

## **TUGAS AKHIR**

# **RANCANG BANGUN APLIKASI MANAJEMEN KEUANGAN BERBASIS ANDROID PADA UNIT KEGIATAN MAHASISWA ILMU PENGETAHUAN DAN TEKNOLOGI DI POLITEKNIK NEGERI PONTIANAK**

Diajukan Sebagai Persyaratan Untuk Menyelesaikan

Program Pendidikan Diploma III Pada

Program Studi Teknik Informatika

Jurusan Teknik Elektro

Politeknik Negeri Pontianak



**OLEH:**

**IRFAN FAUZAN KURNIAWAN**

**3201716075**

**PROGRAM STUDI D-III TEKNIK INFORMATIKA  
JURUSAN TEKNIK ELEKTRO  
POLITEKNIK NEGERI PONTIANAK  
2020**

## HALAMAN PENGESAHAN

### RANCANG BANGUN APLIKASI MANAJEMEN KEUANGAN BERBASIS ANDROID PADA UNIT KEGIATAN MAHASISWA ILMU PENGETAHUAN DAN TEKNOLOGI DI POLITEKNIK NEGERI PONTIANAK

Oleh:  
**Irfan Fauzan Kurniawan**  
**3201716075**

Tugas akhir ini telah diterima dan disahkan sebagai salah satu syarat untuk  
menyelesaikan Program Pendidikan Diploma III pada Program Studi  
Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Pontianak.

Disahkan oleh :

Ketua Jurusan Teknik Elektro

**Hasan, ST, MT**  
**NIP. 197108201999031003**

Ketua Program Studi  
Teknik Informatika

**Suheri, S.T., M.Cs**  
**NIP. 198307172008121005**

Mengetahui,

Direktur Politeknik Negeri Pontianak



**Ir. H. Muh. Toasin Asha, M.Si**  
**NIP. 196112251990111001**

**HALAMAN PERNYATAAN**

**RANCANG BANGUN APLIKASI MANAJEMEN KEUANGAN BERBASIS  
ANDROID PADA UNIT KEGIATAN MAHASISWA ILMU  
PENGETAHUAN DAN TEKNOLOGI  
DI POLITEKNIK NEGERI PONTIANAK**

**Oleh:**

**Irfan Fauzan Kurniawan**

**3201716075**

**Pembimbing :**



**Neny Firdyanti, S.T., M.T.**

**NIP. 197710022014042001**

**Telah dipertahankan di depan penguji pada tanggal 25 Agustus 2020 dan  
dinyatakan memenuhi syarat sebagai Laporan Tugas Akhir.**

**Penguji**



**Tri Bowo Atmojo, ST., M.Cs.**

**NIP. 198407172019031010**

## **HALAMAN PERNYATAAN ORISINALITAS**

Saya yang bertanda tangan dibawah ini :

Nama : Irfan Fauzan Kurniawan  
NIM : 3201716075  
Jurusan / Program Studi : Teknik Elektro / Teknik Informatika  
Judul Tugas Akhir : Rancang Bangun Aplikasi Manajemen Keuangan Berbasis Android Pada Unit Kegiatan Mahasiswa Ilmu Pengetahuan Dan Teknologi Di Politeknik Negeri Pontianak.

Menyatakan dengan sebenar-benarnya bahwa penulisan Tugas Akhir ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah tugas akhir maupun kegiatan yang tercantum sebagai bagian dari Tugas Akhir ini. Jika terdapat karya orang lain, saya akan mencantumkan sumber yang jelas.

Demikian pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di Politeknik Negeri Pontianak.

Demikian pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Pontianak, 25 Agustus 2020

Yang membuat pernyataan,



Irfan Fauzan Kurniawan

NIM. 3201716075

## **RIWAYAT HIDUP**



### **Biodata Mahasiswa :**

Nama Mahasiswa	:	Irfan Fauzan Kurniawan
NIM	:	3201716075
Tempat / Tanggal Lahir	:	Bandung, 31 Maret 1999
Jenis Kelamin	:	Laki-laki
Agama	:	Islam
Alamat	:	Jalan Parit H Muksin 2, Megamas, blok C51
No. Telpon Rumah / Handphone	:	085821073911
Email	:	fauzanadi991@gmail.com

## **ABSTRAK**

Aplikasi manajemen keuangan berbasis android pada Unit Kegiatan Mahasiswa Ilmu Pengetahuan dan Teknologi (UKM IPTEK) bertujuan untuk merekam dan mengelola segala sesuatu yang berhubungan dengan keuangan dalam Lingkup UKM IPTEK termasuk kebutuhan biaya pada setiap kegiatan yang dilaksanakan, memonitoring setiap pengeluaran dan pemasukan serta menampilkan laporan dari setiap penggunaan dana. Aplikasi ini dibuat berdasarkan permasalahan yang terjadi pada pengelolaan data keuangan pada UKM IPTEK yang masih menggunakan media konvensional untuk mencatat data keuangan, dimana sangat mudah hilang dan tercecer.

Aplikasi ini dirancang berdasarkan informasi yang di gali dari bendahara UKM IPTEK dan jurnal – jurnal yang terkait dengan permasalahan yang ada. Aplikasi ini terdiri dari beberapa menu seperti menu pagudana, kegiatan, aset, kas, dan laporan dana. Aplikasi ini menggunakan sistem *multi user* dengan dua jenis *user* atau pengguna yaitu bendahara dan ketua dengan hak akses yang berbeda. Bendahara merupakan pengguna dengan hak akses penuh yang bisa mengelola data di aplikasi, sedangkan ketua adalah pengguna yang berfungsi mengawasi dan mengontrol apa saja yang dilakukan bendahara pada aplikasi.

Aplikasi ini dikembangkan menggunakan Android Studio dengan bahasa pemrograman Java dan database MySQL, dalam aplikasi ini terdapat beberapa fitur seperti daftar kegiatan beserta perkiraan pengeluarannya, mengatur estimasi penggunaan biaya pada setiap kegiatan, laporan keuangan berupa grafik, dan persentase pengeluaran dalam satu tahun.

Kata kunci: Android studio, MySQL, Keuangan, IPTEK.

## **ABSTRACT**

The android-based financial management application in the Science and Technology Student Activity Unit (UKM IPTEK) aims to record and manage everything related to finance in the UKM IPTEK Scope including the cost requirements for each activity carried out, monitoring every expense and income and displaying reports from any use of funds. This application is made based on the problems that occur in the management of financial data in UKM IPTEK that still use conventional media to record financial data, which is easily lost and scattered.

This application is designed based on information extracted from the treasurers of UKM IPTEK and journals related to existing problems. This application consists of several menus such as the pagudana menu, kegiatan, aset, kas, and laporan dana. This application uses a multi-user system with two types of users, namely bendahara and anggota with different access rights. The treasurer is a user with full access rights who can manage data in the application, while members are general users who can only view data.

This application was developed using Android Studio with the Java programming language and MySQL database, in this application there are several features such as a list of activities and their estimated expenses, managing the estimated cost usage for each activity, financial reports in the form of graphs, and the percentage of expenses in one year.

Keywords: Android studio, MySQL, Finance, IPTEK.

## **PRAKATA**

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan pimpinan-Nya yang telah penulis terima selama melaksanakan tugas akhir ini, sehingga pada akhirnya penulis dapat menyelesaikan tugas akhir ini dengan baik. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada orang-orang yang telah berperan sehingga dapat terselesaikannya tugas akhir ini, antara lain :

1. Orang tua yang selalu memberikan nasihat dan dukungan selama berjalannya proses penyusunan Tugas Akhir.
2. Bapak Ir.H.M. Toasin Asha, M.Si selaku Direktur Politeknik Negeri Pontianak.
3. Bapak Hasan, ST., MT selaku Ketua Jurusan Teknik Elektro
4. Bapak Suheri, S.T., M.Cs selaku Ketua Program Studi Teknik Informatika
5. Ibu Sarah Bibi, S.ST., M.Pd selaku Koordinator Tugas Akhir Program Studi Teknik Informatika.
6. Ibu Neny Firdyanti, S.T., M.T. selaku Dosen Pembimbing yang tiada henti memberikan bimbingan, dorongan, saran, kritik dan motivasi kepada penulis
7. Bapak Tri Bowo Atmojo, ST., M.Cs. selaku Dosen Pengaji 1 yang telah memberikan saran dan kritik selama proses penyusunan Tugas Akhir ini
8. Bapak Ramli, S.T., M.T. selaku Dosen Pengaji 2 yang juga telah memberikan saran dan kritik selama proses penyusunan Tugas Akhir ini
9. Seluruh staf pengajar dan administrasi Politeknik Negeri Pontianak, khususnya di Jurusan Teknik Elektro Program Studi Teknik Informatika

Penulis menyadari bahwa dalam penyusunan laporan Tugas Akhir ini masih jauh dari sempurna, untuk itu kritik dan saran dari pembaca yang membangun sangat penulis harapkan untuk menjadi pelajaran di kemudian hari. Penulis berharap semoga laporan Tugas Akhir ini dapat memberikan manfaat bagi kita semua. Terima kasih .

Pontianak, Agustus 2020

Penulis

## DAFTAR ISI

HALAMAN PENGESAHAN.....	i
HALAMAN PERNYATAAN .....	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
RIWAYAT HIDUP.....	iv
ABSTRAK .....	v
ABSTRACT .....	vi
PRAKATA .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
BAB I .....	1
PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan Penelitian .....	3
1.5. Manfaat Penelitian .....	3
1.5.1. Bagi Mahasiswa.....	3
1.5.2. Bagi Pengguna.....	3
1.6. Metodologi Penelitian.....	3
1.6.1. Metode Observasi .....	3
1.6.2. Metode Literatur .....	4
1.6.3. Metode Pengembangan.....	4
BAB II DASAR TEORI.....	6
2.1. Tinjauan Pustaka.....	6
2.2. Dasar Teori.....	7
2.2.1. Manajemen Keuangan .....	7
2.2.2. Android Studio.....	7
2.2.3. Java .....	8
2.2.4. MySQL .....	8
BAB III PERANCANGAN SISTEM .....	9
3.1. Gambaran Umum.....	9
3.2. Analisis Kebutuhan.....	9
3.2.1. Kebutuhan Pengguna .....	9

3.2.2.	Kebutuhan Fungsional .....	10
3.2.3.	Kebutuhan Non-fungsional.....	10
3.3.	Class Diagram.....	11
3.4.	Use Case Diagram.....	12
3.5.	<i>Sequence Diagram</i> .....	22
3.5.1.	<i>Sequence Diagram Login Pengguna.....</i>	22
3.5.2.	<i>Sequence Diagram Pagudana .....</i>	22
3.5.3.	<i>Sequence Diagram kegiatan .....</i>	24
3.5.4.	<i>Sequence Diagram Pengeluaran Kegiatan .....</i>	26
3.5.5.	<i>Sequence Diagram aset .....</i>	27
3.5.6.	<i>Sequence Diagram kas .....</i>	29
3.5.7.	<i>Sequence Diagram Laporan .....</i>	31
3.6.	Perancangan Database.....	31
3.6.1.	Relasi Tabel .....	31
3.6.2.	Struktur Tabel .....	32
3.7.	Perancangan Interface.....	34
3.7.1.	Halaman Utama .....	34
3.7.2.	Halaman Pagudana .....	35
3.7.3.	Halaman Kegiatan .....	36
3.7.4.	Halaman Pengeluaran Kegiatan.....	36
3.7.5.	Halaman Daftar Aset .....	37
3.7.6.	Halaman Daftar Kas .....	38
3.7.7.	Halaman laporan .....	38
BAB IV	<u>HASIL DAN PEMBAHASAN</u> .....	40
4.1.	Implementasi Perangkat Lunak.....	40
4.2.	Implemantasi Perangkat Keras .....	40
4.3.	Peta Situs Aplikasi .....	40
4.3.1.	Peta Situs Aplikasi Pada Tipe Pengguna Bendahara .....	40
4.3.2.	Peta Situs Aplikasi Pada Tipe Pengguna Anggota .....	41
4.4.	Implementasi Sistem Pada Tipe Pengguna Bendahara .....	42
4.3.1.	Implementasi Tampilan <i>Splash Screen</i> .....	42
4.3.2.	Implementasi Tampilan Tipe Pengguna .....	43
4.3.3.	Implementasi Fungsi Login Pengguna .....	44
4.3.4.	Implementasi Fungsi Register Bendahara .....	47
4.3.5.	Implementasi Tampilan Dashboard (Halaman Utama) .....	49

4.3.6.	Implementasi Menu Pagudana.....	50
4.3.7.	Implementasi Menu Daftar Kegiatan.....	53
4.3.8.	Implementasi Menu Daftar Pengeluaran Kegiatan .....	56
4.3.9.	Implementasi Menu Daftar Kas .....	61
4.3.10.	Implementasi Menu Daftar Aset.....	63
4.3.11.	Implementasi Menu Laporan dana .....	65
4.5.	Implementasi Sistem Pada Tipe Pengguna Anggota .....	67
4.6.	Analisis Seluruh Sistem .....	68
BAB V	<u>PENUTUP</u> .....	69
5.1.	Kesimpulan .....	69
5.2.	Saran .....	69
DAFTAR PUSTAKA .....		70
LAMPIRAN .....		71

## **DAFTAR TABEL**

Tabel 3.1. Use case scenario login bendahara.....	13
Tabel 3.2. Use case scenario mengelola pagudana .....	13
Tabel 3.3. Use case scenario mengelola data kegiatan .....	14
Tabel 3.4. Use case scenario mengelola data pengeluaran kegiatan .....	15
Tabel 3.5. Use case scenario mengelola data aset.....	17
Table 3.6. Use case scenario mengelola data kas.....	18
Tabel 3.7. Use case scenario melihat data pagudana .....	19
Tabel 3.8. Use case scenario melihat data kegiatan .....	19
Tabel 3.9. Use case scenario melihat data pengeluaran kegiatan .....	20
Tabel 3.10. Use case scenario melihat data aset .....	20
Tabel 3.11. Use case scenario melihat data kas .....	21
Tabel 3.12. Use case scenario melihat data laporan.....	21
Tabel 3.13. Tabel bendahara.....	32
Tabel 3.14. Tabel pagudana .....	33
Tabel 3.15 Tabel pagudana awal.....	33
Tabel 3.16. Tabel kegiatan.....	33
Tabel 3.17. Tabel pengeluaan kegiatan.....	33
Tabel 3.18. Tabel Aset .....	34
Tabel 3.19. Tabel kas .....	34

## DAFTAR GAMBAR

Gambar 1.1. Metode Waterfall.....	4
Gambar 3.1. Class diagram .....	11
Gambar 3.2. Use case diagram.....	12
Gambar 3.3. Sequence diagram login bendahara.....	22
Gambar 3.4. Sequence diagram menampilkan data pagudana.....	23
Gambar 3.5. Sequence diagram menambahkan dan mengedit data pagudana ..	23
Gambar 3.6. Sequence diagram menghapus data pagudana .....	24
Gambar 3.7. Sequence diagram menampilkan data kegiatan .....	24
Gambar 3.8. Sequence diagram menambah dan mengedit data kegiatan .....	25
Gambar 3.9. Sequence diagram menghapus data kegiatan .....	26
Gambar 3.10. Sequence diagram menampilkan data pengeluaran kegiatan .....	26
Gambar 3.11. Sequence diagram menambah dan mengubah data pengeluaran ...	27
Gambar 3.12. Sequence diagram menghapus data penggeluaran kegiatan .....	27
Gambar 3.13. Sequence diagram menampilkan data aset.....	28
Gambar 3.14. Sequence diagram menambah dan mengubah data aset.....	28
Gambar 3.15. Sequence diagram menghapus data aset .....	29
Gambar 3.16. Sequence diagram menampilkan data kas.....	29
Gambar 3.17. Sequence diagram menambah dan mengubah data kas .....	30
Gambar 3.18. Sequence diagram menghapus data kas .....	30
Gambar 3.19. Sequence diagram menampilkan data laporan dana .....	31
Gambar 3.20. Relasi Tabel.....	32
Gambar 3.21. Rancangan antarmuka halaman utama .....	35
Gambar 3.22. Rancangan antarmuka halaman pagudana .....	35
Gambar 3.23. Rancangan antarmuka halaman daftar kegiatan.....	36
Gambar 3.24. Rancangan antarmuka halaman daftar pengeluaran kegiatan .....	37
Gambar 3.25. Rancangan antarmuka halaman daftar aset .....	37
Gambar 3.26. Rancangan antarmuka halaman daftar kas .....	38
Gambar 3.27. Rancangan antarmuka halaman laporan dana .....	39
Gambar 4.1. Peta Situs untuk tipe pengguna Bendahara .....	41
Gambar 4.2. Peta situs untuk tipe pengguna Anggota .....	41
Gambar 4.3. Tampilan halaman splash screen.....	42
Gambar 4.4. Tampilan tipe pengguna .....	43
Gambar 4.5. Kode program pilihan tipe pengguna .....	44

Gambar 4.6. Tampilan login unruk bendahara.....	45
Gambar 4.7. Kode program untuk verifikasi akun bendahara .....	46
Gambar 4.8. Tampilan pesan jika salah memasukan username atau password ....	47
Gambar 4.9. Kode program untuk menampilkan pesan error.....	47
Gambar 4.10. Tampilan halaman register .....	48
Gambar 4.11. Kode program untuk mengambil data hasil input .....	48
Gambar 4.12. Kode program untuk menyimpan data akun bendahara.....	49
Gambar 4.13. Tampilan dashboard bendahara.....	50
Gambar 4.14. Tampilan halaman pagudana.....	51
Gambar 4.15. Kode program untuk menampilkan data pagudana.....	52
Gambar 4.16. Tampilan halaman form pagudana .....	52
Gambar 4.17. Tampilan daftar kegiatan.....	53
Gambar 4.18. Kode program untuk menampilkan data kegiatan di frontend .....	54
Gambar 4.19. Kode program untuk menampilkan data kegiatan di backend .....	54
Gambar 4.20. Tampilan halaman form kegiatan dan daftar pengeluaran .....	55
Gambar 4.21. Kode untuk melakukan request meyimpan data kegiatan.....	56
Gambar 4.22. Tampilan halaman daftar pengeluaran kegiatan.....	57
Gambar 4.23. Kode untuk menampilkan data pengeluaran pada frontend .....	57
Gambar 4.24. Kode untuk menampilkan data pengeluaran pada backend .....	58
Gambar 4.25. Tampilan form pengeluaran kegiatan dalam edit mode .....	59
Gambar 4.26. Tampilan form pengeluaran kegiatan dalam read mode .....	59
Gambar 4.27. Kode program untuk mengatur fungsi read mode dan edit mode ..	60
Gambar 4.28. Tampilan pesan konfirmasi untuk menghapus.....	61
Gambar 4.29. Tampilan halaman daftar kas .....	62
Gambar 4.30. Tampilan halaman form kas .....	63
Gambar 4.31. Tampilan halaman daftar aset .....	64
Gambar 4.32. Tampilan halaman form aset .....	65
Gambar 4.33. Tampilan halaman laporan dana .....	66
Gambar 4.34. Tampilan halaman dashboard ketua.....	67
Gambar 4.35. Tampilan halaman detail dari tipe pengguna ketua.....	68

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

“Manajemen keuangan adalah suatu aktivitas yang berkaitan dengan bagaimana memperoleh, menggunakan, dan mengelola keuangan dalam sebuah perusahaan, organisasi, rumah tangga, maupun pribadi” [1]. Pada sebuah organisasi dan Unit Kegiatan Mahasiswa (UKM), manajemen keuangan adalah hal yang sangat penting untuk diterapkan, karena keuangan merupakan salah satu pondasi yang kuat dalam pertumbuhan dan perkembangan sebuah organisasi. Jika tidak pandai dalam mengatur keuangan bisa berpengaruh pada pelaksanaan kegiatan yang tidak sesuai dengan perencanaannya, dan pengeluaran yang berlebihan.

Unit Kegiatan Mahasiswa Ilmu Pengetahuan dan Teknologi Politeknik Negeri Pontianak (UKM IPTEK POLNEP) merupakan salah satu UKM dibawah naungan kementerian UKM BEM POLNEP yang bergerak di bidang ilmu pengetahuan dan teknologi. Untuk struktur organisasi dari UKM IPTEK POLNEP adalah mulai dari yang tertinggi ada Ketua, Wakil Ketua, Sekretaris 1, Sekretaris 2, Bendahara 1, Bendahara 2, Divisi Internal, Divisi Jurnalistik, Divisi Karya Tulis Ilmiah, dan Divisi Kominfo.

Permasalahan yang dialami oleh UKM IPTEK POLNEP dalam mengelola keuangan diantara lain, dalam pembuatan Laporan Penganggung Jawaban, bendahara mengalami kesulitan dalam mendata pengeluaran dan pemasukan karena beberapa faktor seperti hilangnya berkas laporan keuangan, perhitungan yang tidak sesuai, dan berbagai masalah yang disebabkan oleh cara penyimpanan data keuangan yang masih menggunakan media kertas sebagai tempat menyimpan data-data penting. Metode seperti ini memungkinkan kertas tersebut terselip atau bahkan hilang.

Selain dari permasalahan di atas penulis juga telah menelusuri beberapa jurnal penelitian yang terkait dengan topik yang akan dibahas diantaranya, dari penelitian yang pertama berjudul “Aplikasi Manajemen Keuangan Pribadi Berbasis Android” yang diajukan sebagai tugas akhir oleh Nila Khoiri

dan Aris J Santoso. Pada aplikasi yang menjadi referensi tersebut ada kekurangan yang dijadikan bahan acuan yaitu pada bagian menampilkan laporan hanya menggunakan export file excel saja, dan untuk referensi yang kedua yaitu sebuah artikel yang berjudul “Perancangan aplikasi manajemen keuangan pribadi berbasis android” yang diajukan oleh Meyta Nastiti dan Andi Sunyoto, M.Kom. memiliki kelemahan yaitu tidak memiliki fitur login..... [2]

Dari berbagai permasalahan di atas penulis bermaksud membuat proposal Tugas Akhir yang berjudul “Rancang Bangun Aplikasi Manajemen Keuangan Berbasis Android pada Unit kegiatan mahasiswa Ilmu pengetahuan dan Teknologi di Politeknik Negeri Pontianak” yang diharapkan bisa mempermudah mahasiswa untuk mengatur dan mengelola keuangan di organisasi..... [3]

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah dituliskan di atas maka dapat dirumuskan masalah sebagai berikut:

- 1) Bagaimana membangun aplikasi manajemen keuangan pada Unit kegiatan mahasiswa Ilmu pengetahuan dan Teknologi (UKM IPTEK) di Politeknik Negeri Pontianak berbasis android.
- 2) Bagaimana membuat rekap data dan laporan keuangan menjadi lebih mudah tanpa harus menghitung secara manual.

## **1.3. Batasan Masalah**

Batasan-batasan masalah dari pembuatan proposal tugas akhir ini adalah sebagai berikut :

- 1) Aplikasi ini dibangun hanya berdasarkan pada masalah di ruang lingkup UKM IPTEK di Politeknik Negeri Pontianak.
- 2) Bahasa pemrograman yang digunakan adalah java dengan menggunakan IDE Android Studio versi 3.5.0.
- 3) *Database* yang digunakan dalam membangun sistem ini adalah menggunakan MySQL.

## **1.4. Tujuan Penelitian**

Tujuan dari pembuatan tugas akhir ini adalah untuk:

- 1) Membangun aplikasi manajemen keuangan pada organisasi mahasiswa berbasis android di Politeknik Negeri Pontianak berbasis android.
- 2) Membuat aplikasi yang bisa membantu aktivitas manajemen keuangan di organisasi mahasiswa.
- 3) Membangun aplikasi manajemen keuangan dengan fitur yang bisa menampilkan laporan dalam bentuk grafik dan ekspor file dokumen excel.
- 4) Membuat aplikasi yang bisa mengelola pemasukan dan pengeluaran dalam manajemen keuangan organisasi mahasiswa menjadi lebih terstruktur dan terorganisir lebih baik.

## **1.5. Manfaat Penelitian**

Adapun manfaat dari penelitian yang dilakukan adalah sebagai berikut:

### **1.5.1. Bagi Mahasiswa**

Melatih mahasiswa dalam menerapkan pengetahuan dan keterampilan dalam yang diperoleh dalam bangku kuliah khususnya dalam ilmu pemrograman berorientasi objek.

### **1.5.2. Bagi Pengguna**

Membantu mempermudah aktivitas manajemen keuangan dalam lingkup UKM IPTEK, dan juga memberi kemudahan dalam membuat catatan dan laporan keuangan.

## **1.6. Metodologi Penelitian**

Teknik Pengumpulan data yang dilakukan dalam menyusun proposal laporan tugas akhir ini adalah sebagai berikut:

### **1.6.1. Metode Observasi**

Pengertian observasi secara umum adalah sebuah pengamatan atau aktivitas yang dilakukan untuk mengetahui sesuatu dari sebuah

fenomena yang didasari pada pengetahuan dan gagasan yang bertujuan untuk memperoleh informasi-informasi terkait dengan suatu fenomena atau peristiwa yang sudah atau sedang terjadi dilingkungan..... [2]

Observasi yang dilakukan penulis adalah mengamati segala kegiatan dan aktivitas pengelolaan keuangan yang masih menggunakan metode konvesional sebagai media penyimpana seperti kertas dan buku, dimana benda-benda tersebut mudah hilang dan tercecer. Serta saat merekap data keuangan yang mengharuskan menelusuri data-data sebelumnya dimana data-data tersebut belum tentu lengkap semua.

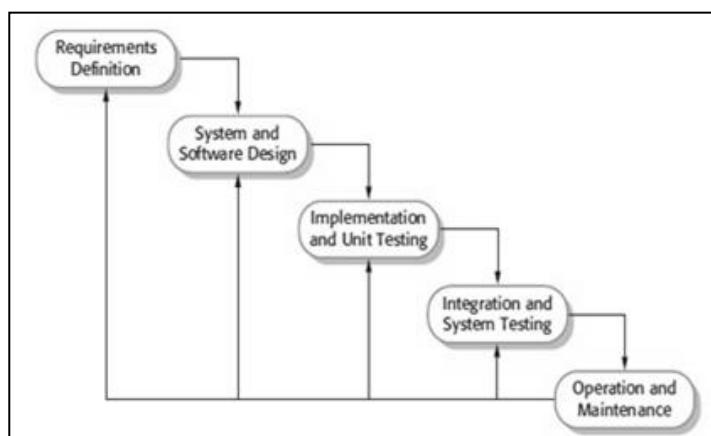
### **1.6.2. Metode Literatur**

Metode literatur adalah metode pengumpulan data dengan menelusuri jurnal-jurnal, modul-modul, e-book (Buku elektronik), buku referensi, maupun situs-situs penyedia informasi.... [2]

Literatur yang diambil oleh penulis berasal dari jurnal, buku, e-book, serta situs-situs di internet dan juga laporan tugas akhir dari senior terdahulu.

### **1.6.3. Metode Pengembangan**

Metode Waterfall model adalah tahapan utama dari pengembangan langsung yang mencerminkan aktifitas pengembangan dasar. Ada 5 langkah pada model pengembangan *waterfall*, yaitu *requirement analysis and definition, system and software design, implementation and unit testing, integration and system testing, dan operation and maintenance*.



Gambar 1.1. Metode Waterfall

Berikut adalah penjelasan dari tahapan-tahapan tersebut :

- *Requirement Analysis and Definition*

Merupakan tahapan penetapan tujuan, kendala, dan fitur sistem dengan melakukan konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai kebutuhan sistem.

Analisis data yang sudah dilakukan oleh penulis adalah observasi mengenai kebutuhan data, dan menerima informasi mengenai permasalahan dari pengguna sistem.

- *System and Software Design*

Dalam tahapan ini adalah pembuatan arsitektur sistem berdasarkan persyaratan dan kebutuhan yang telah ditetapkan. Dan juga mengidentifikasi dan menggambarkan abstraksi dasar – dasar hubungan dalam sistem perangkat lunak

- *Implementation and Unit Testing*

Dalam tahapan ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program. Setiap unit akan diuji apakah sudah memenuhi kebutuhan spesifikasinya.

- *Integration and System Testing*

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem..... [3]

## **BAB II**

### **DASAR TEORI**

#### **2.1. Tinjauan Pustaka**

Dari beberapa referensi yang penulis dapatkan ada sejumlah perbedaan yang dapat dijadikan perbandingan. Dari penelitian pertama yang berjudul “Aplikasi Manajemen Keuangan Pribadi Berbasis Android” yang diajukan sebagai tugas akhir oleh Nila Khoiri dan Aris J Santoso. Pada aplikasi yang menjadi referensi tersebut ada beberapa kekurangan yaitu pada bagian menampilkan laporan hanya menggunakan export file excel saja,... [4]

Referensi yang kedua adalah artikel yang berjudul “Perancangan aplikasi manajemen keuangan pribadi berbasis android” yang diajukan oleh Meyta Nastiti dan Andi Sunyoto, M.Kom.. Pada aplikasi yang menjadi referensi yang kedua ini, jika dibandingkan dengan sistem yang akan dibuat penulis, aplikasi yang dibuat menggunakan *database* lokal, sehingga tidak cocok dengan kebutuhan pengguna pada aplikasi yang dibuat penulis, karena memiliki dua jenis pengguna yang mengakses satu *database* yang sama.... [5]

Berdasarkan kedua penelitian diatas, penulis berencana untuk membuat sistem yang bisa menutupi kelemahan yang ada di kedua penelitian tersebut. Aplikasi yang dikembangkan pada kedua penelitian tersebut ditujukan untuk pengguna umum, sedangkan aplikasi yang akan penulis kembangkan ditujukan khusus pada ruang lingkup Unit Kegiatan Mahasiswa Ilmu Pengetahuan dan Teknologi di Politeknik Negeri Pontianak, dimana terdapat banyak perbedaan yang signifikan pada bagian komponen menu pilihan pada pengeluaran dan pemasukan. Aplikasi yang akan penulis kembangkan juga dilengkapi fitur laporan yang menggunakan format tabel yang berupa nama kegiatan dan estimasi biaya yang digunakan serta jumlah keseluruhan dana yang digunakan selama satu kepemimpinan dan format grafiknya yang bisa mempermudah pengguna dalam membuat laporan keuangan.

## 2.2. Dasar Teori

### 2.2.1. Manajemen Keuangan

Menurut Riyanto. B (1995), keseluruhan aktivitas yang berhubungan dengan usaha untuk mendapatkan dana dan juga mengalokasikan dana tersebut disebut pembelanjaan perusahaan dalam arti luas (business finance) atau manajemen keuangan (financial management). Dapat dikatakan juga bahwa manajemen keuangan merupakan salah satu bentuk penerapan ilmu ekonomi, yang mempunyai beberapa fungsi, dimana fungsi tersebut akan diaplikasikan dalam aktivitas atau operasional perusahaan setelah melalui proses pengambilan keputusan-keputusan yang berhubungan dengan sumber pendanaan dan pengalokasian dana tersebut, serta pengelolaan dan perimbangan seperti pada kas, piutang, persediaan , efek (disebut juga pengelolaan pada aktiva lancar), investasi, aktiva tetap, utang jangka pendek, utang jangka panjang modal sendiri, saham, dan obligasi yang semua itu perlu perencanaan yang matang, pengelolaan, pengkajian, pengawasan, penilaian, dan evaluasi dan setiap komponen tersebut agar terjadi suatu sinergi yang akurat sehingga dapat menggiring kepada pencapaian tujuan perusahaan yaitu profit yang maksimal, biaya dan resiko yang minimal.... [1]

### 2.2.2. Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *software* sebelumnya, IntelliJ IDEA. Selain menggunakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

- 1) Sistem versi berbasis *Gradle* yang fleksibel.
- 2) Emulator yang cepat dan memiliki banyak fitur.
- 3) Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.

- 4) *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
- 5) Template kode dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
- 6) Alat pengujian dan kerangka kerja yang ekstensif.
- 7) Alat *Lint* untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
- 8) Dukungan C++ dan NDK Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*.... [6]

#### **2.2.3. Java**

Java merupakan bahasa pemrograman tingkat tinggi yang dipelopori oleh James Gosling yang merupakan engineer di Sun Microsystem. Java mulai dibangun pada tahun 1991. Versi alpha dan beta dari Java dirilis pada tahun 1995, empat tahun setelah projek Java diinisiasi. Pada tahun 2010, Sun Microsystem diakuisisi oleh Oracle dan menjadikan Java dikembangkan di bawah kuasa Oracle.... [7]

#### **2.2.4. MySQL**

MySQL adalah sistem RDBMS (Relational database management system) yang saat ini paling populer dan banyak digunakan sumber dunia teknologi database terbuka dan sistem penyimpanan data. MySQL menawarkan kehandalan besar dan kemudahan penggunaan.

MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Dua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.... [8]

## **BAB III**

### **PERANCANGAN SISTEM**

#### **3.1. Gambaran Umum**

Aplikasi manajemen keuangan UKM IPTEK adalah aplikasi berbasis android yang dibuat untuk mempermudah bendahara mengelola keuangan dan membantu dalam membuat laporan penanggung jawaban pagudana. Aplikasi ini berfungsi untuk mencatat segala pemasukan dan pengeluaran dari semua jenis transaksi di dalam lingkup UKM IPTEK. Aplikasi ini menggunakan *recyclerview* pada *android studio* untuk menampilkan data berupa daftar yang terdapat fitur *search* yang bisa mempermudah untuk mencari data yang diinginkan hanya dengan menulis nama atau judul konten data tersebut. Dan fitur untuk mengupload bukti nota pembayaran, Serta dengan fitur laporan yang disertai grafik untuk mempermudah bendahara mengidentifikasi keadaan terkini keuangan, seperti melihat sisa pagudana, jumlah kas yang terkumpul jumlah pengeluaran dan sebagainya.

#### **3.2. Analisis Kebutuhan**

Analisis kebutuhan bertujuan untuk mencari dan menemukan segala kebutuhan yang akan digunakan oleh pengguna pada aplikasi. Kemudian kebutuhan pengguna akan dipenuhi oleh layanan perangkat lunak yang dituangkan dalam spesifikasi kebutuhan perangkat lunak. Kebutuhan perangkat lunak terdiri atas kebutuhan fungsional dan kebutuhan non fungsional.

##### **3.2.1. Kebutuhan Pengguna**

Pada aplikasi ini terdapat dua jenis pengguna yaitu bendahara dan pengguna selain bendahara yang dimana dua jenis pengguna ini memiliki hak akses yang berbeda berdasarkan Kebutuhan pada setiap pengguna.

### **3.2.1.1. Bendahara**

Bendahara merupakan pengguna yang memiliki hak akses penuh pada aplikasi untuk mengelola data berupa mencatat, mengubah dan menghapus keseluruhan data yang ada di aplikasi ini.

### **3.2.1.2. Ketua**

Jenis pengguna ini adalah pengguna yang memiliki hak akses terbatas, hanya bisa melihat informasi yang ada di aplikasi tanpa bisa untuk menambahkan, mengubah maupun menghapus data didalamnya.

### **3.2.2. Kebutuhan Fungsional**

Kebutuhan pengguna dilengkapi dengan adanya kebutuhan fungsional, yakni layanan yang tersedia pada sistem. Adapun kebutuhan fungsional yang tersedia pada aplikasi adalah sebagai berikut :

- a. Fungsi *login* akun
- b. Fungsi *logout* akun
- c. Fungsi pengelolaan data pagudana
- d. Fungsi pengelolaan data kegiatan
- e. Fungsi pengelolaan data pengeluaran kegiatan
- f. Fungsi pengelolaan data kas
- g. Fungsi pengelolaan data aset

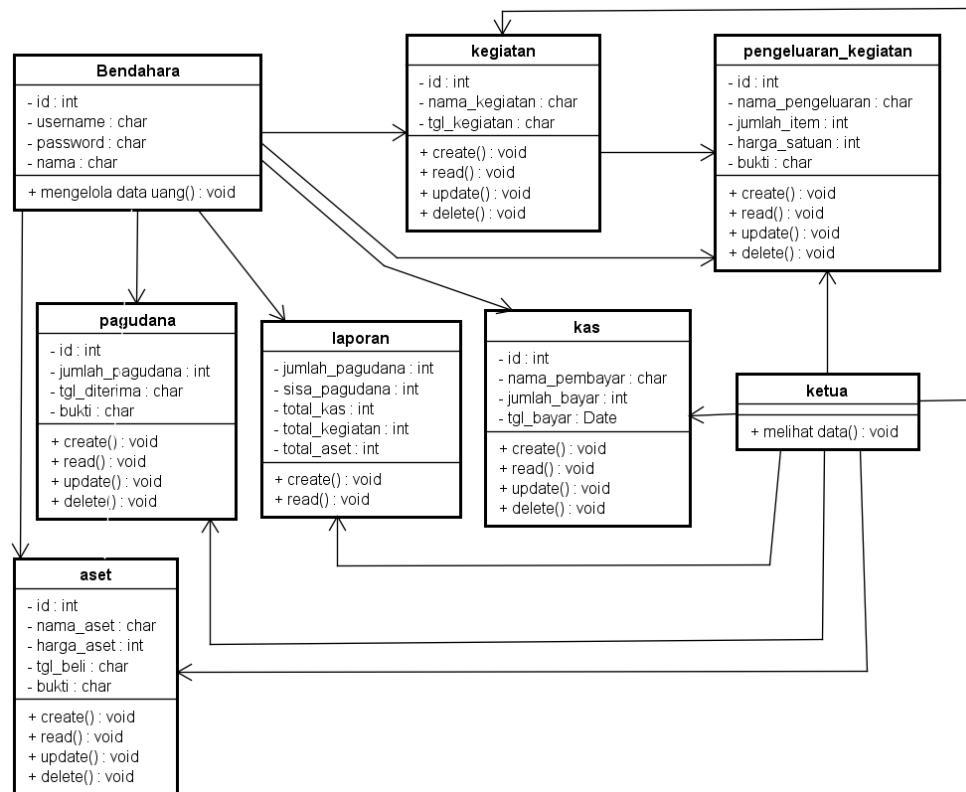
### **3.2.3. Kebutuhan Non-fungsional**

Adapun kebutuhan non fungsional yang tersedia untuk memperkuat kebutuhan fungsional pada aplikasi adalah sebagai berikut:

- a. Aplikasi ini terdapat fitur untuk bendahara mengupload bukti pembayaran atau nota.
- b. Laporan dengan menampilkan grafik penggunaan pagudana.

### 3.3. Class Diagram

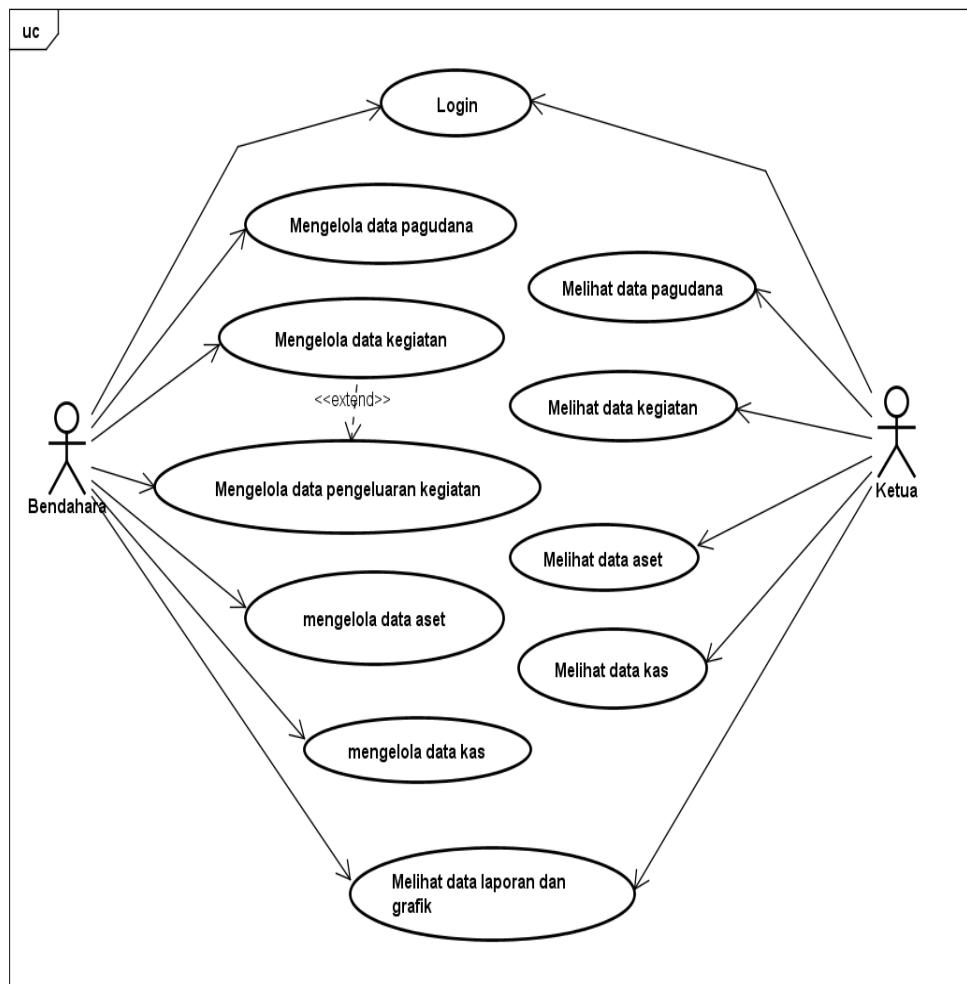
*Class diagram* merupakan diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* yang digunakan untuk membuat aplikasi ini, terdiri dari *class* bendahara, *class* anggota, *class* pagudana, *class* kegiatan, *class* Pengeluaran kegiatan, *class* aset, *class* kas dan *class* laporan. *Class* bendahara bisa mengakses dan menggunakan *class* sebagai pengelola data di *class* pagudana, kegiatan, pengeluaran kegiatan, aset, kas, dan laporan. *Class* kegiatan mengekstend *class* pengeluaran kegiatan, dan *class* anggota bisa mengakses dengan hanya bisa melihat data di *class* pagudana, kegiatan, pengeluaran kegiatan, aset, kas, dan laporan. Adapun rancangan *class diagram*nya adalah sebagai berikut:



Gambar 3.1. *Class diagram*

### 3.4. Use Case Diagram

Use case diagram merupakan gambaran skenario dari interaksi antara pengguna dengan sistem. Use case diagram menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi. Adapun *use case diagram* dari rancangan aplikasi yang telah dibuat adalah sebagai berikut:



Gambar 3.2. *Use case diagram*

Berdasarkan *use case diagram* tersebut dapat dibuat skenarionya sebagai berikut.

Untuk *login* Bendahara, berikut adalah tabel skenarionya.

**Tabel 3.1. Use case scenario login pengguna**

No	01
Nama Use Case	Login
Dekripsi	<i>user</i> melakukan <i>login</i> ke aplikasi
Aktor	Bendahara dan Ketua
Kondisi awal	Aplikasi telah terbuka dan masuk ke halaman login
Kondisi normal	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman login</li> <li>2. Pengguna mengisi data login berupa <i>username</i> dan <i>password</i></li> <li>3. Aplikasi memproses data login</li> <li>4. Aplikasi menampilkan menu utama</li> </ol>
Kondisi alternatif	<ol style="list-style-type: none"> <li>1. Jika <i>field</i> kosong maka akan menampilkan pesan “<i>field</i> ini tidak boleh kosong”</li> <li>2. Jika nama dan password salah, maka akan menampilkan pesan “<i>nama</i> dan <i>password</i> salah” dan kembali ke halaman login</li> </ol>
Kondisi akhir	Aplikasi menampilkan halaman menu utama

Adapun tabel skenario *use case* mengelola pagudana adalah sebagai berikut.

**Tabel 3.2. Use case scenario mengelola pagudana**

No	02
Nama Use Case	Mengelola data pagudana
Dekripsi	User mengelola data pagudana
Aktor	Bendahara
Kondisi awal	Aplikasi telah membuka halaman pagudana
Kondisi normal	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman pagudana</li> <li>2. Jika data kosong maka bendahara melakukan penambahan data. jika tidak, maka terdapat dua pilihan yaitu mengedit atau menghapus data.</li> </ol>

	<ol style="list-style-type: none"> <li>3. Jika bendahara memilih menambahkan data dan mengedit data, maka sistem akan menampilkan form data pagudana</li> <li>4. Masuk ke dalam form data pagudana</li> <li>5. Bendahara mengisi data pagudana berupa jumlah pagudana, tanggal diterima, dan bukti diterima</li> <li>6. Aplikasi memproses untuk menyimpan data</li> <li>7. Aplikasi menampilkan halaman pagudana</li> <li>8. Jika bendahara memilih untuk menghapus, maka akan muncul dialog konfirmasi untuk menghapus.</li> <li>9. Jika bendahara memilih ya, maka data pagudana akan terhapus</li> <li>10. Jika tidak maka sistem akan kembali ke halaman data pagudana</li> </ol>
Kondisi alternatif	Jika <i>field</i> kosong maka akan menampilkan pesan “field ini tidak boleh kosong”
Kondisi akhir	Data pagudana yang telah dikelola akan tersimpan dan selalu terupdate di <i>database</i>

Adapun tabel skenario *use case* mengelola data kegiatan adalah sebagai berikut.

**Tabel 3.3. Use case scenario mengelola data kegiatan**

No	03
Nama Use Case	Mengelola data kegiatan
Dekripsi	<i>user</i> mengelola data kegiatan
Aktor	Bendahara
Kondisi awal	Aplikasi masuk ke halaman daftar kegiatan
Kondisi normal	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman daftar kegiatan</li> <li>2. Jika daftar kegiatan kosong maka, pilihan yang tersedia hanya menambahkan data. Jika tidak</li> </ol>

	<p>kosong, maka pilihan yang tersedia hanya edit dan hapus</p> <ol style="list-style-type: none"> <li>3. Jika <i>user</i> memilih untuk menambahkan atau mengedit data, maka sistem akan membuka form data kegiatan</li> <li>4. Sistem menampilkan halaman form data kegiatan</li> <li>5. <i>User</i> mengisi data pada form data kegiatan berupa nama kegiatan dan tanggal kegiatan</li> <li>6. Sistem memproses untuk menyimpan data</li> <li>7. Data tersimpan dan kembali ke halaman daftar kegiatan</li> <li>8. Jika user memilih untuk menghapus data, maka akan muncul dialog konfirmasi untuk menghapus</li> <li>9. Jika user memilih “yes”, maka data kegiatan akan terhapus,</li> <li>10. Jika user memilih “cancel” maka sistem akan kembali ke halaman daftar kegiatan</li> </ol>
Kondisi alternatif	Jika <i>field</i> kosong maka akan menampilkan pesan “field ini tidak boleh kosong”
Kondisi akhir	Data kegiatan akan tersimpan dan selalu diupdate di <i>database</i>

Adapun tabel skenario *use case* untuk mengelola data pengeluaran kegiatan adalah sebagai berikut.

**Tabel 3.4. Use case scenario mengelola data pengeluaran kegiatan**

No	04
Nama Use Case	Mengelola data pengeluaran kegiatan
Dekripsi	<i>user</i> mengelola data pengeluaran kegiatan
Aktor	Bendahara
Kondisi awal	<i>User</i> memilih salah satu item daftar kegiatan
Kondisi normal	1. <i>User</i> memilih salah satu item daftar kegiatan

	<ol style="list-style-type: none"> <li>2. Aplikasi menampilkan halaman daftar pengeluaran</li> <li>3. Jika daftar pengeluaran kosong maka, pilihan yang tersedia hanya menambahkan data. Jika tidak kosong, maka pilihan yang tersedia hanya edit dan hapus</li> <li>4. Jika <i>user</i> memilih untuk menambahkan atau mengedit data, maka sistem akan membuka form data pengeluaran</li> <li>5. Sistem menampilkan halaman form data pengeluaran</li> <li>6. <i>User</i> mengisi data pada form data pengeluaran berupa nama pengeluaran, jumlah item, harga satuan, dan bukti pembayaran</li> <li>7. Sistem memproses untuk menyimpan data</li> <li>8. Data tersimpan dan kembali ke halaman daftar pengeluaran</li> <li>9. Jika user memilih untuk menghapus data, maka akan muncul dialog konfirmasi untuk menghapus</li> <li>10. Jika user memilih “yes”, maka data pengeluaran akan terhapus,</li> <li>11. Jika user memilih “cancel” maka sistem akan kembali ke halaman daftar pengeluaran</li> </ol>
Kondisi alternatif	Jika <i>field</i> kosong maka akan menampilkan pesan “field ini tidak boleh kosong”
Kondisi akhir	Data pengeluaran kegiatan akan tersimpan dan selalu diupdate di <i>database</i>

Adapun tabel skenario *use case* untuk mengelola data aset adalah sebagai berikut.

**Tabel 3.5. Use case scenario mengelola data aset**

No	05
Nama Use Case	Mengelola data aset
Dekripsi	<i>user</i> mengelola data aset
Aktor	Bendahara
Kondisi awal	Aplikasi masuk ke halaman daftar aset
Kondisi normal	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman daftar aset</li> <li>2. Jika daftar aset kosong maka, pilihan yang tersedia hanya menambahkan data. Jika data tidak kosong, maka pilihan yang tersedia hanya edit dan hapus</li> <li>3. Jika <i>user</i> memilih untuk menambahkan atau mengedit data, maka sistem akan membuka form data aset</li> <li>4. Sistem menampilkan halaman form data aset</li> <li>5. <i>User</i> mengisi data pada form data aset berupa nama aset, harga aset, tanggal pembelian dan bukti pembayaran.</li> <li>6. Sistem memproses untuk menyimpan data</li> <li>7. Data tersimpan dan kembali ke halaman daftar aset</li> <li>8. Jika user memilih untuk menghapus data, maka akan muncul dialog konfirmasi untuk menghapus</li> <li>9. Jika user memilih “yes”, maka data aset akan terhapus,</li> <li>10. Jika user memilih “cancel” maka sistem akan kembali ke halaman daftar aset</li> </ol>
Kondisi alternatif	Jika <i>field</i> kosong maka akan menampilkan pesan “field ini tidak boleh kosong”
Kondisi akhir	Data aset akan tersimpan dan selalu diupdate di <i>database</i>

Adapun tabel skenario *use case* untuk mengelola data kas adalah sebagai berikut.

**Table 3.6. Use case scenario mengelola data kas**

No	06
Nama Use Case	Mengelola data kas
Dekripsi	<i>user</i> mengelola data kas
Aktor	Bendahara
Kondisi awal	Aplikasi masuk ke halaman daftar kas
Kondisi normal	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman daftar kas</li> <li>2. Jika daftar kas kosong maka, pilihan yang tersedia hanya menambahkan data. Jika tidak kosong, maka pilihan yang tersedia hanya edit dan hapus</li> <li>3. Jika <i>user</i> memilih untuk menambahkan atau mengedit data, maka sistem akan membuka form data kas</li> <li>4. Sistem menampilkan halaman form data kas</li> <li>5. <i>User</i> mengisi data pada form data kas berupa nama pembayar, jumlah yang telah dibayarkan, dan tanggal terakhir membayar</li> <li>6. Sistem memproses untuk menyimpan data</li> <li>7. Data tersimpan dan kembali ke halaman daftar kas</li> <li>8. Jika user memilih untuk menghapus data, maka akan muncul dialog konfirmasi untuk menghapus</li> <li>9. Jika user memilih “yes”, maka data kas akan terhapus,</li> <li>10. Jika user memilih “cancel” maka sistem akan kembali ke halaman daftar kas</li> </ol>
Kondisi alternatif	Jika <i>field</i> kosong maka akan menampilkan pesan “field ini tidak boleh kosong”

Kondisi akhir	Data kas akan tersimpan dan selalu diupdate di <i>database</i>
---------------	--

Adapun tabel skenario *use case* untuk melihat data pagudana adalah sebagai berikut.

**Tabel 3.7. Use case scenario melihat data pagudana**

No	07
Nama Use Case	Melihat data pagudana
Dekripsi	<i>user</i> melihat data kas
Aktor	Ketua
Kondisi awal	Aplikasi masuk ke halaman pagudana
Kondisi normal	Aplikasi menampilkan halaman pagudana
Kondisi alternatif	Jika data kosong maka akan menampilkan pesan error
Kondisi akhir	Informasi data pagudana telah ditampilkan

Adapun tabel skenario *use case* untuk melihat data kegiatan adalah sebagai berikut.

**Tabel 3.8. Use case scenario melihat data kegiatan**

No	08
Nama Use Case	Melihat data kegiatan
Dekripsi	<i>user</i> melihat data kegiatan
Aktor	Ketua
Kondisi awal	Aplikasi masuk ke halaman daftar kegiatan
Kondisi normal	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman daftar kegiatan</li> <li>2. <i>User</i> melakukan pencarian dengan data yang ditampilkan berdasarkan hasil pencarian</li> </ol>
Kondisi alternatif	Jika data kosong maka akan menampilkan pesan error
Kondisi akhir	Informasi data kegiatan telah ditampilkan

Adapun tabel skenario *use case* untuk melihat data pengeluaran kegiatan adalah sebagai berikut.

**Tabel 3.9. Use case scenario melihat data pengeluaran kegiatan**

No	09
Nama Use Case	Melihat data pengeluaran kegiatan
Dekripsi	<i>user</i> melihat data pengeluaran kegiatan
Aktor	Ketua
Kondisi awal	Aplikasi masuk ke halaman daftar pengeluaran kegiatan
Kondisi normal	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman daftar pengeluaran kegiatan</li> <li>2. <i>User</i> melakukan pencarian dengan data yang ditampilkan berdasarkan hasil pencarian</li> </ol>
Kondisi alternatif	Jika data kosong maka akan menampilkan pesan error
Kondisi akhir	Informasi data pengeluaran kegiatan telah ditampilkan

Adapun tabel skenario *use case* untuk melihat data aset adalah sebagai berikut.

**Tabel 3.10. Use case scenario melihat data aset**

No	10
Nama Use Case	Melihat data aset
Dekripsi	<i>user</i> melihat data aset
Aktor	Ketua
Kondisi awal	Aplikasi masuk ke halaman daftar aset
Kondisi normal	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman daftar aset</li> <li>2. <i>User</i> melakukan pencarian dengan data yang ditampilkan berdasarkan hasil pencarian</li> </ol>
Kondisi alternatif	Jika data kosong maka akan menampilkan pesan error
Kondisi akhir	Informasi data aset telah ditampilkan

Adapun tabel skenario *use case* untuk melihat data kas adalah sebagai berikut.

**Tabel 3.11. Use case scenario melihat data kas**

No	11
Nama Use Case	Melihat data kas
Dekripsi	<i>user</i> melihat data kas
Aktor	Ketua
Kondisi awal	Aplikasi masuk ke halaman daftar kas
Kondisi normal	<ol style="list-style-type: none"> <li>1. Aplikasi menampilkan halaman daftar kas</li> <li>2. <i>User</i> melakukan pencarian dengan data yang ditampilkan berdasarkan hasil pencarian</li> </ol>
Kondisi alternatif	Jika data kosong maka akan menampilkan pesan error
Kondisi akhir	Informasi data kas telah ditampilkan

Adapun tabel skenario *use case* untuk melihat data laporan adalah sebagai berikut.

**Tabel 3.12. Use case scenario melihat data laporan**

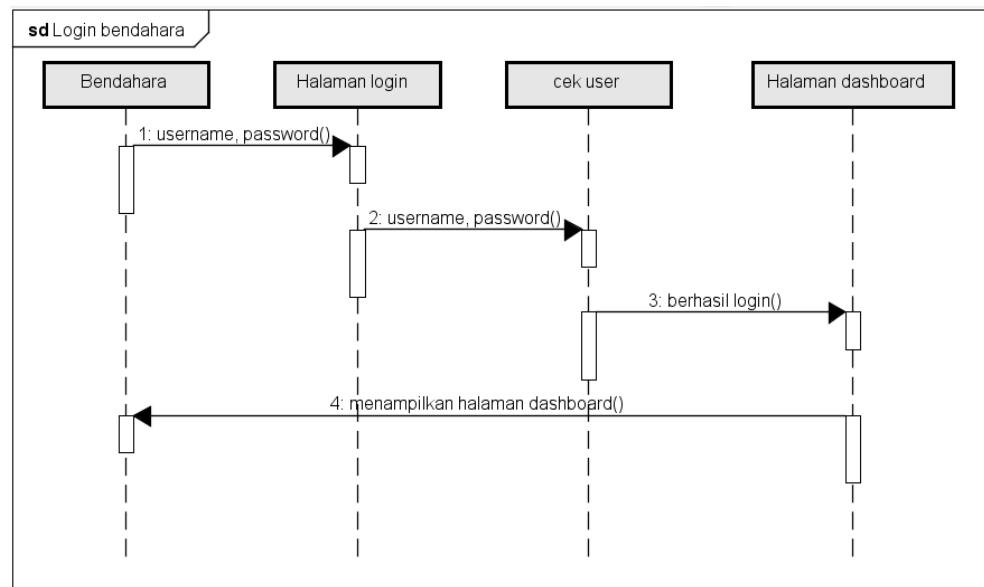
No	12
Nama Use Case	Melihat data laporan
Dekripsi	<i>user</i> melihat data laporan
Aktor	Bendahara dan Ketua
Kondisi awal	Aplikasi masuk ke halaman laporan
Kondisi normal	Aplikasi menampilkan halaman laporan
Kondisi alternatif	Jika data kosong maka akan menampilkan pesan error
Kondisi akhir	Informasi data laporan telah ditampilkan

### 3.5. Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *message* terhadap waktu. Pembuatan *sequence diagram* bertujuan perancangan aplikasi lebih mudah dan terarah.

#### 3.5.1. Sequence Diagram Login Pengguna

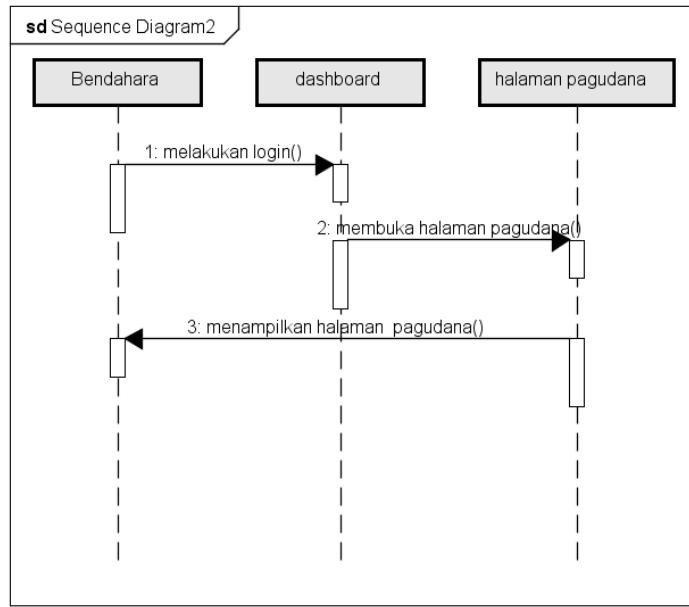
Pengguna yang sudah memiliki akun melakukan login dengan memasukan *username* dan *password* di *field* yang sudah disediakan, kemudian sistem akan melakukan verifikasi apakah akun yang di *inputkan* ada di *database* serta *username* dan *password* benar penulisannya. Kemudian setelah berhasil diverifikasi, halaman *dashboard* akan ditampilkan. Adapun *sequence diagram* login bendahara adalah sebagai berikut.



Gambar 3.3. *Sequence diagram* login bendahara

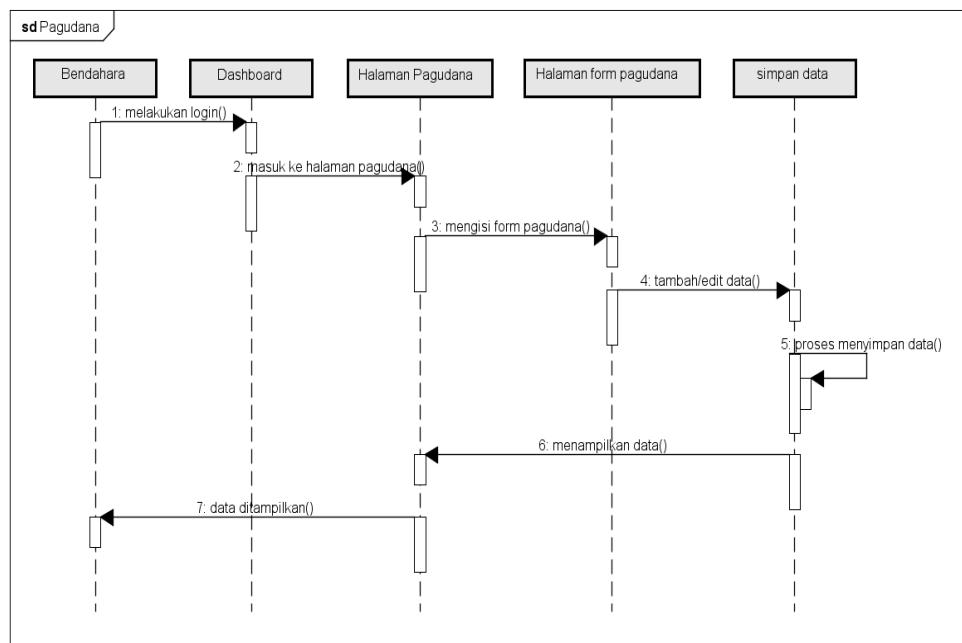
#### 3.5.2. Sequence Diagram Pagudana

Proses untuk menampilkan halaman pagudana digambarkan dengan *sequence diagram* sebagai berikut.



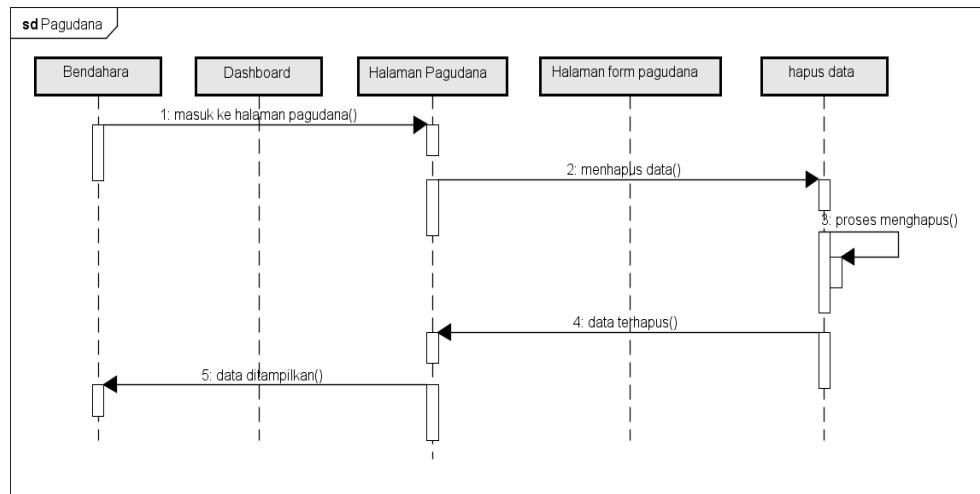
Gambar 3.4. *Sequence diagram* menampilkan data pagudana

Setelah halaman pagudana terbuka, jika data kosong, pilihan yang tersedia untuk menambahkan data, jika data sudah ada, pilihan yang tersedia ada pilihan untuk mengedit data dan menghapus data. Adapun *sequence diagram* untuk menambahkan dan mengedit data adalah sebagai berikut.



Gambar 3.5. *Sequence diagram* menambahkan dan mengedit data pagudana

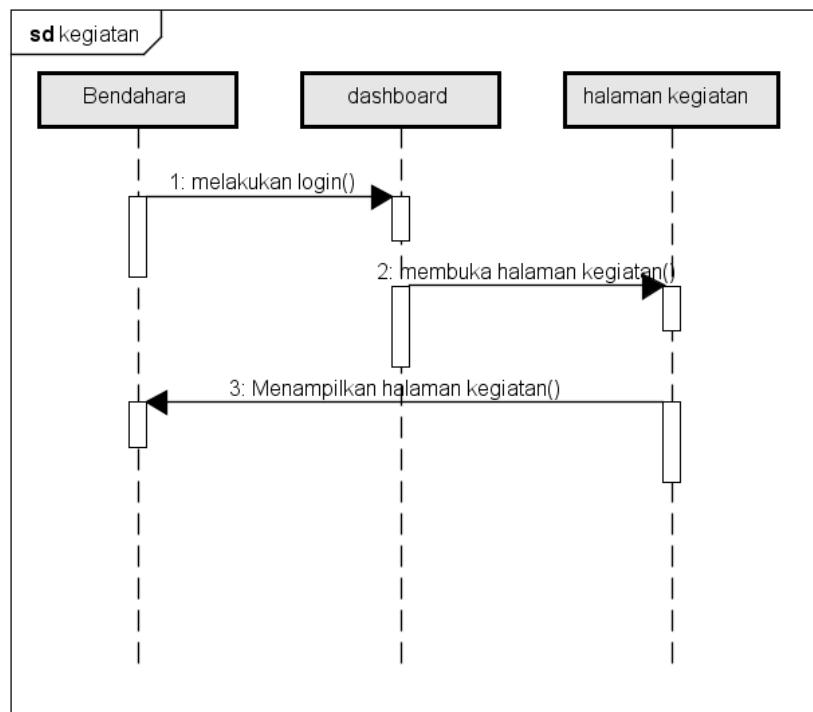
Dan untuk proses menghapus data, menggunakan *sequence diagram* seperti gambar dibawah.



Gambar 3.6. *Sequence diagram* menghapus data pagudana

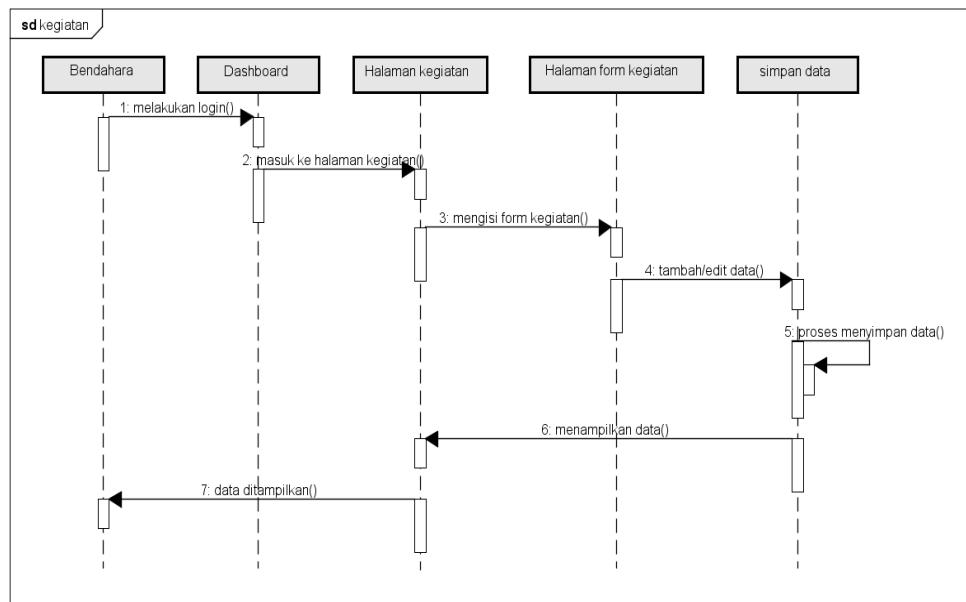
### 3.5.3. Sequence Diagram kegiatan

Proses untuk menampilkan halaman daftar kegiatan, menggunakan *sequence diagram* seperti gambar dibawah.



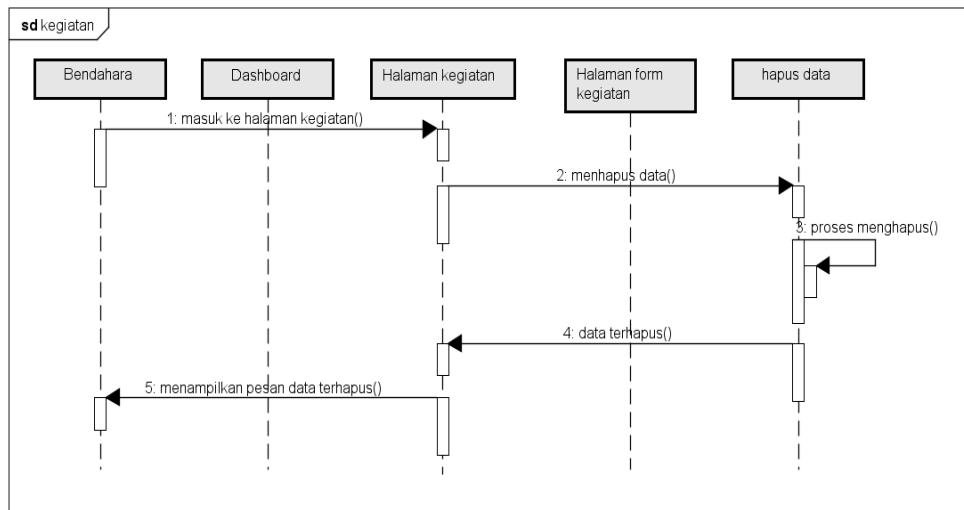
Gambar 3.7. *Sequence diagram* menampilkan data kegiatan

Jika pengguna ingin menambahkan data atau mengedit data, pengguna akan membuka halaman form kegiatan, kemudian mengikan data kemudian menekan tombol simpan. Setelah itu sistem akan memproses dan menyimpan data tersebut di *database*. Untuk proses penmbahan dan pengubahan data dilakukan dengan menggunakan *sequence diagram* sebagai berikut.



Gambar 3.8. *Sequence diagram* menambah dan mengedit data kegiatan

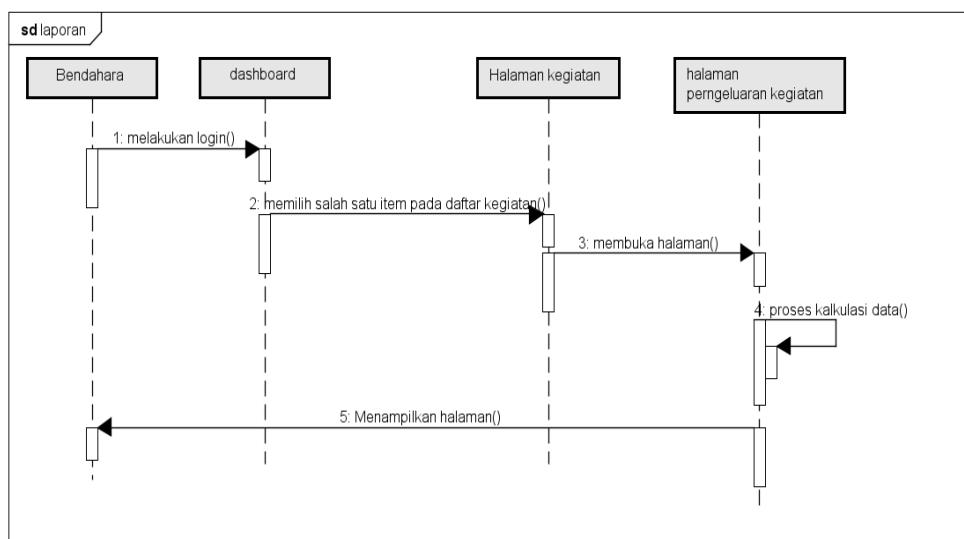
Dan untuk menghapus data, prosesnya digambarkan dengan *sequence diagram* berikut.



Gambar 3.9. *Sequence diagram* menghapus data kegiatan

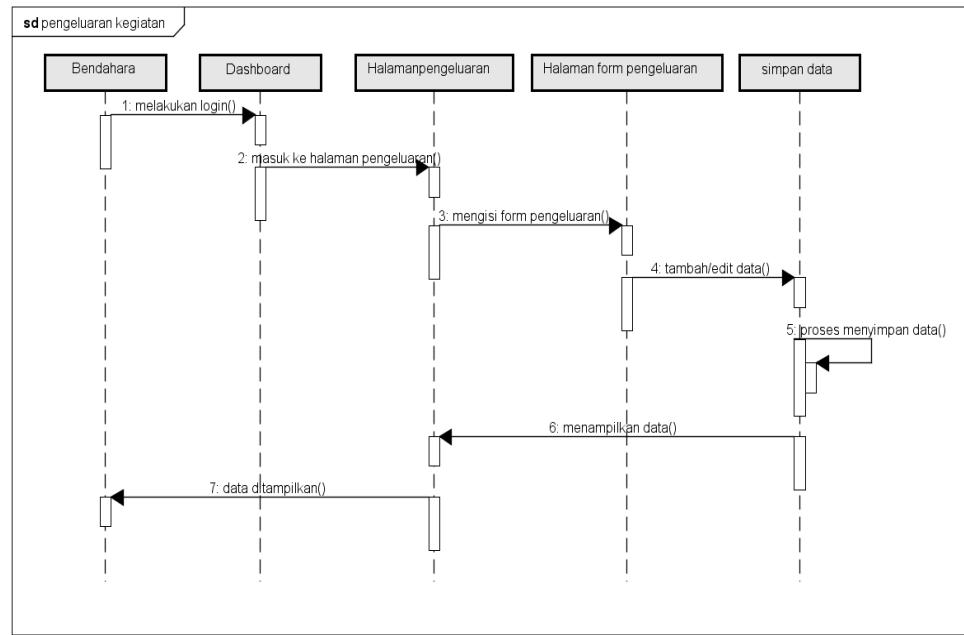
### 3.5.4. Sequence Diagram Pengeluaran Kegiatan

Ketika pengguna memilih salah satu item pada daftar kegiatan, kemudian akan membuka halaman pengeluaran berdasarkan kegiatan yang dipilih. Berikut merupakan sequence diagram untuk menampilkan halaman pengeluaran kegiatan.



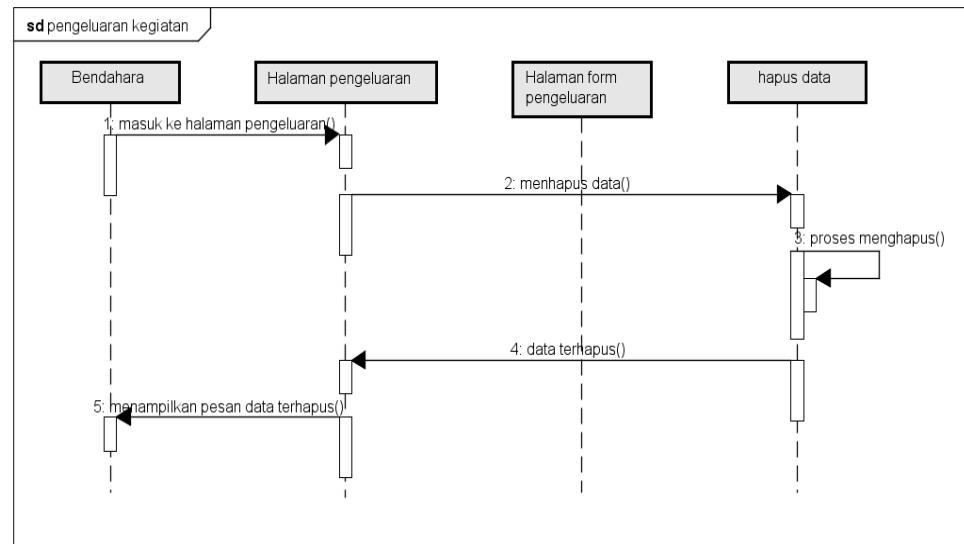
Gambar 3.10. *Sequence diagram* menampilkan data pengeluaran kegiatan

Proses untuk menambahkan data dan menedit data digambarkan dengan *sequence diagram* berikut.



Gambar 3.11. *Sequence diagram* menambah dan mengubah data pengeluaran

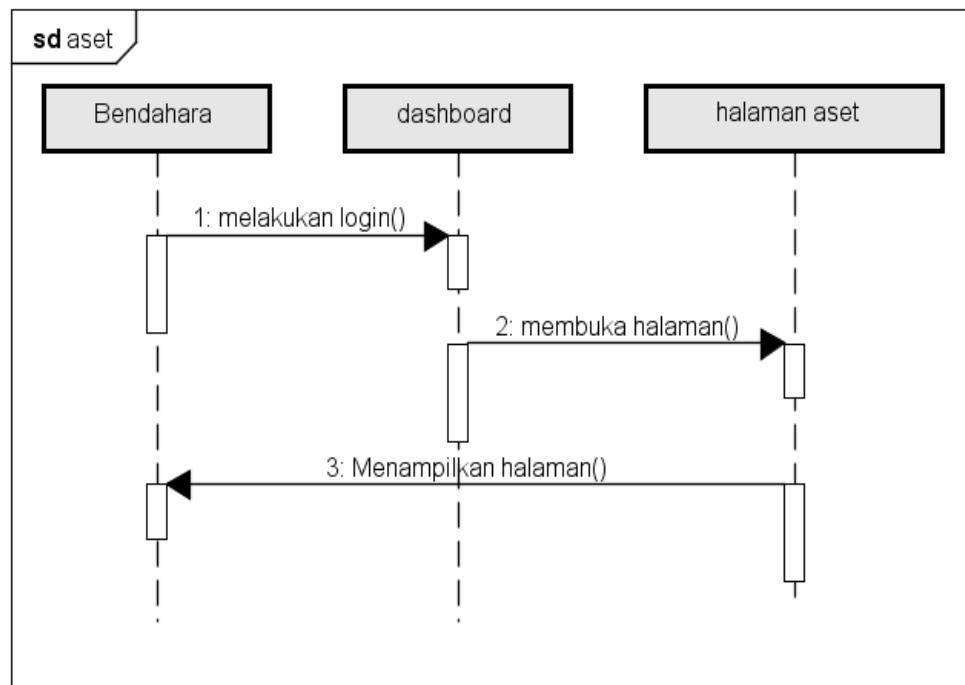
Proses untuk menghapus data pengeluaran kegiatan digambarkan dengan sequence diagram berikut.



Gambar 3.12. *Sequence diagram* menghapus data penggeluaran kegiatan

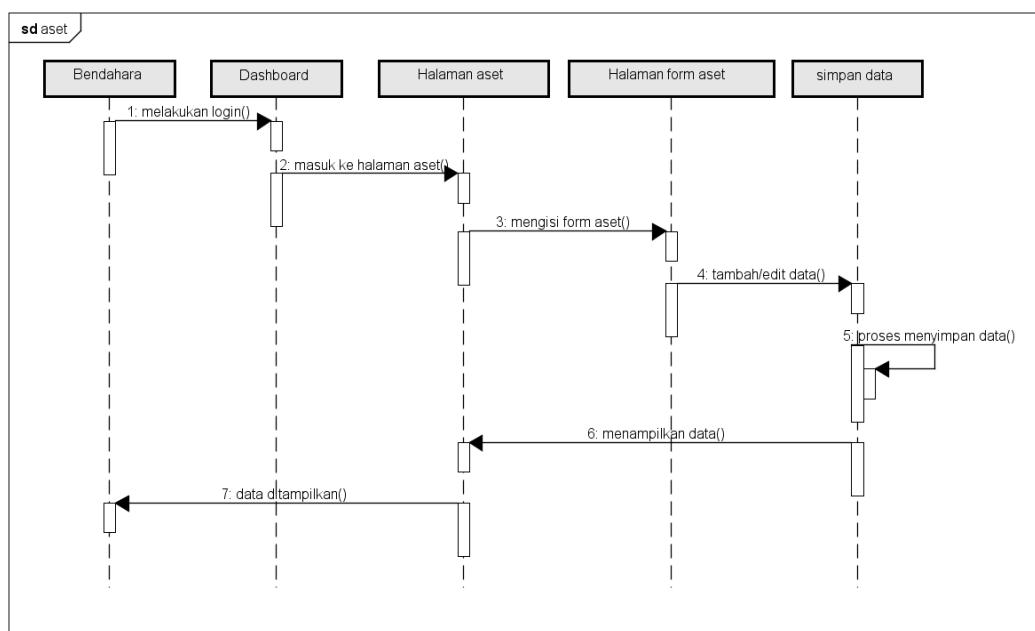
### 3.5.5. Sequence Diagram asset

Proses penampilan halaman daftar asset digambarkan dengan sequence diagram sebagai berikut.



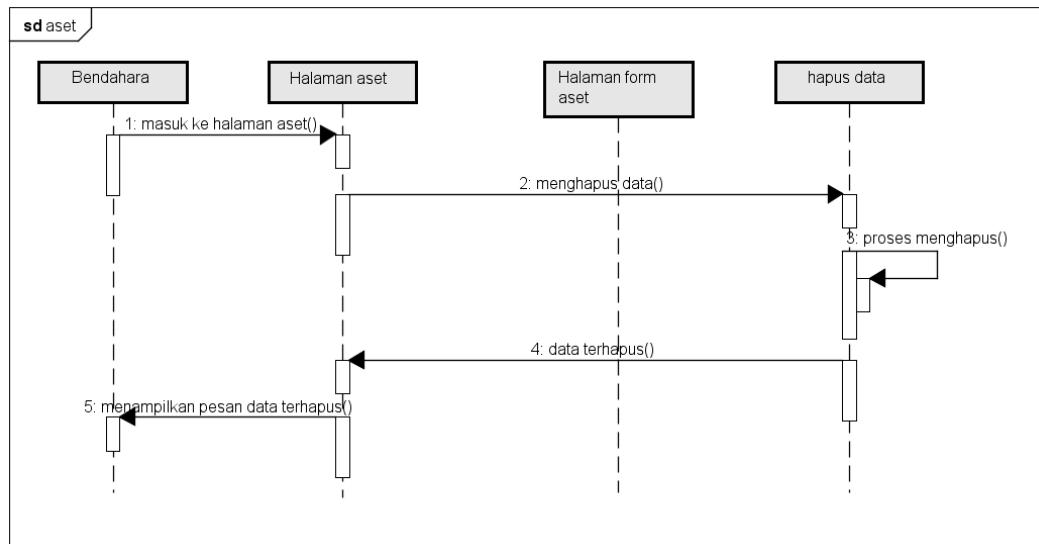
Gambar 3.13. *Sequence diagram* menampilkan data asset

Untuk proses menambah data dan mengedit data asset digambarkan didalam *sequence diagram* dibawah.



Gambar 3.14. *Sequence diagram* menambah dan mengubah data asset

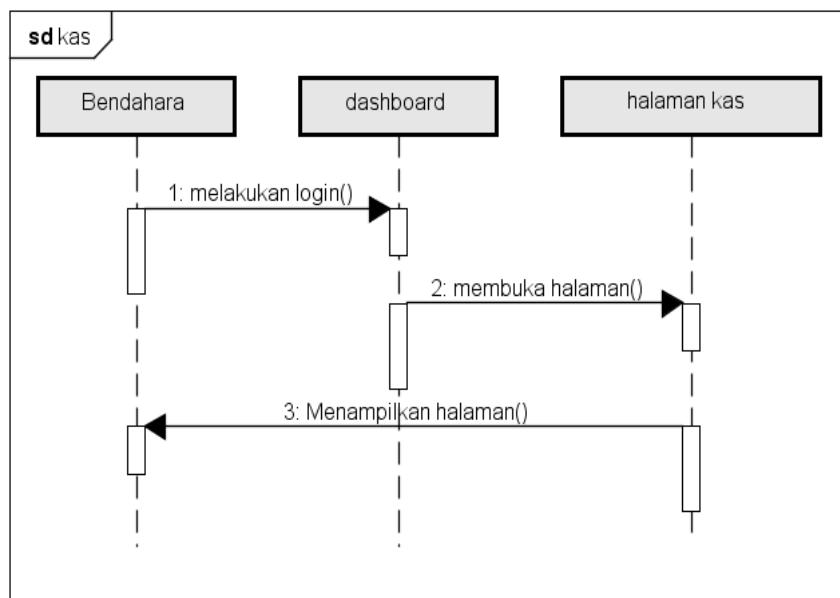
Untuk proses ketika menghapus data digambarkan pada sequence diagram dibawah.



Gambar 3.15. *Sequence diagram* menghapus data aset

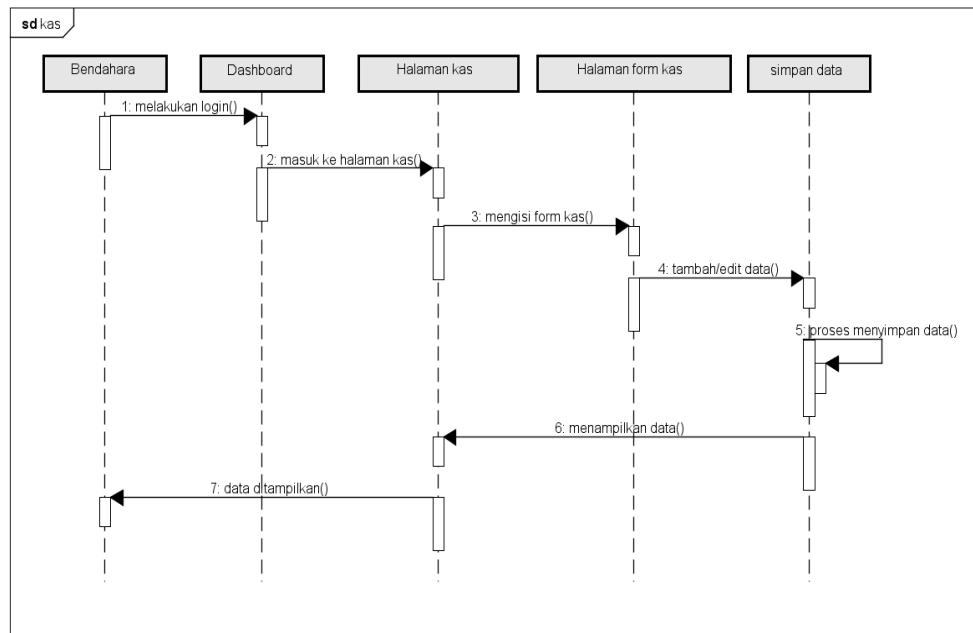
### 3.5.6. Sequence Diagram kas

Proses penampilan halaman daftar kas digambarkan dengan sequence diagram sebagai berikut.



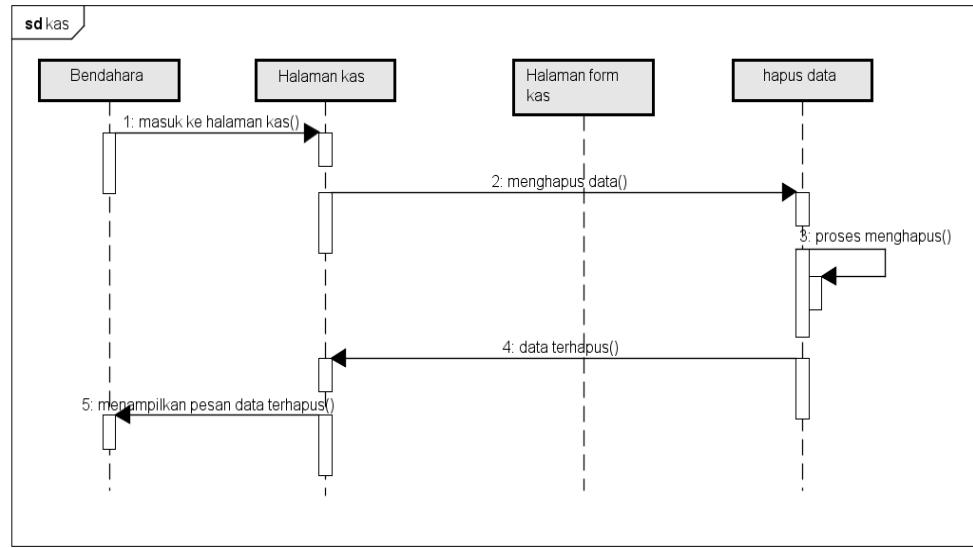
Gambar 3.16. *Sequence diagram* menampilkan data kas

Proses menambah dan mengedit data kas digambarkan dengan sequence diagram berikut.



Gambar 3.17. Sequence diagram menambah dan mengubah data kas

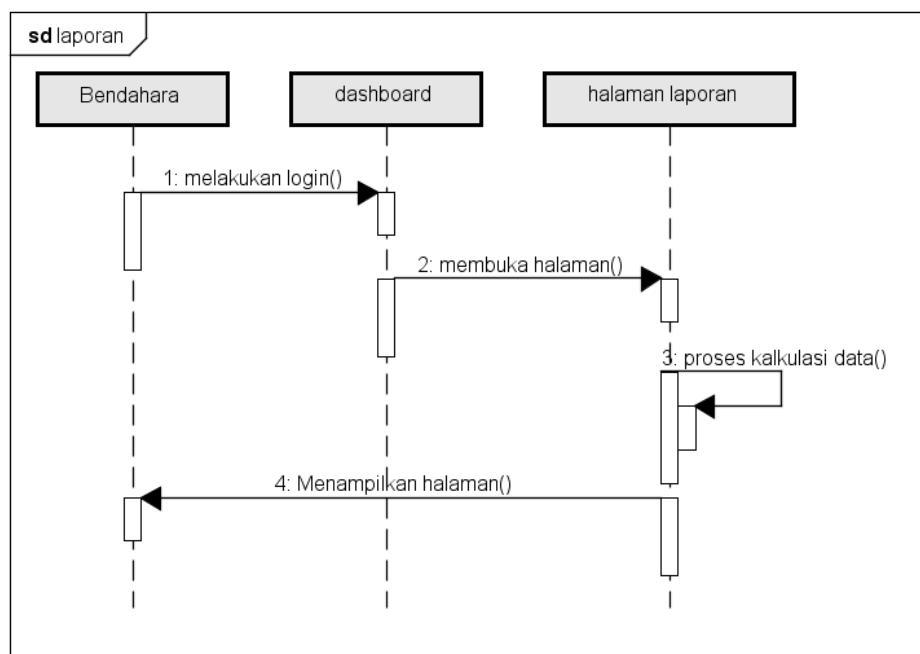
Untuk proses menghapus data kas digambarkan dengan sequence diagram berikut.



Gambar 3.18. Sequence diagram menghapus data kas

### 3.5.7. Sequence Diagram Laporan

Pada menu laporan dana manampilkan total pengeluaran dan sisa pagudana dengan menghitung data pada menu pengeluaran kegiatan, aset, kas, dan pagudana kemudian ditampilkan pada halaman laporan. Adapun proses penampilan laporan dana digambarkan dengan *sequence diagram* dibawah



Gambar 3.19. *Sequence diagram* menampilkan data laporan dana

## 3.6. Perancangan Database

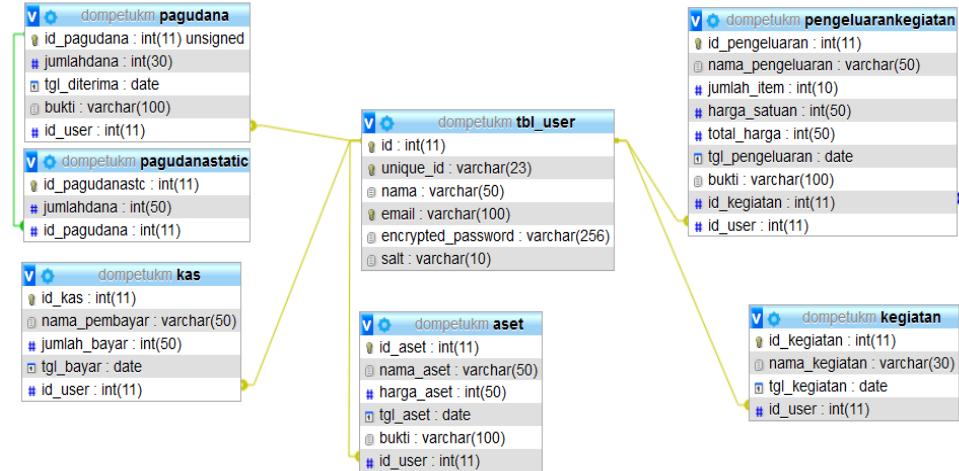
Adapun rancangan *database* pada aplikasi manajemen keuangan IPTEK ini adalah sebagai berikut:

### 3.6.1. Relasi Tabel

Tabel relasi merupakan sekumpulan tabel yang saling berelasi atau berhubungan. Tabel-tabel tersebut terletak dalam satu *database* yang sama. Pada suatu tabel terdapat *primary key* (kunci utama) dan jika tabel tersebut berelasi, maka harus memiliki *foreign key* (kunci tamu) di tabel lainnya.

Adapun relasi yang ada di aplikasi ini diantaranya, tabel user berelasi dengan tabel pagudana, tabel kegiatan, tabel pengeluaran kegiatan, tabel

aset, dan tabel kas, tabel pagudana berelasi dengan tabel pagudanastatic, serta tabel kegiatan berelasi dengan pengeluaran kegiatan.



Gambar 3.20. Relasi Tabel

### 3.6.2. Struktur Tabel

Berdasarkan rancangan dan kebutuhan pengguna, berikut merupakan tabel – tabel yang digunakan dalam pembuatan aplikasi ini.

Tabel 3.13. Tabel benda

Nama Field	Tipe data	keterangan
id	Int(11)	Primary key, not null
Unique_id	Varchar(23)	Unique, not null
Nama	Varchar(50)	not null
Email	Varchar(100)	not null
Encrypted_password	Varchar(256)	not null
salt	Varchar(10)	not null

**Tabel 3.14. Tabel pagudana**

Nama Field	Tipe data	keterangan
Id_pagudana	Int(11)	Primary key, not null
Jumlahdana	Int(30)	not null
Tgl_diterima	Date	not null
bukti	Varchar(100)	not null
Id_user	Varchar(11)	Foreign, not null

**Tabel 3.15 Tabel pagudana awal**

Nama Field	Tipe data	keterangan
Id_pagudanastc	Int(11)	Primary key, not null
Jumlahdana	Int(30)	not null
Id_pagudana	Int(11)	Foreign, not null

**Tabel 3.16. Tabel kegiatan**

Nama Field	Tipe data	keterangan
Id_kegiatan	Int(11)	Primary key, not null
Nama_kegiatan	Varchar(50)	not null
Tgl_kegiatan	date	not null
Id_user	Varchar(11)	Foreign, not null

**Tabel 3.17. Tabel pengeluaan kegiatan**

Nama Field	Tipe data	keterangan
Id_pengeluaran	Int(11)	Primary key, not null
Nama_pengeluaran	Varchar(50)	not null
Jumlah_item	Int(10)	not null
Harga_satuan	Int(50)	not null
Total_harga	Int(50)	not null
Tgl_pengeluaran	date	not null
bukti	Varchar(100)	not null
Id_kegiatan	Int(11)	Foreign, not null
Id_user	Int(11)	Foreign, not null

**Tabel 3.18. Tabel Aset**

Nama Field	Tipe data	keterangan
Id_aset	Int(11)	Primary key, not null
Nama_aset	Varchar(50)	not null
Harga_aset	Int(50)	not null
Tgl_aset	date	not null
bukti	Varchar(100)	Not null
Id_user	Int(11)	Foreign, not null

**Tabel 3.19. Tabel kas**

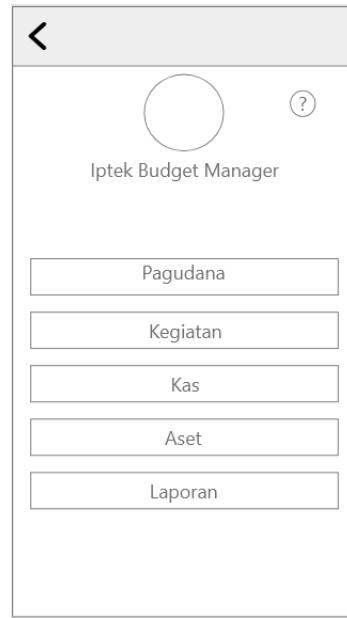
Nama Field	Tipe data	keterangan
Id_kas	Int(11)	Primary key, not null
Nama_pembayar	Varchar(50)	not null
Jumlah_bayar	Int(50)	not null
Tgl_bayar	date	not null
Id_user	Int(11)	Foreign, not null

### 3.7. Perancangan Interface

Berikut merupakan rancangan *interface*(antar muka) aplikasi manajemen keuangan IPTEK yang telah dibuat.

#### 3.7.1. Halaman Utama

Halaman dasar yang menjadi pusat dari aplikasi yang berisi menu – menu utama dari aplikasi. Gambar berikut merupakan rancangan *interface* dari halaman utama.



Gambar 3.21. Rancangan antarmuka halaman utama

### 3.7.2. Halaman Pagudana

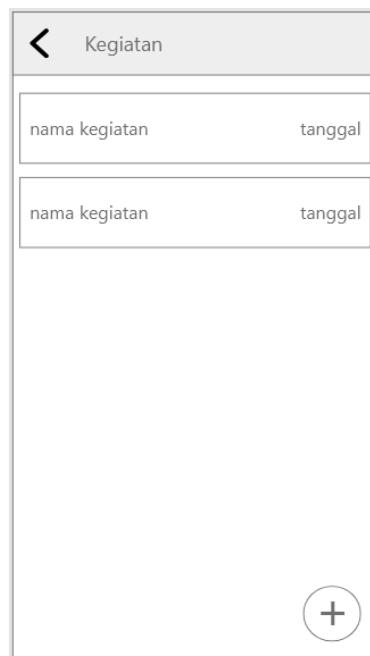
Halaman ini berfungsi untuk menampilkan informasi tentang pagudana berupa jumlah, tanggal diterima, dan bukti terima. Berikut merupakan rancangan dari halaman pagudana.



Gambar 3.22. Rancangan antarmuka halaman pagudana

### 3.7.3. Halaman Kegiatan

Halaman ini menampilkan daftar kegiatan yang telah masukan dengan menggunakan tampilan recyclerview. Berikut merupakan rancangan halaman kegiatan.



Gambar 3.23. Rancangan antarmuka halaman daftar kegiatan

### 3.7.4. Halaman Pengeluaran Kegiatan

Halaman ini menampilkan daftar pengeluaran per kegiatan yang ditampilkan ketika memilih salah satu item di daftar kegiatan. Berikut rancangan halaman pengeluaran kegiatan.

Rancangan antarmuka halaman daftar pengeluaran kegiatan. Tampilan ini menunjukkan daftar pengeluaran kegiatan yang telah disimpan. Di bagian atas terdapat tombol kembali (<) dan judul "Pengeluaran kegiatan". Berikutnya ada dua baris data dalam tabel:

nama barang	tanggal
harga X Jumlah item	

Dilanjut dengan dua baris data lainnya yang sama strukturnya:

nama barang	tanggal
harga X Jumlah item	

Pada bagian bawah terdapat simbol tambah (+) di dalam lingkaran.

Gambar 3.24. Rancangan antarmuka halaman daftar pengeluaran kegiatan

### 3.7.5. Halaman Daftar Aset

Halaman ini menampilkan daftar aset yang telah disimpan, berikut rancangan dari halaman daftar aset.

Rancangan antarmuka halaman daftar aset. Tampilan ini menunjukkan daftar aset yang telah disimpan. Di bagian atas terdapat tombol kembali (<) dan judul "Aset". Berikutnya ada dua baris data dalam tabel:

nama aset	tanggal
harga	

Dilanjut dengan dua baris data lainnya yang sama strukturnya:

nama aset	tanggal
harga	

Pada bagian bawah terdapat simbol tambah (+) di dalam lingkaran.

Gambar 3.25. Rancangan antarmuka halaman daftar aset

### 3.7.6. Halaman Daftar Kas

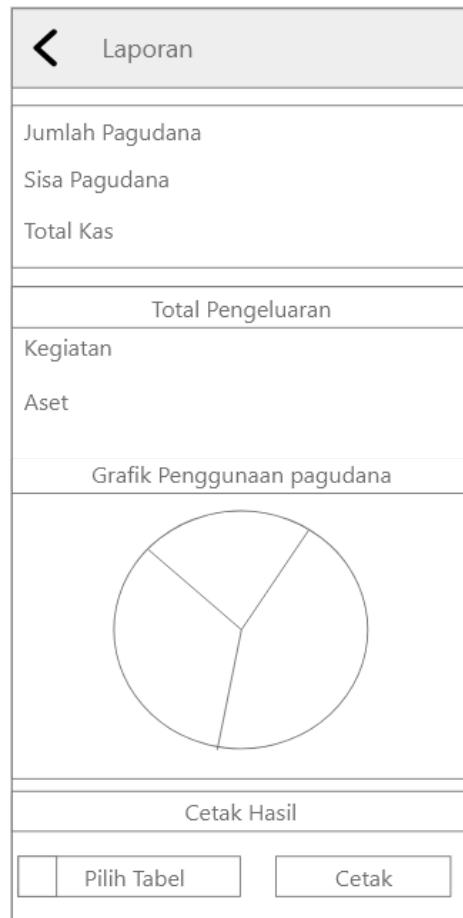
Halaman ini menampilkan daftar nama anggota yang telah atau pernah membayar uang kas dengan menampilkan nama pembayar, jumlah yang sudah dibayarkan, dan tanggal terakhir membayar.



Gambar 3.26. Rancangan antarmuka halaman daftar kas

### 3.7.7. Halaman laporan

Halaman ini menampilkan rincian dari pengeluaran dan penggunaan pagudana serta jumlah total kas yang terkumpul. Adapun rancangan dari halaman laporan dana adalah sebagai berikut.



Gambar 3.27. Rancangan antarmuka halaman laporan dana

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1. Implementasi Perangkat Lunak**

Perangkat lunak yang digunakan untuk membangun aplikasi ini adalah sebagai berikut:

- 1) Sistem operasi Windows 10
- 2) IDE Android Studio versi 3.5.0
- 3) Android SDK versi 30.0.2
- 4) Java
- 5) Xampp
- 6) Visual studio code
- 7) Memu android emulator
- 8) Sistem operasi android versi 7.1 (*Nougat*) API level 25

#### **4.2. Implementasi Perangkat Keras**

Rincian spesifikasi perangkat keras yang digunakan untuk membangun aplikasi ini adalah sebagai berikut:

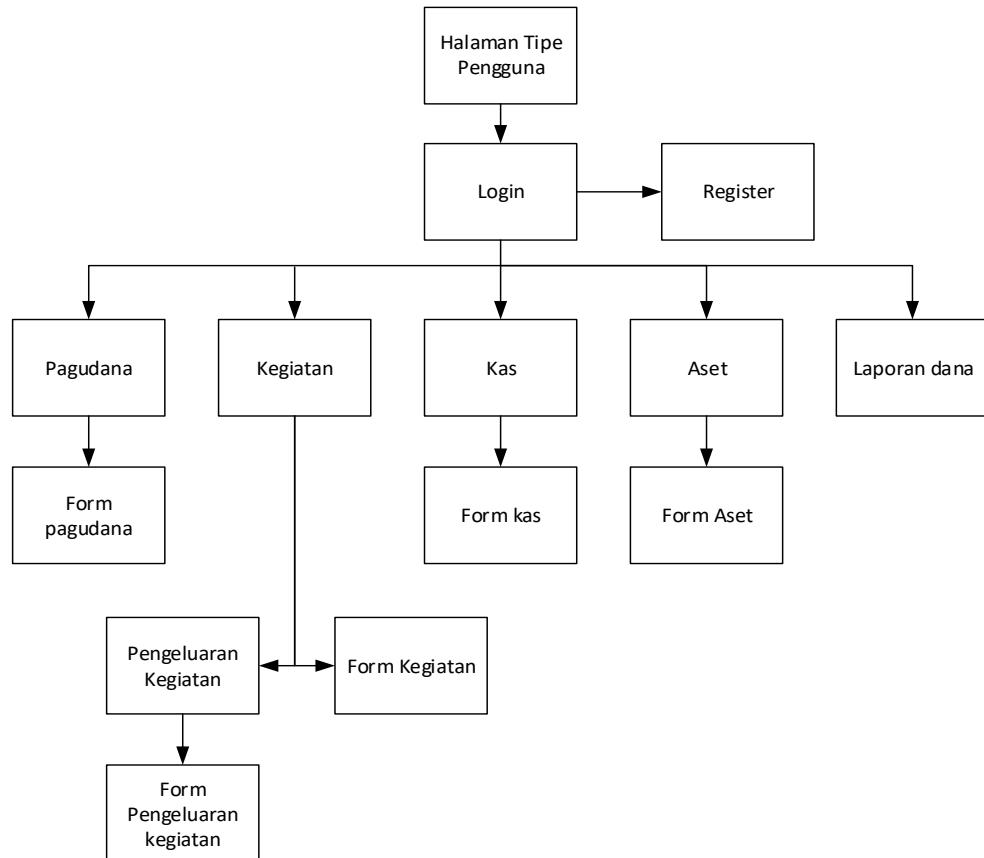
- 1) Amd A4 9125
- 2) RAM 8GB
- 3) Radeon R3
- 4) Harddisk 512GB
- 5) Smartphone android Huawei y7 Prime

#### **4.3. Peta Situs Aplikasi**

Peta situs atau *site map* adalah halaman yang dapat membantu pengguna dalam mencari informasi dari suatu aplikasi dalam bentuk struktural. Adapun peta situs untuk tipe pengguna bendahara dan anggota adalah sebagai beriku.

##### **4.3.1. Peta Situs Aplikasi Pada Tipe Pengguna Bendahara**

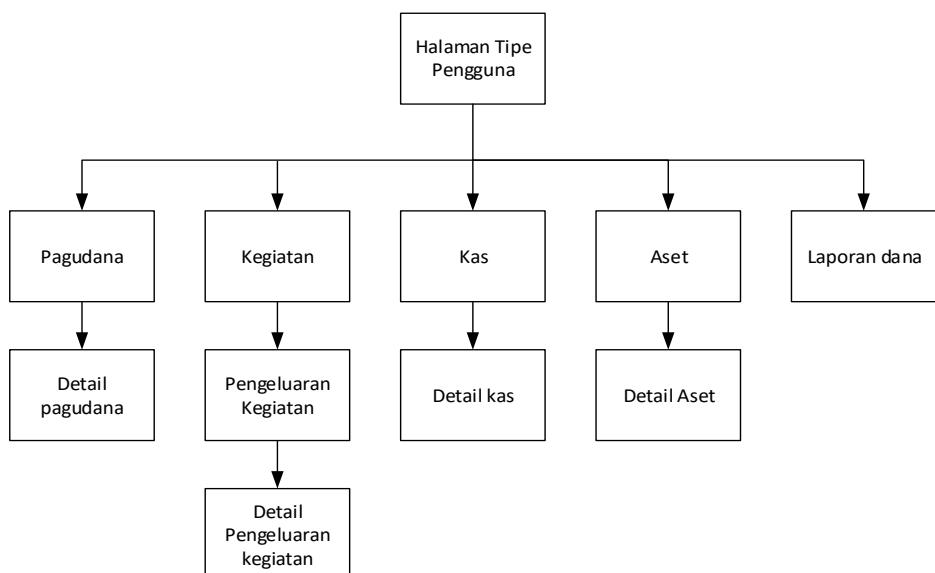
Adapun peta situs pada tipe pengguna Bendahara adalah sebagai berikut.



Gambar 4.1. Peta Situs untuk tipe pengguna Bendahara

#### 4.3.2. Peta Situs Aplikasi Pada Tipe Pengguna Ketua

Adapun peta situs pada tipe pengguna Bendahara adalah sebagai berikut.



Gambar 4.2. Peta situs untuk tipe pengguna Anggota

#### 4.4. Implementasi Sistem Pada Tipe Pengguna Bendahara

Implementasi sistem merupakan tahap akhir pembangunan sistem, yang dimana pada tahap ini, sistem akan diuji cobakan. Bendahara adalah tipe pengguna yang memiliki akses penuh pada aplikasi untuk mengelola data. Adapun hasil dari implementasi sistem pada tipe pengguna bendahara adalah sebagai berikut adalah sebagai berikut.

##### 4.3.1. Implementasi Tampilan *Splash Screen*

*Splash screen* memberikan waktu 4 detik dengan menampilkan logo dan nama aplikasi sebelum membuka halaman tipe pengguna. Halaman ini sebagai penanda bahwa sistem masih melakukan proses loading. Berikut merupakan tampilan dari halaman *splash screen*.



Gambar 4.3. Tampilan halaman *splash screen*

#### 4.3.2. Implementasi Tampilan Tipe Pengguna

Saat pertama kali aplikasi dibuka, halaman yang pertama kali terbuka adalah halaman tipe pengguna. Disini pengguna akan memilih tipe pengguna berdasarkan jabatan di organisasi. Ada dua jenis pengguna yaitu Bendahara dan Pengurus/anggota lainnya. Jika pengguna memilih bendahara, sistem akan menampilkan halaman login, jika pengguna memilih pengurus/anggota lainnya, sistem akan menampilkan *dashboard*(halaman utama) umum.



Gambar 4.4. Tampilan tipe pengguna

Pengguna memilih tipe pada *radio button*, kemudian menekan tombol masuk. Untuk mendapatkan aksi pada setiap opsi pilihan tipe pengguna, dengan menggunakan *switch case* seperti pada potongan program berikut.

```

public void onClick(View v) {
    int id = radio_usertype.getCheckedRadioButtonId();
    switch (id){
        case R.id.radio_bendahara :
            startActivity(new Intent( packageContext: UserType.this, LoginActivity.class ));
            finish();
            break;
        case R.id.radio_anggota :
            startActivity(new Intent( packageContext: UserType.this, Dashboard2Activity.class ));
            finish();
            break;
    }
}

```

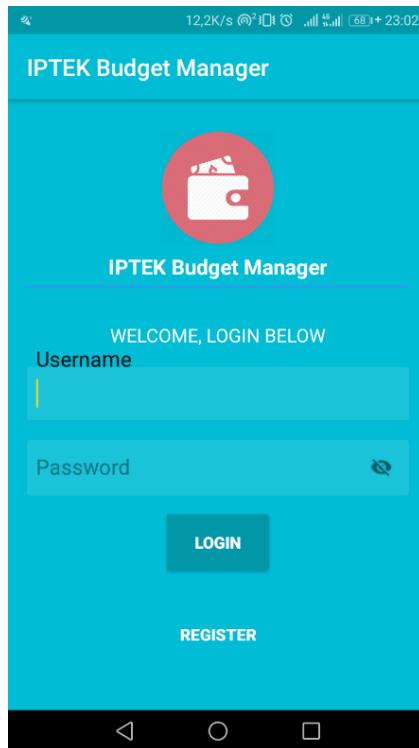
Gambar 4.5. Kode program pilihan tipe pengguna

*Radio\_bendahara* adalah *radio button* untuk pilihan bendahara, dan *radio\_anggota* adalah untuk pilihan pengurus/anggota lainnya. Untuk pilihan *radio\_bendahara* dengan menggunakan *intent* untuk pindah ke halaman login, sedangkan pilihan *radio\_anggota* pindah ke halaman *dashboard*(halaman utama).

#### 4.3.3. Implementasi Fungsi Login Pengguna

Jika pengguna memilih tipe pengguna bendahara, sistem akan menampilkan halaman login, pengguna akan memasukan informasi berupa username dan password jika sudah terdata di sistem.

Pengguna yang sudah terdaftar di *database*, melakukan login aplikasi dengan memasukan *username* dan *password* terdaftar untuk bisa masuk ke sistem. Adapun untuk tampilan halaman login seperti pada gambar.



Gambar 4.6. Tampilan *login* unruk bendahara

Setelah memasukan *username* dan *password*, sistem akan melakukan parsing data dari json di *frontend*(android studio) ke SQL di *backend*(REST API), dengan data akan di cek di *backend* apakah data tersebut ada serta huruf atau angka yang di inputkan benar.

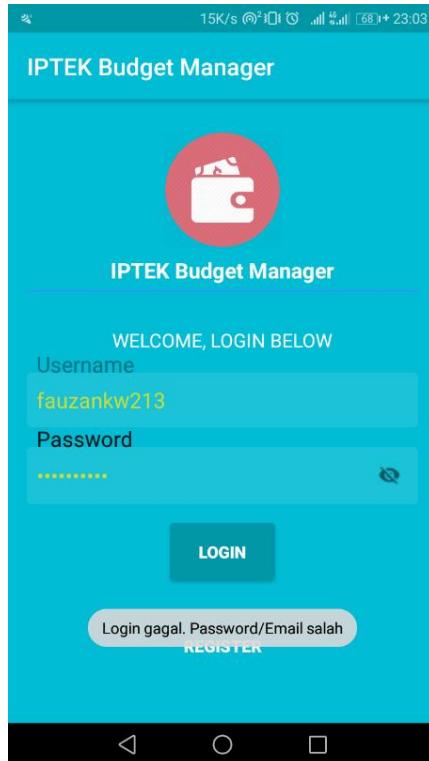
Pengecekan data *username* dan *password* dilakukan dengan mengambil data user dengan menggunakan perintah SQL *select* berdasarkan *username* yang di inputkan, kemudian, melakukan verifikasi *password* apakah *password* yang diinputkan sesuai atau tidak dengan menggunakan fungsi *checkhash* untuk mengecek *password* yang dienkripsi. Berikut merupakan potongan program pada *backend* untuk melakukan verifikasi *username* dan *password*.

```
/*
 * Get user berdasarkan username dan password
 */
public function getUserByUsernameAndPassword($username, $password) {
    $stmt = $this->conn->prepare("SELECT * FROM tbl_user WHERE email = ?");
    $stmt->bind_param("s", $username);
    if ($stmt->execute()) {
        $user = $stmt->get_result()->fetch_assoc();
        $stmt->close();

        // verifikasi password user
        $salt = $user['salt'];
        $encrypted_password = $user['encrypted_password'];
        $hash = $this->checkhashSSHA($salt, $password);
        // cek password jika sesuai
        if ($encrypted_password == $hash) {
            // autentikasi user berhasil
            return $user;
        }
    } else {
        return NULL;
    }
}
```

Gambar 4.7. Kode program untuk verifikasi akun bendahara

Kemudian jika telah melakukan verifikasi pada *username* dan *password*, akan dilakukan umpan balik ke pengguna dengan pesan jika *login* berhasil atau gagal. Jika *login* berhasil, sistem akan memindahkan halaman ke halaman utama dan jika gagal, akan muncul pesan bahwa *login* gagal.



Gambar 4.8. Tampilan pesan jika salah memasukan *username* atau *password*

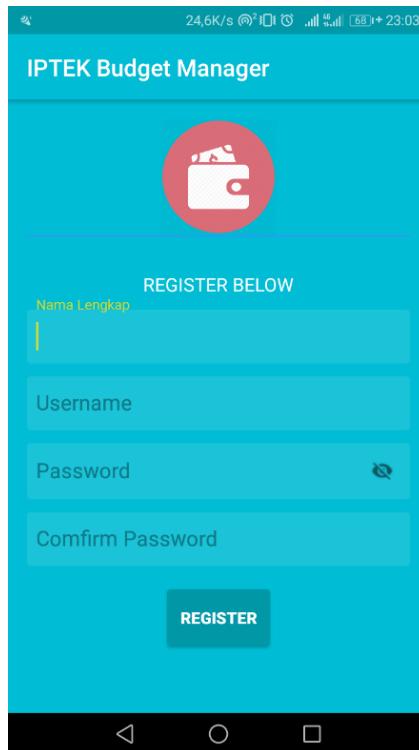
Adapun potongan program di sisi *backend* untuk menampilkan pesan bahwa *login gagal* adalah sebagai berikut.

```
// user tidak ditemukan password/username salah
$response["error"] = TRUE;
$response["error_msg"] = "Login gagal. Password/Username salah";
echo json_encode($response);
```

Gambar 4.9. Kode program untuk menampilkan pesan *error*

#### 4.3.4. Implementasi Fungsi Register Bendahara

Halaman *register* berfungsi untuk mendaftarkan pengguna jika belum terdaftar di *database*. Pada halaman *register* ini terdapat nama, *username*, dan *password* yang wajib diisi oleh pengguna. Adapun tampilan antarmuka halaman *register* seperti pada gambar.



Gambar 4.10. Tampilan halaman register

Setelah pengguna mengisi semua data dan menekan tombol *register*, sistem akan melakukan *parsing* data ke *backend* dan menyimpan data tersebut ke *database*. Untuk mengambil data yang diinputkan dan diberikan ke *backend* untuk diproses, menggunakan potongan program berikut.

```
private void requestRegister() {
    mApiService.registerRequest(txt_nama.getText().toString(),
        txt_username.getText().toString(),
        txt_password.getText().toString()
    .enqueue(new Callback<ResponseBody>() {
```

Gambar 4.11. Kode program untuk mengambil data hasil *input*

Kemudian di sisi backend data akan disimpan dengan menggunakan perintah SQL *insert*, serta mengenkripsi password dengan *hash* dan *salt*, dengan menggunakan potongan program berikut.

```

public function simpanUser($nama, $username, $password) {
    $uuid = uniqid('', true);
    $uuid = substr($unique, strlen($uuid) - 4, strlen($uuid));
    $hash = $this->hashSSHA($password);
    $encrypted_password = $hash["encrypted"]; // encrypted password
    $salt = $hash["salt"]; // salt

    $stmt = $this->conn->prepare("INSERT INTO tbl_user(unique_id, nama,
                                    email, encrypted_password, salt)
                                    VALUES(?, ?, ?, ?, ?)");
    $stmt->bind_param("sssss", $uuid, $nama, $username, $encrypted_password, $salt);
    $result = $stmt->execute();
    $stmt->close();

    // cek jika sudah sukses
    if ($result) {
        $stmt = $this->conn->prepare("SELECT * FROM tbl_user WHERE email = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $user = $stmt->get_result()->fetch_assoc();
        $stmt->close();

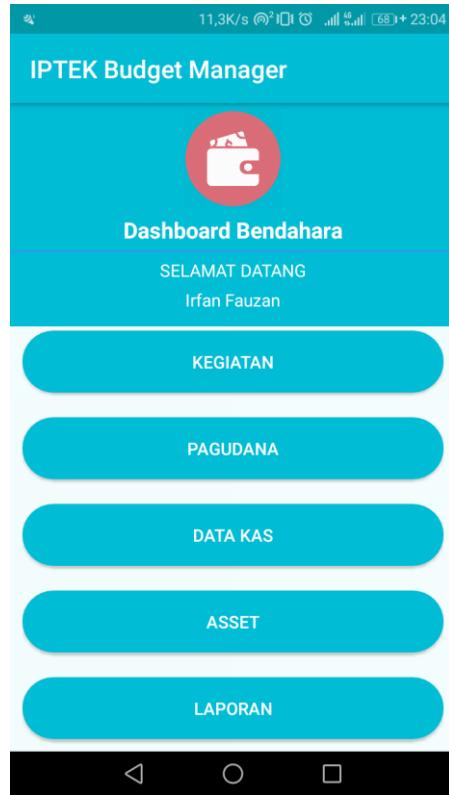
        return $user;
    } else {
        return false;
    }
}

```

Gambar 4.12. Kode program untuk menyimpan data akun bendahara

#### 4.3.5. Implementasi Tampilan Dashboard (Halaman Utama)

*Dashboard* merupakan tampilan dasar dari aplikasi yang merupakan pusat dari semua aktivitas sistem. Pada halaman ini, terdapat menu utama yang terdiri dari kegiatan, pagudana, aset, kas, laporan, panduan aplikasi dan tentang. Tampilan *dashboard* bendahara dan anggota berbeda. Berikut adalah tampilan dari *dashboard* bendahara dan anggota.



Gambar 4.13. Tampilan *dashboard* bendahara

#### 4.3.6. Implementasi Menu Pagudana

Halaman ini berfungsi untuk melihat detail informasi dari pagudana berupa jumlah pagudana, tanggal diterima, dan bukti terima. Pada pojok kanan atas terdapat tool untuk mengedit data dan menghapus data. Adapun tampilan dari halaman pagudana ini adalah sebagai berikut.



Gambar 4.14. Tampilan halaman pagudana

Untuk menampilkannya, dengan melakukan *request* ke *backend*, kemudian di *backend* akan mengeksekusi perintah SQL untuk menampilkan data – data tersebut. Data –data tersebut di parsing ke json yang kemudian akan diproses di *frontend*(android studio). Data yang sudah dalam format json tersebut dipanggil di *interface base ApiService* dengan menggunakan kode berikut.

```
@POST("get_dana.php")
Call<List<pagudana>> get_dana();
```

dan diimplementasikan di kelas *incomeActivity*. Berikut merupakan code untuk menampilkan data pagudana pada kelas *incomeActivity*.

```
public void tampilDana() {
    Call<List<pagudana>> call = baseApi.get_dana();
```

```

id_pagudana = addlist.get(0).getId_pagudana();
jumlahpagudana = addlist.get(0).getJumlahhdana();
tgl_diterima = addlist.get(0).getTgl_diterima();
bukti_diterima = addlist.get(0).getBukti();

jumlahdana.setText(formatRupiah.format((double)jumlahpagudana));
tglderima.setText(tgl_diterima);

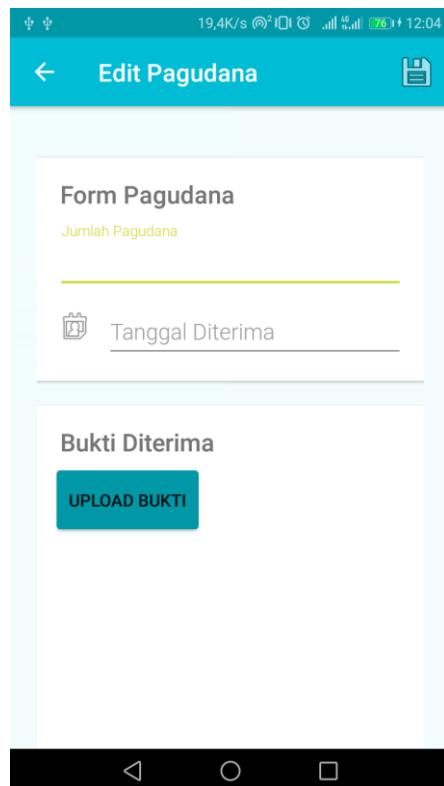
RequestOptions requestOptions = new RequestOptions();
requestOptions.skipMemoryCache(true);
requestOptions.diskCacheStrategy(DiskCacheStrategy.NONE);
requestOptions.placeholder(R.drawable.ic_menu_gallery);
requestOptions.error(R.drawable.ic_menu_gallery);

Glide.with( activity: IncomeActivity.this).load(bukti_diterima)
    .apply(requestOptions).into(buktiderima);

```

Gambar 4.15. Kode program untuk menampilkan data pagudana

Untuk mengedit atau menambahkan pagudana emnggunakan ikon edit dan tambah pada *action bar*. Kemudian akan muncul halaman form pagudana seperti pada gambar.

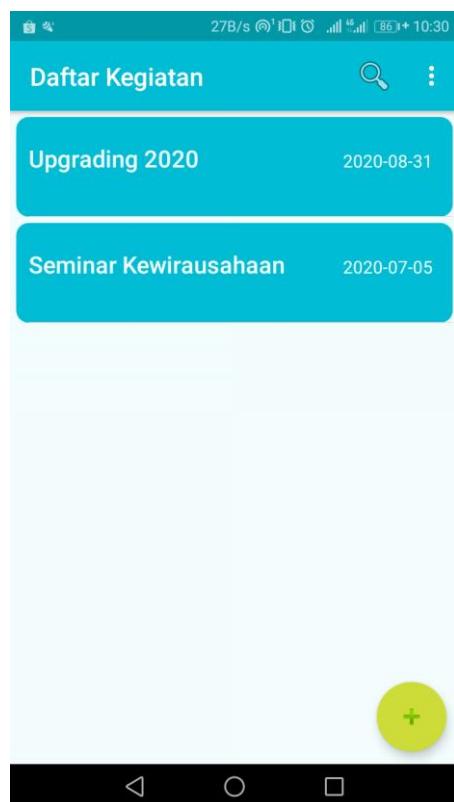


Gambar 4.16. Tampilan halaman form pagudana

Pada form pagudana terdapat jumlah dana, tanggal diterima dan *upload* bukti diterima.

#### 4.3.7. Implementasi Menu Daftar Kegiatan

Halaman ini menampilkan daftar kegiatan yang telah disimpan oleh sistem. Pada setiap item pada daftar kegiatan terdapat nama kegiatan, dan tanggal kegiatan. Daftar tersebut ditampilkan dengan menggunakan *Recyclerview* pada android studio yang berfungsi untuk menampilkan data berupa *array*. Sistem akan *retrieve* data dari *database*, kemudian menampilkan setiap item menggunakan *recyclerview adapter*, kemudian, melakukan pengulangan untuk menampilkan setiap item menjadi sebuah daftar. Berikut merupakan tampilan dari daftar kegiatan.



Gambar 4.17. Tampilan daftar kegiatan

Untuk mendapatkan data dari *database*, sistem melakukan *request* ke *backend*, kemudian di *backend*, dilakukan perintah SQL *select* untuk mengambil data dari *database*. Adapun untuk menampilkan daftar kegiatan, menggunakan potongan program berikut.

```
public void getkegiatan() {
    Call<List<kegiatan>> call = apiInterface.get_kegiatan();
    call.enqueue(new Callback<List<kegiatan>>() {
        @Override
        public void onResponse(Call<List<kegiatan>> call, Response<List<kegiatan>> response) {
            progressBar.setVisibility(View.GONE);
            kegiatanlist = response.body();
            Log.i(kegiatanList.class.getSimpleName(), response.body().toString());
            adapter = new kegiatanAdapter(kegiatanlist, context: kegiatanList.this, listener);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }

        @Override
        public void onFailure(Call<List<kegiatan>> call, Throwable t) {
            Toast.makeText(context: kegiatanList.this, text: "error :" +
                t.getMessage().toString(),
                Toast.LENGTH_SHORT).show();
        }
    });
}
```

Gambar 4.18. Kode program untuk menampilkan data kegiatan di *frontend*

Adapun di bagian *backend*, menggunakan potongan kode berikut.

```
$query = "SELECT * FROM kegiatan ORDER BY id_kegiatan DESC ";
$result = mysqli_query($conn, $query);
$response = array();

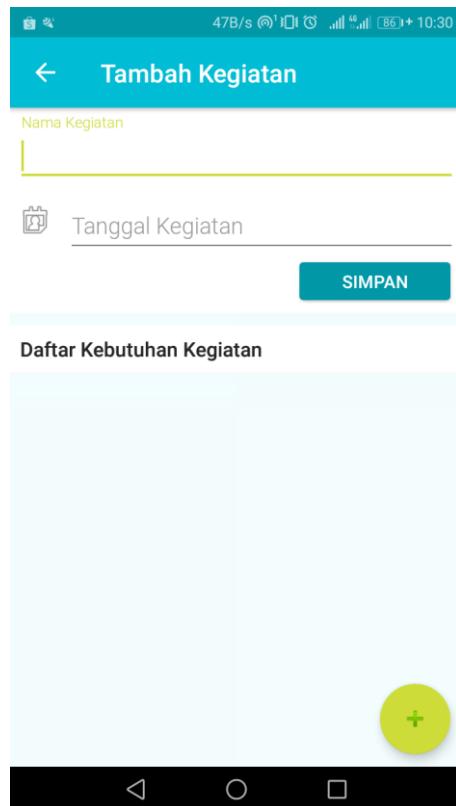
// $server_name = $_SERVER['SERVER_ADDR'];
$server_name = '192.168.43.209';

while( $row = mysqli_fetch_assoc($result) ){
    array_push($response,
    array(
        'id_kegiatan'      =>$row['id_kegiatan'],
        'nama_kegiatan'   =>$row['nama_kegiatan'],
        'tgl_kegiatan'     =>$row['tgl_kegiatan'])
    );
}
```

Gambar 4.19. Kode program untuk menampilkan data kegiatan di *backend*

Untuk menambahkan data, menggunakan tombol *floatingbutton* pada pojok kanan bawah, setelah itu sistem akan menampilkan halaman form

kegiatan untuk menginputkan data kegiatan berupa nama kegiatan dan tanggal kegiatan.



Gambar 4.20. Tampilan halaman form kegiatan dan daftar pengeluaran

Untuk menyimpan data, sama seperti di bagian *register* bendahara, sistem akan melakukan *request* ke *backend* untuk mengeksekusi perintah SQL. Untuk melakukan *request*, menggunakan potongan program seperti pada gambar dibawah.

```
private void postKegiatan(final String key){  
    String nama_kegiatan = nama_kegiatan_ed.getText().toString().trim();  
    String tgl_kegiatan = tgl_kegiatan_ed.getText().toString().trim();
```

```

apiInterface = UtilsApi.getAPIService();

Call<kegiatan> call = apiInterface.add_kegiatan(key, nama_kegiatan, tgl_kegiatan);

call.enqueue(new Callback<kegiatan>() {
    @Override
    public void onResponse(Call<kegiatan> call, Response<kegiatan> response) {

        progressDialog.dismiss();

        Log.i(eventneedsList.class.getSimpleName(), response.toString());

        String value = response.body().getValue();
        String message = response.body().getMessage();

        if (value.equals("1")){
            finish();
        } else {
            Toast.makeText(context: eventneedsList.this, message, Toast.LENGTH_SHORT).show();
        }
    }
})

```

Gambar 4.21. Kode program untuk melakukan *request* meyimpan data kegiatan

#### 4.3.8. Implementasi Menu Daftar Pengeluaran Kegiatan

Halaman ini berfungsi untuk mengisi atau mengubah data kegiatan sekaligus menampilkan daftar pengeluaran dari kegiatan tersebut. Pada daftar pengeluaran ini terdapat di setiap item pada daftar kegiatan setelah memilih salah satu item pada daftar kegiatan kemudian akan menampilkan halaman ini. Pada halaman ini terdapat nama kegiatan dan tanggal kegiatan dan daftar pengeluarannya berupa nama pengeluaran, harga, jumlah item, tanggal, serta gambar bukti pembayaran. Berikut adalah tampilan dari halaman pengeluaran kegiatan.

Untuk menampilkan daftar pengeluaran kegiatan, menggunakan metode dan kode yang sama dengan daftar kegiatan. Adapun tampilan dari daftar pengeluaran dapat dilihat pada gambar dibawah.



Gambar 4.22. Tampilan halaman daftar pengeluaran kegiatan

Adapun kode program untuk melakukan *request* menampilkan data pengeluaran adalah sebagai berikut

```
public void getPengeluaran(final int id_kegiatan){
    Call<List<eventneeds>> call = apiInterface.get_eventneeds(id_kegiatan);
    call.enqueue(new Callback<List<eventneeds>>() {
        @Override
        public void onResponse(Call<List<eventneeds>> call, Response<List<eventneeds>> response) {
            progressBar.setVisibility(View.GONE);
            eventneedslist = response.body();
            Log.i(eventneedsList.class.getSimpleName(), response.body().toString());
            adapter = new eventneedsAdapter(eventneedslist, context: eventneedsList.this, listener);
            recyclerView.setAdapter(adapter);
            adapter.notifyDataSetChanged();
        }

        @Override
        public void onFailure(Call<List<eventneeds>> call, Throwable t) {
            Toast.makeText(context: eventneedsList.this, text: "error :" +
                t.getMessage().toString(),
                Toast.LENGTH_SHORT).show();
        }
    });
}
```

Gambar 4.23. Kode program untuk menampilkan data pengeluaran kegiatan pada *frontend*

Dan kode pada *backend* untuk mendapatkan data pengeluaran dengan eksekusi perintah SQL *insert*. Adapun kode programnya sebagai berikut.

```
<?php
header("Content-Type: application/json; charset=UTF-8");

require_once 'connect.php';

$id_kegiatan = $_POST['id_kegiatan'];
$id_kegiatan = (int)$id_kegiatan;
$query = "SELECT * FROM pengeluarankegiatan WHERE id_kegiatan=$id_kegiatan ORDER BY id_pengeluaran DESC ";
$result = mysqli_query($conn, $query);
$response = array();

//$server_name = '10.0.2.2';
$server_name = '192.168.43.209';

while( $row = mysqli_fetch_assoc($result) ){
    array_push($response,
        array(
            'id_pengekuaran' =>$row['id_pengeluaran'],
            'nama_pengekuaran' =>$row['nama_pengeluaran'],
            'jumlah_item' =>$row['jumlah_item'],
            'harga_satuan' =>$row['harga_satuan'],
            'tgl_pengekuaran' =>date('d M Y', strtotime($row['tgl_pengekuaran'])),
            'bukti' =>"http://$server_name" . $row['bukti'],
            'id_kegiatan' =>$row['id_kegiatan'],
            'nama_kegiatan' =>$row['id_kegiatan']
        )
    );
}

echo json_encode($response);

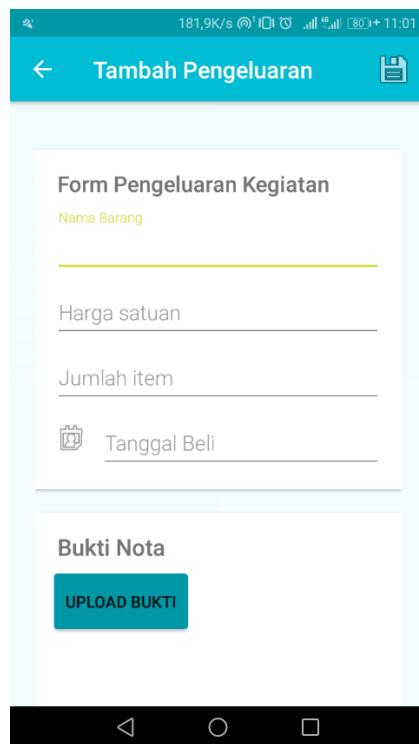
mysqli_close($conn);

?>
```

Gambar 4.24. Kode program untuk menampilkan data pengeluaran pada *backend*

Pada pojok kanan atas terdapat icon untuk mengedit dan menghapus data kegiatan, dan untuk menambah data pengeluaran pada kegiatan tersebut, menggunakan tombol pada pojok kanan bawah.

Pada halaman form pengeluaran kegiatan ini merupakan tempat untuk melihat detail dan mengelola data seperti menambah, mengedit, dan menghapus data. Menu untuk mengelola data ada di pojok kanan atas. Dari tampilan form pengeluaran kegiatan ini ada dua mode yaitu *read mode* dan *edit mode*. Adapun tampilan dari form pengeluaran kegiatan dengan *edit mode* adalah sebagai berikut..



Gambar 4.25. Tampilan form pengeluaran kegiatan dalam *edit mode*

Adapun untuk tampilan *read mode* adalah sebagai berikut.



Gambar 4.26. Tampilan form pengeluaran kegiatan dalam *read mode*

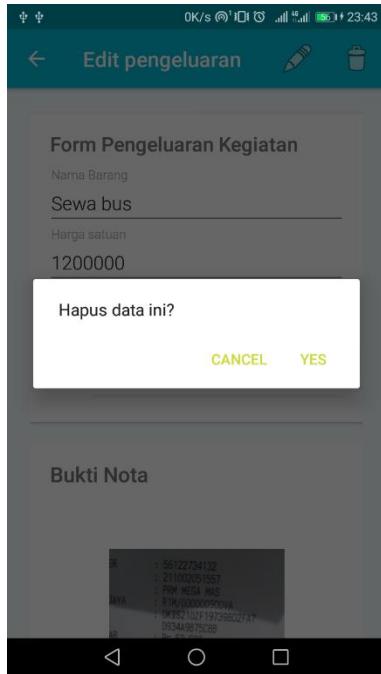
Adapun perbedaan pada *read mode* dan *edit mode*, pada *read mode*, item hanya bisa dilihat tanpa bisa diedit dengan mengatur dan menonaktifkan fungsi edit dan *visibility* pada *edittext*. Sedangkan pada *edit mode*, semua fungsi pada *edittext* di aktifkan kembali dan memungkinkan item bisa diedit. Fungsi pada *read mode* dan *edit mode* juga berlaku pada menu aset dan kas. Adapun kodennya seperti berikut.

```
private void readMode() {
    nama_pengeluaran_ed.setFocusableInTouchMode(false);
    harga_satuan_ed.setFocusableInTouchMode(false);
    jumlah_item_ed.setFocusableInTouchMode(false);
    tgl_pengeluaran_ed.setEnabled(false);
    ChoosePic.setVisibility(View.INVISIBLE);
}

private void editMode() {
    nama_pengeluaran_ed.setFocusableInTouchMode(true);
    harga_satuan_ed.setFocusableInTouchMode(true);
    jumlah_item_ed.setFocusableInTouchMode(true);
    tgl_pengeluaran_ed.setFocusableInTouchMode(true);
    tgl_pengeluaran_ed.setEnabled(true);
    ChoosePic.setVisibility(View.VISIBLE);
}
```

Gambar 4.27. Kode program untuk mengatur fungsi *read mode* dan *edit mode*

Untuk menghapus data, pada halaman form pengeluaran *read mode* terdapat pilihan untuk menghapus data di bagian pojok kanan atas. Setelah mengklik ikon tersebut maka akan muncul konfirmasi untuk menghapus data seperti pada gambar dibawah.



Gambar 4.28. Tampilan pesan konfirmasi untuk menghapus

#### 4.3.9. Implementasi Menu Daftar Kas

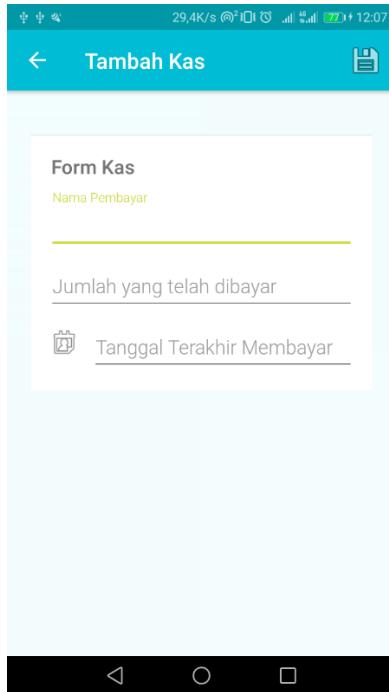
Daftar kas berisi tentang anggota yang sudah membayar kas. Pada daftar kas ini terdapat nama pembayar, jumlah yang sudah dibayarkan dan tanggal terakhir membayar data – data tersebut di *input*kan melalui form kas. Adapun tampilan dari daftar kas adalah sebagai berikut.



Gambar 4.29. Tampilan halaman daftar kas

Untuk menampilkan daftar, sama dengan metode yang digunakan untuk menampilkan daftar kegiatan dan daftar pengeluaran kegiatan.

Ketika pengguna memilih atau mengklik bagian item pada daftar tersebut, akan muncul halaman form kas dengan *read mode*. Pada pojok kanan bawah terdapat tombol untuk menambahkan data kas baru jika pengguna ingin menambahkan data baru. Adapun tampilan dari form kas adalah sebagai berikut.

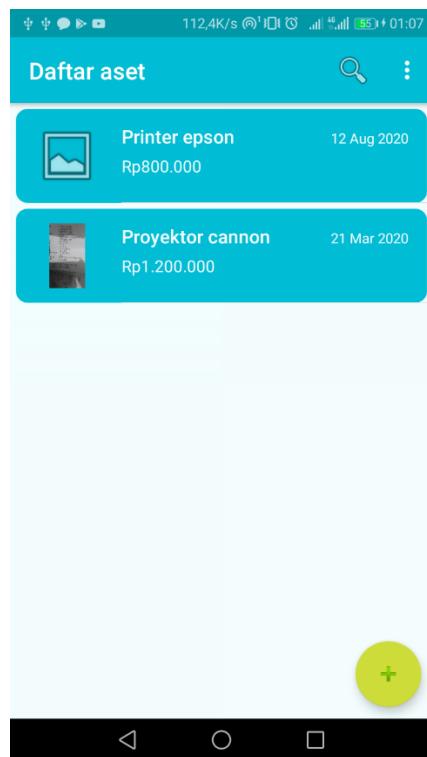


Gambar 4.30. Tampilan halaman form kas

Untuk menambahkan, mengedit, atau menghapus data, menggunakan metode yang sama dengan menu pengeluaran kegiatan sebelumnya.

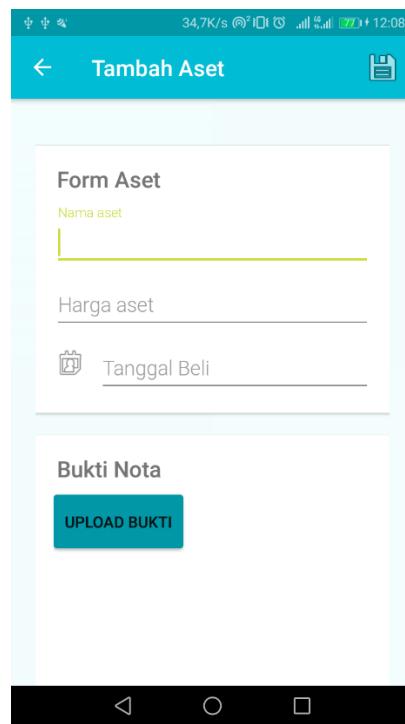
#### 4.3.10. Implementasi Menu Daftar Aset

Halaman ini berisi daftar aset dari organisasi yang dibeli melalui pagudana, yang memiliki jangka waktu pakai yang panjang. Pada daftar aset ini terdapat nama aset, harga aset, tanggal dibeli, serta bukti pembayaran. Dan terdapat tombol menambahkan pada pojok kanan bawah. Adapun tampilan dari daftar aset dan form aset adalah sebagai berikut.



Gambar 4.31. Tampilan halaman daftar aset

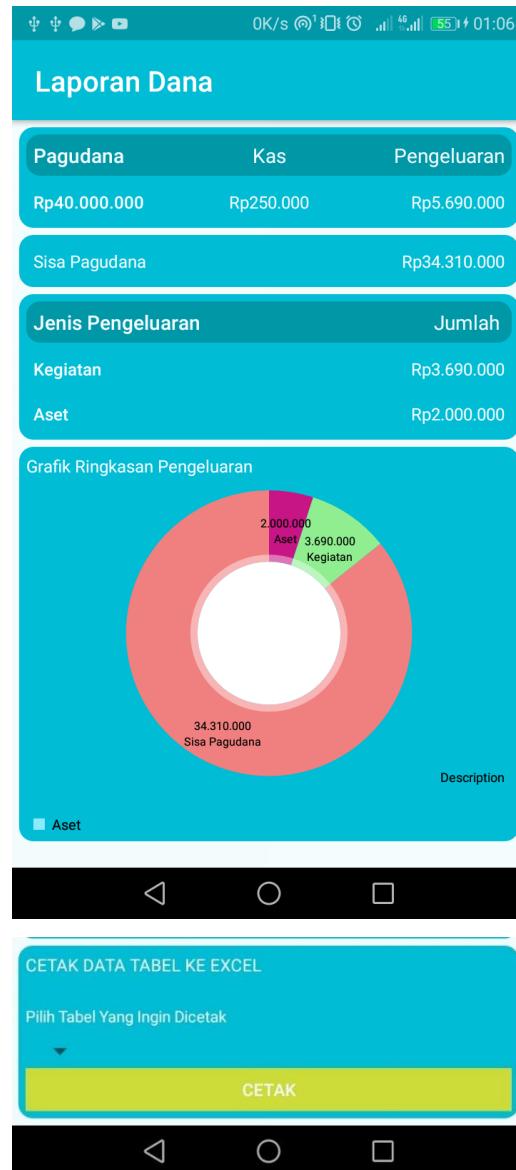
Untuk melihat detail data pada setiap item bisa dengan mengklik pada salah satu item di daftar aset untuk membuka halaman form data aset. Pada form data aset juga bisa melakukan edit dan menghapus data. Tampilan form data aset adalah sebagai berikut.



Gambar 4.32. Tampilan halaman form asset

#### 4.3.11. Implementasi Menu Laporan dana

Halaman ini berisi ringkasan dari penggunaan pagudana, total pengeluaran pada kegiatan maupun aset, jumlah kas yang terkumpul, dan terdapat grafik penggunaan pagudana. Adapun tampilan dari laporan dana ini adalah sebagai berikut.



Gambar 4.33. Tampilan halaman laporan dana

Laporan dana merupakan ringkasan tentang aktivitas keuangan pada UKM yang terdiri dari total kas dan pengeluaran, pagudana, dan sisa pagudana, serta menampilkan grafik dari pengeluaran yang menggunakan pagudana dan cetak data tabel ke excel. Pada grafik tersebut terdapat sisa pagudana, kegiatan, dan aset, yang dimana keseluruhan dari grafik sama dengan jumlah awal pagudana dan untuk bagian cetak tabel berguna untuk mencetak tabel kegiatan, dan aset ke dalam bentuk *file excel*.

#### 4.5. Implementasi Sistem Pada Tipe Pengguna Ketua

Implementasi sistem pada tipe pengguna anggota memiliki beberapa perbedaan dengan tipe bendahara dari segi tampilan dan fungsi seperti pada tampilan halaman *dashboard*(halaman utama), halaman form kegiatan, pengeluaran kegiatan, kas dan aset diubah menjadi halaman detail.

Untuk keseluruhan tampilan dari tipe anggota ini hampir sama dengan bendahara namun dengan hak akses yang berbeda. Tipe ketua hanya bisa melihat informasi data saja tanpa bisa mengelola data didalamnya.

Contoh tampilan pada tipe pengguna ketua dapat dilihat pada tampilan *dashboard* seperti pada gambar.



Gambar 4.34. Tampilan halaman *dashboard* ketua

Dan juga pada halaman form untuk mengelola data pada tipe bendahara diubah menjadi halaman detail seperti halaman detail pengeluaran kegiatan pada gambar dibawah.



Gambar 4.35. Tampilan halaman detail dari tipe pengguna ketua

#### 4.6. Analisis Seluruh Sistem

Aplikasi manajemen keuangan UKM IPTEK merupakan aplikasi yang dibangun menggunakan IDE *Android Studio* dengan menggunakan *database MySQL*. Aplikasi ini berguna untuk merekam segala kegiatan yang berhubungan dengan keuangan di UKM IPTEK. Serta membantu bendahara untuk mengetahui detail pengeluaran dan pemasukan pada satu masa kepemimpinan.

Aplikasi ini juga bisa digunakan oleh ketua untuk bisa melihat informasi pada aplikasi, mengontrol dan mengawasi kinerja bendahara pada aplikasi tersebut dengan menyediakan hak akses yang berbeda pada kedua jenis pengguna tersebut.

Terdapat beberapa menu yang tersedia di aplikasi ini seperti daftar kegiatan, pagudana, daftar kas, daftar aset, dan laporan dana. Setiap menu memiliki fungsi dan tujuan masing – masing sesuai dengan rancangan dan kebutuhan pengguna. Pada menu pagudana, terdapat fitur penambahan bukti terima. Dan pada menu pengeluaran kegiatan, dan aset terdapat fitur untuk menambahkan bukti pembayaran.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan pada uraian – uraian yang telah di jelaskan di bab – bab sebelumnya dapat disimpulkan bahwa:

- a. Aplikasi manajemen keuangan pada UKM IPTEK merupakan aplikasi yang digunakan untuk merekam segala aktivitas keuangan pada UKM IPTEK yang menggunakan *platform* android. Aplikasi ini dibangun menggunakan IDE *Android Studio* versi 3.5.2, dan menggunakan *database* MySQL.
- b. Fitur upload bukti pembayaran digunakan sebagai pelengkap bukti bahwa transaksi tersebut legal dan dapat dipertanggung jawabkan.
- c. Fitur laporan dengan grafik memberikan gambaran ringkas penggunaan pagudana dan memberitahukan informasi mengenai jumlah pengeluaran dan sisa pagudana yang tersedia.

#### **5.2. Saran**

Berdasarkan hasil pada kesimpulan dan uraian pada bab sebelumnya adapun saran yang bisa membantu dalam pengembangan sistem kedepannya adalah.

- a. Dapat mengimplementasikan sistem tidak hanya di lingkup UKM IPTEK saja, tapi bisa digunakan pada organisasi – organisasi lainnya dengan mengevaluasi dan merombak kebutuhan pengguna sehingga bisa di sesuaikan dengan kebutuhan pengguna di organisasi lain.
- b. Dapat menambahkan fitur pencarian dengan *sorting* data berdasarkan waktu atau tanggal.

## **DAFTAR PUSTAKA**

- [1] S. M. Herispon, Buku Ajar Manajemen Keuangan (Financial Management), Pekanbaru: Sekolah Tinggi Ilmu Ekonomi Riau, 2018.
- [2] J. Simarmata, Rekayasa Perangkat Lunak, Yogyakarta: CV. Andi Offset, 2010.
- [3] I. Sommerville, Software Engineering (Rekayasa Perangkat Lunak), Jakarta: Erlangga, 2011.
- [4] N. Khoiri and A. J. Santoso, "Aplikasi Manajemen Keuangan Berbasis Android," *Tugas Akhir*, vol. I, pp. 79-80, 2013.
- [5] M. Nastiti and A. Sunyoto, "Perancangan Aplikasi Manajemen Keuangan Pribadi berbasis Android," *Artikel*, vol. XIII, p. 5, 2012.
- [6] Intelij, "Mengenal Android Studio," Intelij, January 2020. [Online]. Available: <https://developer.android.com/studio/intro/index.html>. [Accessed 28 January 2020].
- [7] M. B. Adam and N. F. Firman, Pemrograman Berorientasi Objek Menggunakan Java, Bandung: Informatika, 2018.
- [8] A. Solichin, Mysql 5: Dari Pemula Hingga Mahir, Jakarta: Universitas Budi Luhur, 2010.

# **LAMPIRAN**

## LAMPIRAN 1

### Tugas dan Peran Bendahara

**GARIS-GARIS BESAR HALUAN KERJA (GBHK)  
UNIT KEGIATAN MAHASISWA (UKM)  
ILMU PENGETAHUAN & TEKNOLOGI (IPTEK)  
POLITEKNIK NEGERI PONTIANAK (POLNEP)  
PERIODE TAHUN 2020**

**BAB X  
BENDAHARA**

**Pasal 14**

1. Posisi, Peran dan Fungsi
  - a. Posisi Bendahara berada dibawah Ketua dan Wakil Ketua.
  - b. Peran Bendahara adalah mengatur segala sesuatu yang berkaitan dengan Finansial UKM IPTEK POLNEP.
  - c. Bendahara terdapat dua fungsi, yaitu sebagai penghimpun dana serta mengatur pengeluaran dan pemasukan dana bagi kegiatan-kegiatan UKM IPTEK POLNEP.
  - d. Peran Bendahara sebagai pengelola keuangan UKM IPTEK POLNEP.
2. Pola dasar Bendahara
  - a. Mencari dan membuat format pencarian dana yang strategis.
  - b. Pengalokasian dana UKM IPTEK POLNEP sesuai dengan kebutuhan masing-masing divisi.
3. Orientasi Program Kerja
  - a. Pemantapan tentang pemahaman laporan penggunaan dana UKM IPTEK POLNEP.
  - b. Membuat laporan keuangan secara transparan.
  - c. Melaporkan hasil keuangan dari setiap kegiatan.

**BAB XI  
DIVISI-DIVISI**

**Pasal 15  
DIVISI KARYA TULIS ILMIAH**

1. Posisi, Peran dan Fungsi
  - a. Divisi Karya Tulis Ilmiah berada dibawah Ketua dan Wakil Ketua.
  - b. Peranan Divisi Karya Tulis Ilmiah sebagai pusat pengembangan Karya Tulis Ilmiah mahasiswa POLNEP.

**LAMPIRAN 2****Kuisisioner Pengujian Aplikasi****Kuisisioner Penelitian**

**Rancang Bangun Aplikasi Manajemen Keuangan Berbasis Android pada Unit Kegiatan  
Mahasiswa Ilmu Pengetahuan dan Teknologi di Politeknik Negeri Pontianak**

Nama : *Silvi..Nuthaliza..*  
Jabatan : *Bendahara.....*

Berilah tanda silang (x) pada nilai yang anda anggap sesuai

Keterangan :

1. Sangat Setuju
2. Setuju
3. Netral
4. Tidak setuju
5. Sangat tidak setuju

No	Pertanyaan	Nilai				
		1	2	3	4	5
1	Apakah aplikasi yang telah dibuat mudah digunakan oleh pengguna?	X				
2	Apakah aplikasi ini sudah sesuai dengan kebutuhan pengguna ?	X				
3	Apakah dengan adanya aplikasi ini dapat membantu kinerja pengelolaan keuangan pada UKM IPTEK ?	X				
4	Apakah tampilan menu pada aplikasi mudah untuk dikenali ?	X				
5	Apakah informasi yang ditampilkan pada aplikasi mudah dimengerti ?		X			
6	Apakah pengguna puas dengan fitur Laporan pada aplikasi ini?	X				
7	Apakah fitur utama di aplikasi ini sudah lengkap berdasarkan kebutuhan pengguna?	X				