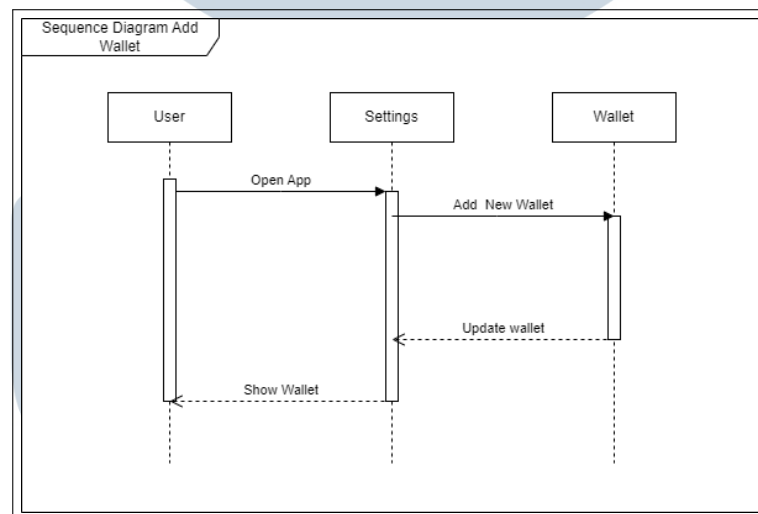
M U L T I M E D I A
N U S A N T A R A

Pada Gambar 3.14 merupakan sequence diagram yang menunjukkan interaksi antar objek pada aktivitas *delete* transaksi.



Gambar 3.14. *Sequence Diagram Delete* Transaksi

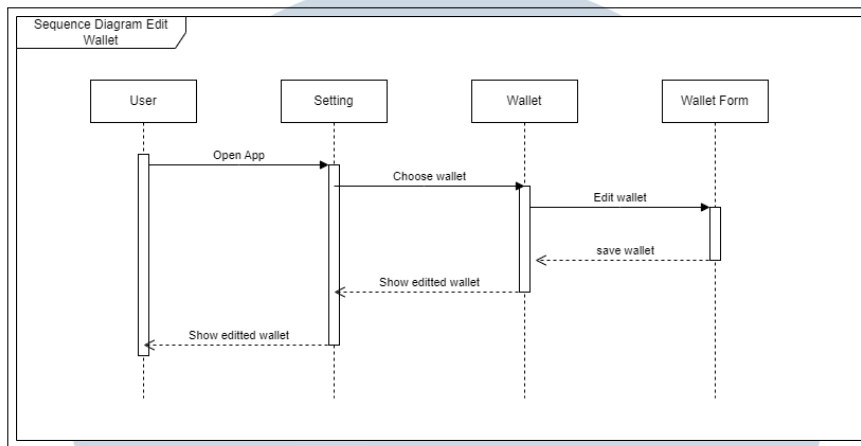
Pada Gambar 3.15 merupakan sequence diagram yang menunjukkan interaksi antar objek pada aktivitas *add* dompet.



Gambar 3.15. *Sequence Diagram Add* Dompet

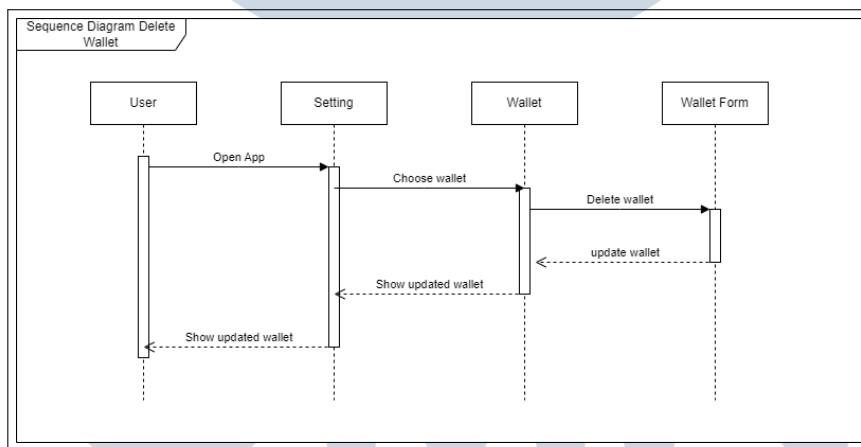
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada Gambar 3.16 merupakan sequence diagram yang menunjukkan interaksi antar objek pada aktivitas *edit* dompet.



Gambar 3.16. *Sequence Diagram Edit* Dompet

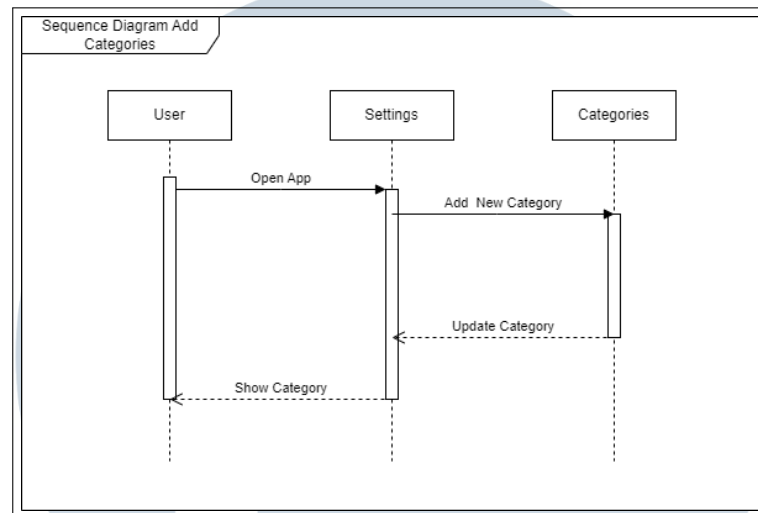
Pada Gambar 3.17 merupakan sequence diagram yang menunjukkan interaksi antar objek pada aktivitas *delete* dompet.



Gambar 3.17. *Sequence Diagram Delete* Dompet

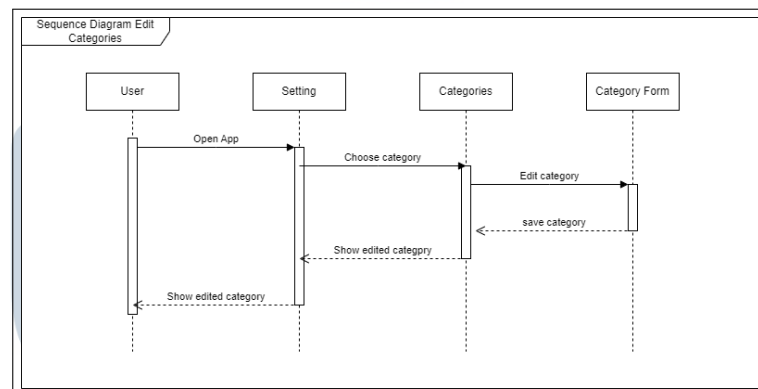
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada Gambar 3.18 merupakan sequence diagram yang menunjukkan interaksi antar objek pada aktivitas *add* kategori.



Gambar 3.18. *Sequence Diagram Add* Kategori

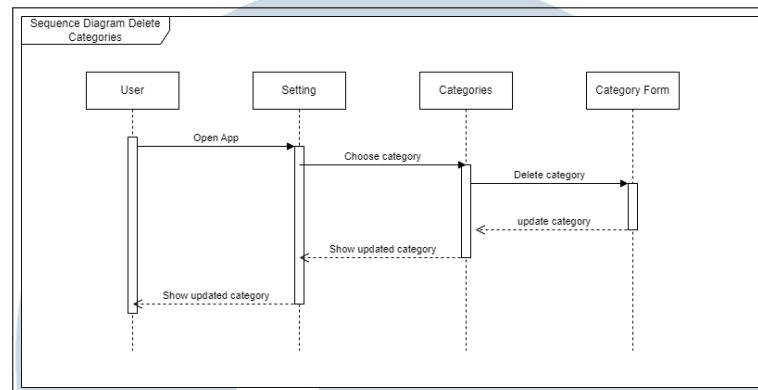
Pada Gambar 3.19 merupakan sequence diagram yang menunjukkan interaksi antar objek pada aktivitas *edit* kategori.



Gambar 3.19. *Sequence Diagram Edit* Kategori

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada Gambar 3.20 merupakan sequence diagram yang menunjukkan interaksi antar objek pada aktivitas *delete* kategori.



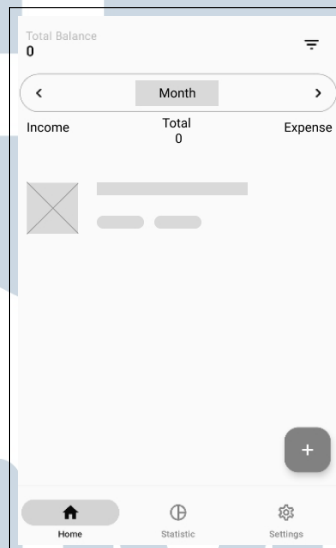
Gambar 3.20. *Sequence Diagram delete* Kategori

E Perancangan Antarmuka

Dalam proses perancangan suatu sistem, diperlukan perancangan antarmuka atau yang sering disebut sebagai mockup untuk memudahkan pembangunan aplikasi. Berikut ini adalah contoh mockup yang digunakan sebagai panduan dalam proses pembangunan aplikasi secara garis besar.

E.1 Mock-Up Home

Pada Gambar 3.21 merupakan tampilan pada halaman *Home*. Halaman ini merupakan halaman awal dari aplikasi yang akan dibangun.

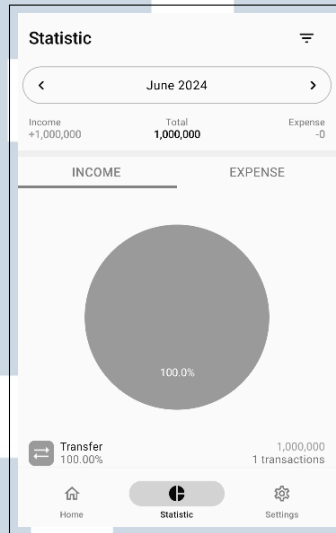


Gambar 3.21. *Mock-Up Home*

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

E.2 Mock-Up Statistic

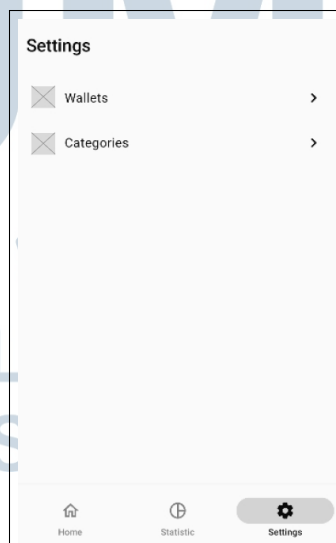
Pada Gambar 3.22 merupakan tampilan dari halaman *Statistic*. Pada halaman ini menampilkan *pie chart* dari setiap transaksi.



Gambar 3.22. *Mock-Up Upcoming*

E.3 Mock-Up Settings

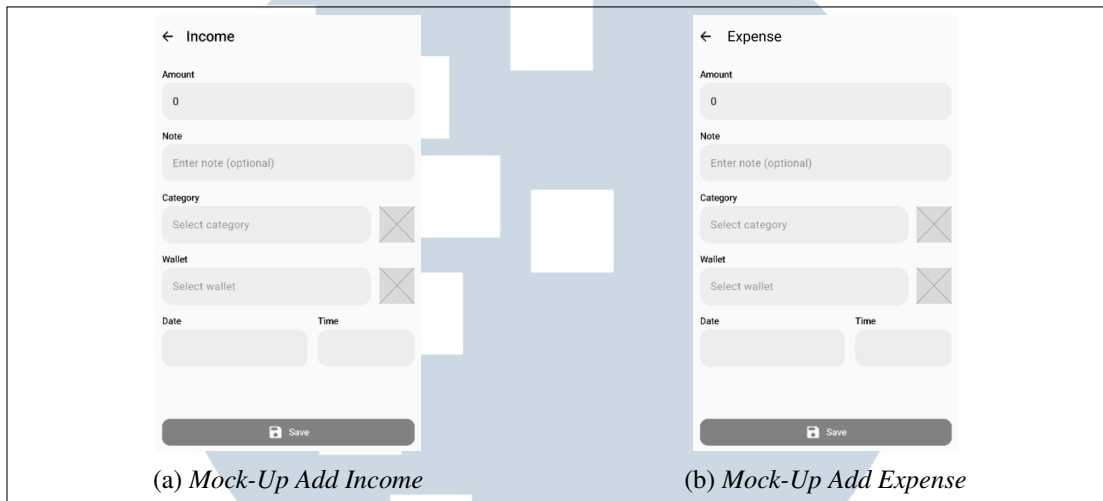
Pada Gambar 3.23 menampilkan halaman *Settings*. pada halaman ini pengguna dapat mengatur dompet dan juga kategori pemasukan dan pengeluaran.



Gambar 3.23. Halaman *Settings*

E.4 Mock-Up Add Transactions

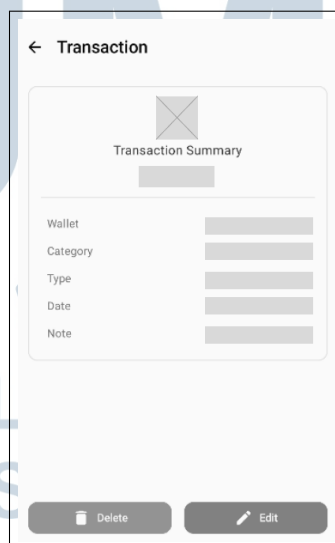
Pada Gambar 3.24 merupakan tampilan saat ingin menambah transaksi baik pengeluaran maupun pemasukan.



Gambar 3.24. *Mock-up add transaction*

E.5 Mock-Up Transaction Detail

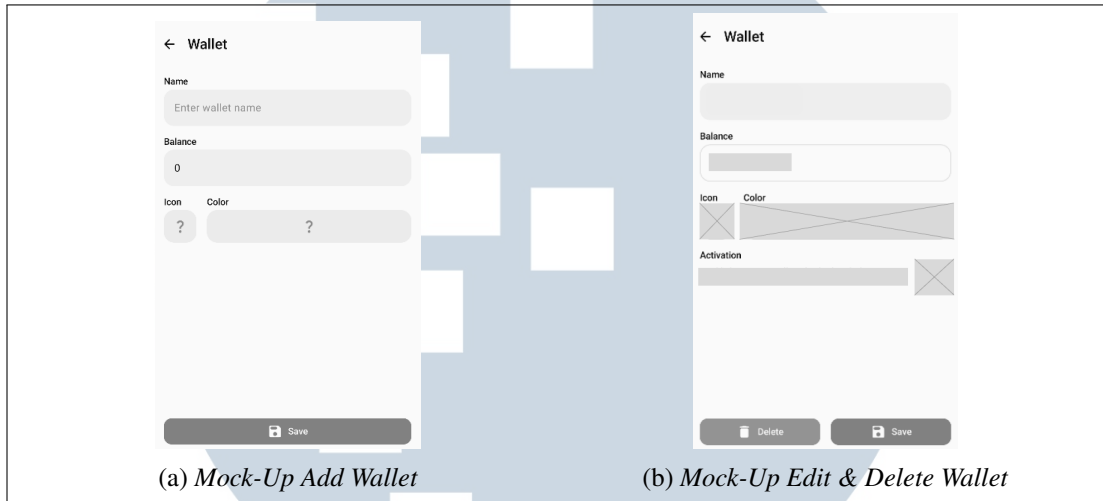
Pada Gambar 3.25 menampilkan halaman *Transaction detail*. pada halaman ini pengguna dapat mengubah atau menghapus transaksi.



Gambar 3.25. Halaman *Transaction detail*

E.6 Mock-Up Add, Edit & Delete Wallet

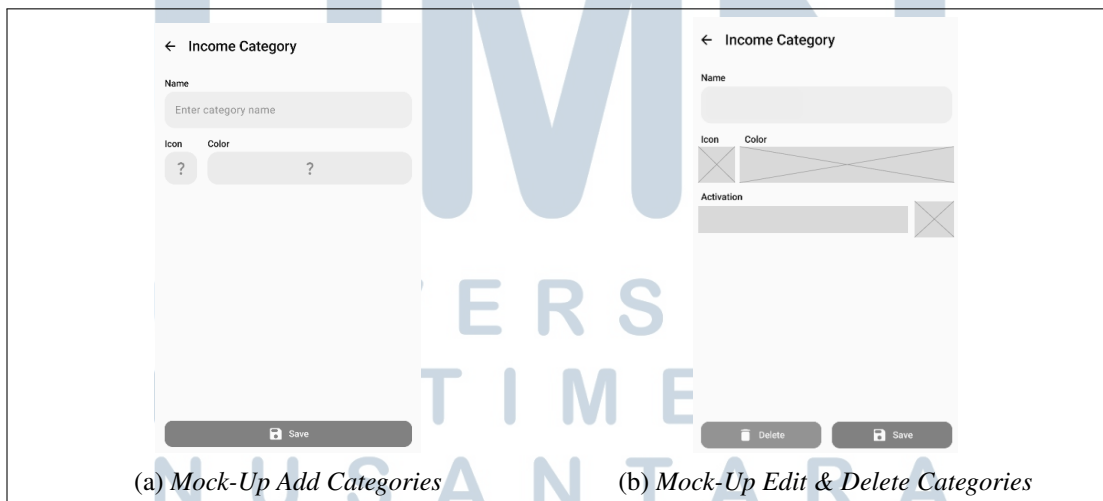
Pada Gambar 3.26 merupakan tampilan saat ingin menambah, mengubah atau menghapus dompet.



Gambar 3.26. *Mock-up Wallet*

E.7 Mock-Up Add, Edit & Delete Categories

Pada Gambar 3.27 merupakan tampilan saat ingin menambah, mengubah atau menghapus kategori.



Gambar 3.27. *Mock-up Wallet*

3.3.3 Pembangunan Aplikasi (Implementasi)

Pada tahap ini, proses pembangunan aplikasi dilakukan dengan mendasarkan pada desain antarmuka yang telah dirancang pada tahap sebelumnya. Aplikasi ini dikembangkan menggunakan bahasa pemrograman Kotlin, yang dipilih karena kemudahan dan keefektifannya dalam pengembangan aplikasi Android. Selama tahap perancangan dan pembangunan aplikasi ini, spesifikasi perangkat yang digunakan sangat penting untuk diperhatikan. Selain itu ada spesifikasi minimal untuk menginstall aplikasi ini. Berikut informasinya.

1. Perangkat Keras:

- Sistem Operasi: Microsoft Windows 11 Home Single Language
- *Processor*: 12th Gen Intel(R) Core(TM) i7-12650H 2.30 GHz
- Kartu Grafis: NVIDIA GeForce RTX 3050 Laptop GPU GDDR6 @ 4GB (128 bits)
- RAM: 16 GB

2. Perangkat Lunak:

- Android Studio
- Google Chrome
- Emulator Android
- Draw.io
- Figma

3. Minimum Requirement For App:

- Android 7.0 (Nougat)

3.3.4 Pengujian Penerimaan Pengguna

Dalam pengujian penerimaan pengguna aplikasi mobile manajemen keuangan personal berbasis Android yang dirancang, akan digunakan skala Likert yang terdiri dari beberapa pernyataan. Responden akan diminta untuk memberikan penilaian mereka terhadap masing-masing pernyataan dengan skala dari 1 (sangat tidak setuju) hingga 5 (sangat setuju). Berikut adalah pertanyaan yang diajukan kepada para responden.

- Saya merasa aplikasi ini mudah digunakan.
- Saya dapat dengan cepat mempelajari cara menggunakan semua fitur aplikasi ini.
- Antarmuka pengguna dalam aplikasi ini intuitif.
- Aplikasi ini berfungsi dengan baik tanpa koneksi internet.
- Saya dapat mengakses informasi pengeluaran saya secara offline kapan saja yang saya butuhkan.
- Aplikasi ini menyediakan semua fitur yang saya butuhkan untuk mengelola keuangan pribadi.
- Fitur pencatatan pengeluaran dalam aplikasi ini membantu saya mengontrol pengeluaran rutin.
- Saya puas dengan kinerja aplikasi ini.
- Aplikasi ini memenuhi kebutuhan manajemen keuangan saya sebagai mahasiswa.
- Saya akan merekomendasikan aplikasi ini kepada teman-teman saya.
- Aplikasi ini berjalan lancar di perangkat Android saya.
- Penggunaan warna untuk membedakan jenis pengeluaran membantu saya dalam mengelola keuangan dengan lebih efisien.

3.4 Penulisan Laporan

Pada fase ini, proses penulisan laporan mengenai penelitian yang telah dilakukan telah berlangsung. Selama proses penulisan, setiap langkah dalam perancangan dan pembangunan sistem telah didokumentasikan secara rinci. Dokumentasi tersebut kemudian disusun menjadi sebuah laporan penelitian yang komprehensif.