**RANCANG BANGUN APLIKASI SIKIBE (SISTEM INFORMASI KOPERASI BE7) PADA KOPERASI SIMPAN PINJAM MENGGUNAKAN ALGORITMA *SEQUENTIAL SEARCH* DI SMK NEGERI 7 BALEENDAH**

**SKRIPSI**

**Karya tulis sebagai syarat untuk memperoleh**

**Gelar Sarjana Komputer dari Fakultas Teknologi Informasi**

**Universitas Bale Bandung**

Disusun oleh:

DIKA SETIA BUDHI

NIM. 301200011



PROGRAM STRATA 1

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

BANDUNG

2024

**LEMBAR PERSETUJUAN PEMBIMBING**

RANCANG BANGUN APLIKASI SIKIBE (SISTEM INFORMASI KOPERASI BE7) PADA KOPERASI SIMPAN PINJAM MENGGUNAKAN ALGORITMA *SEQUENTIAL* *SEARCH* DI SMK NEGERI 7 BALEENDAH

Disusun Oleh :

DIKA SETIA BUDHI

NIM. 301200011

Telah diterima dan disetujui untuk memenuhi pernyataan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2024

Disetujui Oleh

|  |  |
| --- | --- |
| Pembimbing Utama | Pembimbing Pendamping |
| Yudi Herdiana, S.T., M.T.  NIK. 0410808008 | Rustiyana, S.T., M.T.  NIK. 04104808015 |

**LEMBAR PERSETUJUAN PENGUJI**

RANCANG BANGUN APLIKASI SIKIBE (SISTEM INFORMASI KOPERASI BE7) PADA KOPERASI SIMPAN PINJAM MENGGUNAKAN ALGORITMA *SEQUENTIAL SEARCH* DI SMK NEGERI 7 BALEENDAH

Disusun Oleh:

DIKA SETIA BUDHI

NIM. 301200011

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2024

Disetujui Oleh :

|  |  |
| --- | --- |
| Penguji 1 | Penguji 2 |
| Sukiman, S.Tr.Kom., M.Kom.  NIK. 04104821001 | Mohammad Bayu Anggara, S.Kom., M.Kom.  NIK. 04104823002 |

**LEMBAR PERSETUJUAN PROGRAM STUDI**

RANCANG BANGUN APLIKASI SIKIBE (SISTEM INFORMASI KOPERASI BE7) PADA KOPERASI SIMPAN PINJAM MENGGUNAKAN ALGORITMA *SEQUENTIAL SEARCH* DI SMK NEGERI 7 BALEENDAH

Disusun Oleh:

DIKA SETIA BUDHI

NIM. 301200011

SKRIPSI ini telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2024

Disetujui Oleh :

|  |  |
| --- | --- |
| Mengetahui,  Dekan | Mengesahkan,  Ketua Program Studi |
| Yudi Herdiana, S.T., M.T.  NIK. 0410808008 | Yusuf Muharam, S.Kom., M.Kom. NIK. 04104820003 |

**HALAMAN PERNYATAAN**

Yang Bertanda tangan dibawah ini, saya :

Nama : Dika Setia Budhi

NIM : 301200011

Jurusan : Teknik Informatika

Fakultas : Fakultas Teknologi Informasi

Judul : Rancang Bangun Aplikasi SIKIBE (Sistem Informasi Koperasi  
 BE7) Pada Koperasi Simpan Pinjam Menggunakan Algoritma  
 *Sequential Search* Di SMK Negeri 7 Baleendah

Dengan ini penulis menyatakan sebenar-benarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran, dan pemaparan asli dari penulis sendiri, baik untuk naskah laporan maupun kegiatan programming yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, penulis mencuntumkan sumber yang jelas mengikuti tata penulisan karya ilmiah yang seharusnya.

Bandung, Agustus 2024

Materai 10.000

Dika Setia Budhi  
 NIM. 301200011

# ABSTRAK

Koperasi SMK Negeri 7 Baleendah adalah sebuah Lembaga yang ada di sekolah, menyangkut ketua dan para anggota yang mengikutinya. Dengan adanya koperasi dapat menjadi alternatif bagi para anggota dalam peminjaman uang untuk kebutuhan anggota. Namun, dalam pengelolaan data dan proses transaksi simpan pinjam masih dilakukan secara manual dengan buku besar.

Penelitian ini bertujuan untuk merancang dan membangun Aplikasi SIKIBE pada koperasi simpan pinjam Di SMK Negeri 7 Baleendah dengan metode algoritma *Sequential Search*. Dan Penerapan digitalisasi di koperasi SMK Negeri 7 Baleendah dapat mempercepat proses pencarian data, mempermudah pencatatan transaksi tanpa perlu mencatat manual, dan mempercepat pembuatan catatan laporan transaksi yang dapat di unduh. Aplikasi ini diharapkan dapat membantu koperasi SMK Negeri 7 Baleendah dalam meningkatkan efisiensi kinerja petugas koperasi dalam pencarian data anggota dengan mudah, pencatatan transaksi simpan pinjam yang bisa langsung di input dimana saja melalui sistem web, pembuatan catatan laporan koperasi simpan pinjam yang data anggotannya berada di database, dan laporannya akan otomatis tercatat di aplikasi dengan mudah petugas koperasi unduh hasil pencatatannya. Metode *Sequential Search* dipilih karena kesederhanaanya dalam pencarian data.

Hasil penelitian menunjukkan bahwa penerapan algoritma *Sequential Search* ini efektif dalam mempercepat proses pencarian data anggota, terutama dalam konteks basis data dengan jumlah entri yang relatif kecil hingga menengah. Selain itu, algoritma ini juga lebih mudah diimplementasikan dan dipahami dibandingkan dengan algoritma pencarian lainnya. Namun, kelemahan algoritma ini terletak pada efisiensinya yang menurun seiring dengan bertambahnya ukuran basis data. Oleh karena itu, penelitian ini merekomendasikan penggunaan algoritma sequential search untuk koperasi dengan skala data yang terbatas dan menyarankan pengembangan lebih lanjut untuk mengatasi keterbatasan tersebut pada basis data yang lebih besar.

**Kata Kunci**: Koperasi, *Sequential Search*, Aplikasi SIKIBE

# *ABSTRACT*

*The Baleendah 7 State Vocational School Cooperative is an institution within the school, involving the chairman and the members who follow him. The existence of cooperatives can be an alternative for members in borrowing money for members' needs. However, data management and savings and loan transaction processes are still done manually using ledgers.*

*This research aims to design and build the SIKIBE application for the savings and loan cooperative at SMK Negeri 7 Baleendah using the Sequential Search algorithm method. And the implementation of digitalization in the SMK Negeri 7 Baleendah cooperative can speed up the data search process, make it easier to record transactions without the need for manual recording, and speed up the creation of downloadable transaction report notes. This application is expected to help the SMK Negeri 7 Baleendah cooperative in improving the efficiency of the performance of cooperative officers in searching for member data easily, recording savings and loan transactions which can be directly input anywhere via the web system, making records of savings and loan cooperative reports whose member data is in the database , and the report will be automatically recorded in the application, with cooperative officers easily downloading the recording results. The Sequential Search method was chosen because of its simplicity in searching data.*

*The research results show that the application of the Sequential Search algorithm is effective in speeding up the member data search process, especially in the context of databases with a relatively small to medium number of entries. Apart from that, this algorithm is also easier to implement and understand compared to other search algorithms. However, the weakness of this algorithm lies in its efficiency which decreases as the database size increases. Therefore, this research recommends the use of sequential search algorithms for cooperatives with limited data scales and suggests further development to overcome these limitations on larger databases.*

***Keywords***: *Cooperative, Sequential Search, SIKIBE Application*

# KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang senantiasa melimpahkan rahmat, taufik dan karunia-nya, sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul “Rancang Bangun Aplikasi Sibike pada Koperasi Simpan Pinjam Menggunakan Algoritma *Sequential Search* di SMK Negeri 7 Baleendah”ini dengan baik.

Dalam penyusunan-nya penulis memperoleh banyak bantuan dari berbagai pihak, maka dari itu penulis mengucapkan terimakasih yang sebanyak-banyak nya kepada.

1. Allah SWT yang telah memberikan kemudahan serta kelancaran sehingga penulis dapat menjalani proses demi proses pelaksanakaan Tugas Akhir ini.
2. Bapak Yudi Herdiana, S.T.,M.T., selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.
3. Bapak Yusuf Muharam, S.Kom., M.Kom., selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.
4. Dedi Junaedi, S.E selaku Kepala Sekolah SMK Negeri 7 Baleendah
5. Semua pihak yang telah membantu dan memberikan dukungan kepada penulis untuk menyelesaikan proposal ini.

Penyusun menyadari bahwa masih banyak kekurangan dalam laporan skripsi dan segala kritik dan saran yang membangun akan penyusun terima dengan baik. Akhir kata semoga laporan skripsi bisa diterima dan bermanfaat bagi kita semua.

Bandung, Maret 2024

Dika Setia Budhi  
NIM. 301200011

# DAFTAR ISI

[ABSTRAK vi](#_Toc175049786)

[*ABSTRACT* vii](#_Toc175049787)

[KATA PENGANTAR viii](#_Toc175049788)

[DAFTAR ISI ix](#_Toc175049789)

[DAFTAR TABEL xii](#_Toc175049790)

[DAFTAR GAMBAR xiii](#_Toc175049791)  
DAFTAR LAMPIRAN xv

[BAB I PENDAHULUAN 1](#_Toc175049792)

[1.1 Latar Belakang 1](#_Toc175049793)

[1.2 Rumusan Masalah 3](#_Toc175049794)

[1.3 Batasan Masalah 3](#_Toc175049795)

[1.4 Tujuan Penelitian 3](#_Toc175049796)

[1.5 Metode Penelitian 4](#_Toc175049797)

[1.5.1 Metode Pengumpulan Data 4](#_Toc175049798)

[1.5.2 Metode Pengembangan Sistem 4](#_Toc175049799)

[1.6 Sistematika Penulisan 5](#_Toc175049800)

[BAB II TINJAUAN PUSTAKA 6](#_Toc175049807)

[2.1 Landasan Teori 6](#_Toc175049808)

[2.2 Dasar Teori 9](#_Toc175049809)

[2.2.1 Website 9](#_Toc175049810)

[2.2.2 Koperasi Simpan Pinjam 10](#_Toc175049811)

[2.2.3 *Sequential Search* 11](#_Toc175049812)

[2.2.4 *Database* 14](#_Toc175049813)

[2.2.5 Perancangan Sistem 14](#_Toc175049814)

[2.2.6 *MYSQL* 20](#_Toc175049815)

[2.2.7 *PHP* 22](#_Toc175049816)

[2.2.8 Metode *Waterfall* 24](#_Toc175049817)

[2.2.9 XAMPP 26](#_Toc175049818)

[2.2.10 *Laravel* 28](#_Toc175049819)

[2.2.11 *Visual Studio Code* 30](#_Toc175049820)

[2.2.12 Figma 32](#_Toc175049821)

[BAB III ANALISIS DAN PERANCANGAN 33](#_Toc175049822)

[3.1 Kerangka Pikir 33](#_Toc175049823)

[3.2 Deskripsi 34](#_Toc175049824)

[3.2.1 Identifikasi Masalah 34](#_Toc175049825)

[3.2.2 Pengumpulan Data 34](#_Toc175049826)

[3.3 Analisis Kebutuhan 35](#_Toc175049827)

[3.3.2 Desain 36](#_Toc175049828)

[3.3.3 Implementasi 38](#_Toc175049829)

[3.3.4 Pengujian 38](#_Toc175049830)

[3.3.5 Pelaporan 39](#_Toc175049831)

[BAB IV ANALISIS DAN PERANCANGAN 40](#_Toc175049832)

[4.1 Analisis 40](#_Toc175049833)

[4.1.1 Analisis Masalah 40](#_Toc175049834)

[4.1.2 Analisis Software 40](#_Toc175049835)

[4.1.3 Analisis Pengguna 41](#_Toc175049836)

[4.1.4 *User interface* 41](#_Toc175049837)

[4.1.5 Fitur-fitur 42](#_Toc175049838)

[4.1.6 Analisis Data 43](#_Toc175049839)

[4.1.7 Analisis Biaya 44](#_Toc175049840)

[4.2 Perancangan 44](#_Toc175049841)

[4.2.1 *Unified Modeling Language* (UML) 44](#_Toc175049842)

[4.2.2 Desain sistem 69](#_Toc175049843)

[BAB V IMPLEMENTASI DAN PENGUJIAN 82](#_Toc175049844)

[5.1 Implementasi 82](#_Toc175049845)

[5.1.1 *Listing* Program 82](#_Toc175049846)

[5.1.2 Implementasi Sistem 92](#_Toc175049847)

[5.1.3 Spesifikasi Sistem 92](#_Toc175049848)

[5.1.4 Installasi Sistem 93](#_Toc175049849)

[5.1.5 Menjalankan Sistem 94](#_Toc175049850)

[5.2 Pengujian 106](#_Toc175049851)

[5.2.1 Pengujian *Alpha* 106](#_Toc175049852)

[5.2.2 Pengujian *Beta* 108](#_Toc175049853)

[BAB VI KESIMPULAN 98](#_Toc175049854)

[6.1 Kesimpulan 98](#_Toc175049855)

[6.2 Saran 99](#_Toc175049856)

[DAFTAR PUSTAKA 100](#_Toc175049857)

[LAMPIRAN 102](#_Toc175049858)

# DAFTAR TABEL

[Tabel 2.1 Acuan Penelitian 6](#_Toc175049860)

[Tabel 2.2 Simbol *Use Case* Diagram 16](#_Toc175049861)

[Tabel 2.3 Simbol *Class* Diagram 17](#_Toc175049862)

[Tabel 2.4 Simbol *Activity* Diagram 19](#_Toc175049863)

[Tabel 4.1 Analisis Biaya 44](#_Toc175049864)

[Tabel 4.2 Deskripsi Aktor 46](#_Toc175049865)

[Tabel 4.3 Tabel *Users* 55](#_Toc175049866)

[Tabel 4.4 Tabel *Approvals* 56](#_Toc175049867)

[Tabel 4.5 Tabel *Customers* 57](#_Toc175049868)

[Tabel 4.6 Tabel *Deposits* 58](#_Toc175049869)

[Tabel 4.7 Tabel *Histories* 60](#_Toc175049870)

[Tabel 4.8 Tabel *Invoices* 61](#_Toc175049871)

[Tabel 4.9 Tabel *Loans* 62](#_Toc175049872)

[Tabel 4.10 Tabel *Loan\_Instalments* 63](#_Toc175049873)

[Tabel 4.11 Tabel *Payments* 64](#_Toc175049874)

[Tabel 4.12 Tabel *Payment\_Methods* 66](#_Toc175049875)

[Tabel 4.13 Tabel *Transactions* 67](#_Toc175049876)

[Tabel 5.1 Spesifikasi Perangkat Keras 93](#_Toc175049877)

[Tabel 5.2 Spesifikasi Perangkat Lunak 93](#_Toc175049878)

[Tabel 5.3 Hasil Pengujian Aplikasi Dengan Metode *Blackbox* 107](#_Toc175049879)

[Tabel 5.4 Pertanyaan Pengujian *Beta* 109](#_Toc175049880)

[Tabel 5.5 Pengujian Validitas 111](#_Toc175049881)

# DAFTAR GAMBAR

Gambar 2.1 Arsitektur *Laravel* 29

[Gambar 3.1 Kerangka Pikir 33](#_Toc175049882)

[Gambar 4.1 *Usecase* Diagram Admin 45](#_Toc175049883)

[Gambar 4.2 *Usecase* Diagram Anggota 45](#_Toc175049884)

[Gambar 4.3 *Activity* Diagram *Login* Admin 47](#_Toc175049885)

[Gambar 4.4 *Activity* Diagram *Login* *User* 48](#_Toc175049886)

[Gambar 4.5 *Activity* Diagram *Register* 49](#_Toc175049887)

[Gambar 4.6 *Activity* Diagram Lupa *Password* 50](#_Toc175049888)

[Gambar 4.7 *Activity* Diagram Kelola Anggota 51](#_Toc175049889)

[Gambar 4.8 *Activity* Diagram Kelola Simpanan 52](#_Toc175049890)

[Gambar 4.9 *Activity* Diagram Kelola Pinjaman 53](#_Toc175049891)

[Gambar 4.10 *Activity* Diagram *Edit Profile* 54](#_Toc175049892)

[Gambar 4.11 *Class* Diagram Struktur Tabel 55](#_Toc175049893)

[Gambar 4.12 *User Interface* Halaman *Login Admin* 69](#_Toc175049894)  
Gambar 4.13 *User Interface* Halaman Login *User* 70  
Gambar 4.14 *User Interface* Halaman Beranda 70  
Gambar 4.15 *User Interface* Halaman Pengguna 71  
Gambar 4.16 *User Interface* Tambah Pengguna 71  
Gambar 4.17 *User Interface* Ubah Pengguna 72

[Gambar 4.18 *User Interface* Halaman Hapus Pengguna 72](#_Toc175049895)

[Gambar 4.19 *User Interface* Halaman Anggota 73](#_Toc175049896)

[Gambar 4.20 *User Interface* Halaman Tambah Anggota 73](#_Toc175049897)

[Gambar 4.21 *User Interface* Halaman Ubah Pengguna 74](#_Toc175049898)

[Gambar 4.22 *User Interface* Halaman Hapus Anggota 74](#_Toc175049899)

[Gambar 4.23 *User Interface* Halaman Simpanan 75](#_Toc175049900)

[Gambar 4.24 *User Interface* Halaman Buat Simpanan 75](#_Toc175049901)

[Gambar 4.25 *User Interface* Halaman Batalkan Simpanan 76](#_Toc175049902)

[Gambar 4.26 *User Interface* Halaman Pinjaman 76](#_Toc175049903)

[Gambar 4.27 *User Interface* Halaman Buat Pinjam 77](#_Toc175049904)

[Gambar 4.28 *User Interface* Halaman Batalkan Pinjam 77](#_Toc175049905)  
Gambar 4.29 Halaman Beranda User 78  
Gambar 4.30 Halaman Simpanan User 78  
Gambar 4.31 Halaman Pinjaman User 79

[Gambar 5.1 Halaman *Login* Admin 95](#_Toc175049906)  
Gambar 5.2 Halaman Login *User* 95  
Gambar 5.3 Halaman Register 96  
Gambar 5.4 Halaman Beranda 96  
Gambar 5.5 Halaman Pengguna 97  
Gambar 5.6 Halaman Tambah Pengguna 97  
Gambar 5.7 Halaman Ubah Pengguna 98  
Gambar 5.8 Halaman Hapus Pengguna 98  
Gambar 5.9 Halaman Anggota 99  
Gambar 5.10 Halaman Tambah Anggota 99  
Gambar 5.11 Halaman Ubah Anggota 100  
Gambar 5.12 Halaman Hapus Anggota 100

[Gambar 5.13 Halaman Simpanan 100](#_Toc175049907)   
Gambar 5.14 Halaman Tambah Simpanan 101  
Gambar 5.15 Halaman Ubah Simpanan 102  
Gambar 5.16 Halaman Hapus Simpanan 102  
Gambar 5.17 Halaman Pinjaman 103  
Gambar 5.18 Halaman Tambah Pinjaman 103  
Gambar 5.19 Halaman Ubah Pinjaman 104  
Gambar 5.20 Halaman Hapus Pinjaman 104  
Gambar 5.21 Halaman Beranda *User* 105  
Gambar 5.22 Halaman Simpanan *User* 105  
Gambar 5.23 Halaman Pinjaman *User* 106  
Gambar 5.24 Pengujian Realibilitas 106

**DAFTAR LAMPIRAN**

LAMPIRAN 1: Surat Penelitian 102

LAMPIRAN 2: Surat Balasan Penelitian 103

LAMPIRAN 3: Dokumentasi 104

LAMPIRAN 4: Transkip Wawancara 105

LAMPIRAN 5: Data Pengujian 107

LAMPIRAN 6: ToR *(Term Of Reference)* 123

LAMPIRAN 7: Listing Program 124

# BAB I PENDAHULUAN

## Latar Belakang

Keberadaan koperasi di lingkungan masyarakat di Indonesia bisa mendukung serta memudahkan beban perekonomian yang dapat menekan masyarakat indonesia tersebut. Koperasi berencana memajukan kemakmuran anggotannya, sebab itu koperasi mulai dimasyarakatkan ke beragam penjuru tanah air indonesia. Termasuk juga ke dalam lingkungan pendidikan, lingkungan para peserta didik, serta ke lingkungan mahasiswa yang dimulai dari mereka sudah belajar tentang berlembaga dalam koperasi. Dimana koperasi yang ada di dalam lingkungan pelajar disebut dengan koperasi sekolah. (Sudarsono, 2010)

Koperasi SMK Negeri 7 Baleendah berdiri pada tanggal 20 februari 2015 dengan legalitas badan hukum 11/BH/518-KOP/II/2015. Ibu Tetet Sopiah, S.Pd selaku kepala koperasi dan Anggota yang mengikuti meliputi tenaga pendidik, tenaga kependidikan dan masyarakat umum yang berjumlah 110 anggota. Pada koperasi SMK Negeri 7 Baleendah yang berlokasi di Jalan Siliwangi KM.15 RT 04 RW 17 Desa Manggahang Kecamatan Baleendah Kabupaten bandung jawa barat terdapat simpanan pokok, simpanan wajib, simpanan sukarela dan pinjaman. Setiap konsumen diwajibkan setiap bulan nya membayar simpanan wajib sebesar seratus ribu dan pihak koperasi juga telah menyediakan peminjaman uang kepada konsumen anggota.

Setelah melakukan observasi dan wawancara dengan kepala koperasi permasalahan yang ada dan di alami koperasi SMK Negeri 7 Baleendah proses pencarian data konsumen masih manual, pencatatan transaksi konsumen masih menulis di buku besar, dan pencatatan pelaporan transaksi simpan pinjam juga masih menggunakan buku besar dibantu dengan *Microsoft word*. Dengan adanya permasalahan yang ada di SMK Negeri 7 Baleendah ini mempunyai dampak signifikan terhadap bagian koperasi menyebabkan lamanya pengambilan keputusan, dan menyulitkan petugas koperasi dalam pencarian data konsumen sehingga menghasilkan informasi kepada konsumen yang tidak akurat, dan pencatatan pelaporan transaksi konsumen yang membutuhkan waktu cukup lama. Selain itu, laporan yang disajikan sangat lambat karena masih menggunakan arsip yang tersimpan didalam buku penyimpanan, dan harus mencarinya satu persatu. Dengan jumlah anggota yang cukup banyak dan petugas koperasi yang sangat terbatas, maka akan menimbulkan lambannya kinerja koperasi SMK Negeri 7 Baleendah. Koperasi belum memanfaatkan digitalisasi atau aplikasi khusus yang dapat memudahkan petugas koperasi, selain itu belum juga penerapan digitalisasi aplikasi dalam pembuatan dokumen sehingga semua data arsip konsumen anggota masih dalam bentuk kertas yang di catat oleh bagian koperasi secara manual.

Penelitian oleh Lasriana dan Aris Gunaryati (2022) dalam "Sistem Informasi Apotek Berbasis *Web* Menggunakan Algoritma *Sequential Search* dan *Selection Sort*" menunjukkan bahwa Algoritma *Sequential Search* efektif untuk pencarian data dan informasi akurat. Hidayat (2021) dalam "Penerapan Algoritma *Sequential Search* pada Sistem Informasi Perpustakaan" juga menemukan algoritma ini meningkatkan efisiensi pengelolaan informasi perpustakaan. Suryani (2020) dalam "Implementasi Algoritma *Sequential Search* pada Sistem Informasi Kepegawaian" menegaskan manfaatnya dalam pencarian data kepegawaian dan kinerja sistem. Penelitian saya menawarkan beberapa kelebihan yaitu fokus khusus pada koperasi simpan pinjam sehingga solusi lebih relevan, penggunaan eksklusif algoritma *Sequential Search* untuk optimalisasi, aspek digitalisasi yang mencakup pengelolaan data anggota, transaksi, dan layanan konsumen, serta aplikasi SIKIBE yang diharapkan mempermudah tugas dan meningkatkan efisiensi operasional koperasi.

Berdasarkan hasil observasi dan wawancara dengan kepala koperasi, berencana mengimplementasikan metode Algoritma *Sequential Search* untuk meningkatkan kinerja koperasi SMK Negeri 7 Baleendah. Metode ini akan memudahkan petugas koperasi dalam pencarian data koperasi dan mendukung digitalisasi, sehingga membantu pengelolaan koperasi serta layanan kepada konsumen. Motivasi ini mendasari penulisan skripsi berjudul “RANCANG BANGUN APLIKASI SIKIBE (SISTEM INFORMASI KOPERASI BE7) PADA KOPERASI SIMPAN PINJAM MENGGUNAKAN METODE ALGORITMA *SEQUENTIAL SEARCH*,” dengan harapan aplikasi tersebut dapat mengatasi permasalahan dan mempermudah tugas pokok serta fungsi petugas koperasi.

## Rumusan Masalah

Berdasarkan uraian pada latar belakang, maka dapat dirumuskan bahwa masalah penelitian yaitu:

1. Bagaimana penerapan algoritma *Sequential Search* dalam proses pencarian data anggota koperasi simpan pinjam di SMK Negeri 7 Baleendah?
2. Bagaimana penerapan algoritma *Sequential Search* dalam proses pencatatan transaksi di koperasi SMK Negeri 7 Baleendah?
3. Bagaimana cara mempercepat pembuatan laporan koperasi simpan pinjam agar tidak membutuhkan waktu yang lama?

## Batasan Masalah

Adapun batasan masalah dalam pelaksanaan penelitian dan pembuatan sistem informasi yakni:

1. Objek penelitian dilakukan di SMKN 7 Baleendah.
2. Aplikasi yang dibuat berbasis *Website.*
3. Data yang digunakan adalah data dari SMKN 7 Baleendah.
4. Algoritma yang digunakan untuk pencarian data adalah sequential search.
5. Aktor yang terlibat dalam sistem meliputi admin, petugas koperasi, dan anggota koperasi.
6. Sistem menggunakan fitur otentikasi, manajemen anggota, dan transaksi.
7. Sistem yang akan dibangun menggunakan framework laravel, Bahasa pemograman *PHP* dan *MySqL* sebagai database serta *Visual Studio Code* sebagai text editor.

## Tujuan Penelitian

Berdasarkan rumusan masalah sebelumnya, maka tujuan penelitian yang ingin dicapai adalah:

1. Penerapan algoritma *Sequential Search* dalam proses pencarian data anggota koperasi simpan pinjam SMK Negeri 7 Baleendah.
2. Penerapan algoritma *Sequential Search* dalam proses pencatatan transaksi di koperasi simpan pinjam SMK Negeri 7 Baleendah.
3. Mempercepat pembuatan laporan koperasi simpan pinjam SMK Negeri 7 Baleendah agar tidak membutuhkan waktu yang lama.

## Metode Penelitian

### Metode Pengumpulan Data

Pengumpulan data dilakukan melalui observasi langsung, wawancara dengan pihak terkait, dan studi literatur untuk mendukung pengembangan sistem.

Observasi   
 Yaitu dilakukan dengan cara mengamati langsung sistem informasi koperasi yang berjalan di SMKN 7 Baleendah dan penulis mendapatkan informasi dari petugas koperasi agar sistem yang sedang berjalan ini bisa di perbaiki dengan menerapkan digitalisas

Wawancara (Interview)

Yaitu dilakukan dengan proses tanya jawab dengan kepala koperasi SMKN 7 Baleendah mengenai sistem koperasi simpan pinjam pada yang sedang berjalan saat ini. Selain itu, kepala koperasi ibu tetet sopiah mengetahui kesulitan dan kebutuhan apa saja yang diperlukan dalam sistem yang akan dibangun oleh penulis.

Studi Pustaka

Yaitu dilakukan dengan mengumpulkan data dari beberapa buku, jurnal, dan sumber lainnya yang bersangkutan dengan aplikasi koperasi simpan pinjam menggunakan algoritma *sequential search.*

### Metode Pengembangan Sistem

Adapun metode penelitian yang digunakana adalah RAD (Rapid Application Development), pendekatan pengembangan perangkat lunak yang berfokus pada pengembangan iteratif dan kolaboratif antara pengembang perangkat lunak dan pemangku kepentingan proyek. Metode ini sangat sesuai untuk proyek-proyek yang memiliki batasan waktu yang ketat dan memerlukan iterasi cepat untuk mendapatkan umpan balik dari pengguna.

## Sistematika Penulisan

Sistematika penulisan bertujuan untuk menyajikan informasi secara terstruktur dan mudah dipahami oleh pembaca. Adapun sistematika penulisan yang digunakan adalah sebagai berikut:

### BAB I PENDAHULUAN

Bab ini merupakan bagian utama yang mengemukakan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan laporan hasil penelitian.

### BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang landasan teori dan dasar teori pendukung dalam penelitian. Tinjauan Pustaka ini bersumber dari beberapa sumber seperti buku, jurnal, dan website.

### BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metodologi penelitian yang dipakai dan kerangka penelitian atau tahap-tahap penulis dalam melakukan penelitian di SMK Negeri 7 Baleendah.

### BAB IV ANALISIS, PERANCANGAN DAN HASIL

Bab ini berisi tentang analisis dan perancangan perangkat lunak, perancangan database, perancangan antarmuka serta penjelasan tentang perancangan perangkat lunak yang akan dibangun.

### BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi penyajian tahap pengembangan aplikasi yang akan dijelaskan tiap langkahnya sertca contoh tampilan dari setiap aplikasi.

### BAB VI PENUTUP

Bab ini berisi kesimpulan dan penyajian tahap pengembangan yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

# BAB II TINJAUAN PUSTAKA

## Landasan Teori

Landasan teori ini berisi ringkasan berdasarkan jurnal yang berkaitan dengan penelitian yang dilakukan dan pengetahuan yang diterima selama perkuliahan untuk pengembangan perangkat lunak aplikasi. Pengetahuan digunakan meliputi:

1. Teori Algoritma 1 dan Algoritma 2.
2. Basis Data.
3. Teori Interaksi Manusia dan Komputer.
4. Teori Pemrograman Internet.
5. Rekayasa Perangkat Lunak

Pada penelitian ini, penyusun mempelajari penelitian terdahulu sebagai acuan dalam penelitian sebagai berikut:

Tabel 2.1 Acuan Penelitian

|  |  |  |
| --- | --- | --- |
| **No.** | **Acuan Jurnal Penelitian** | **Hasil** |
| 1 | Judul: Sistem Informasi Manajemen Koperasi Siswa Smkn 3 Mataram Berbasis *Website* | * **Kesimpulan: S**istem informasi manajemen koperasi siswa SMKN 3 Mataram berhasil membuat transaksi di koperasi siswa SMKN 3 Mataram menjadi lebih efisien. * **Kesamaan:** Sistem yang dibuat berbasis *website* * **Pembeda:** Menggunakan metode *Rapid Application Development* (RAD) |
| Nama Penulis: Muhammad Rizqi Aditia, Arik Aranta, Puji Astuti |
| Jurnal: JBegaTI |
| Tahun: 2022 |
| 2 | Judul: Klasifikasi Kelayakan Pinjaman Pada Koperasi Karyawan Menggunakan Metode *Naïve Bayes Classifier* Berbasis *Web* | * **Kesimpulan:** Sistem yang dibuat dapat mengklasifikasikan pengajuan pinjaman para karyawan dengan catatan harus memenuhi syarat yang telah ditentukan sebelumnya. * **Kesamaan:** Sistem yang dibuat berbasis *website* * **Pembeda:** Algoritma pencarian menggunakan algoritma *sequential search* |
| Nama Penulis: Iwan Riswanto, Riffa Haviani Laluma |
| Jurnal: Jurnal Infotronik |
| Tahun: 2020 |
| 3 | Judul: Sistem Informasi Simpan Pinjam Pada Koperasi Smk Yuppentek 2 Curug Tangerang Dengan Metodologi Berorientasi Obyek | * **Kesimpulan:** Dengan adanya sistem yang terkomputerisasi dapat mempermudah Bendahara Koperasi mendapatkan laporan potongan simpanan, daftar pinjaman, pengunduran diri, potongan pinjaman, penarikan simpanan suka rela dan rekapitulasi simpanan. * **Kesamaan:** Membuat Sistem Informasi Simpan Pinjam * **Pembeda:** Sistem yang dibuat berbasis *website* |
| Nama Penulis: Riyan Bagus Prastian, Lusi Fajarita |
| Jurnal: Jurnal IDEALIS |
| Tahun: 2019 |
| 4 | Judul: Sistem Informasi Koperasi Simpan Pinjam Studi Kasus Koperasi Kodanua Serang | * **Kesimpulan:** Sistem yang dirancang telah sesuai dengan kebutuhan dan keinginan pengguna, dalam hal ini pengurus koperasi Kodanua. * **Kesamaan:** Membuat Sistem Informasi Simpan Pinjam * **Pembeda:** Menerapkan algoritma pada sistem yang dibuat |
| Nama Penulis: Tuti Handayani, Arief Herdiansah, Nofitri Hariyani, Tomi Nugroho |
| Jurnal: Jurnal Teknik Informatika |
| Tahun: 2020 |
| 5 | Judul: Klasifikasi Penentuan Kelayakan Pemberian Kredit Menggunakan Metode *Naive Bayes Classifier* (Kasus: Koperasi Simpan Pinjam Artha Segara) | * **Kesimpulan:** Kelayakan pemberian kredit kepada nasabah Koperasi Simpan Pinjam Artha Segara, pengujian data sebanyak 1.200 dataset dari tahun 2018 sampai dengan tahun 2020 dengan 15 atribut yang diuji. Dimana 1.200 dataset dibagi menjadi 2 jenis data yaitu data latih (data *training*) dan data uji (data *testing*) dengan perbandingan 70%:30%. Data training sebesar 860 data dan data testing sebesar 360 data. * **Kesamaan:** Mengangkat topik penelitian yang sama * **Pembeda:** Aplikasi yang dibuat berbasis *website* |
| Nama Penulis: Ni Komang Ayu Suarpurningsih, Nengah Widya Utami, Ni Made Estiyanti |
| Jurnal: Jurnal Sains Komputer & Informatika |
| Tahun: 2022 |

**2.1.2**. Kesimpulan tabel penelitian

Penerapan sistem informasi berbasis komputer pada koperasi simpan pinjam telah membawa dalam pengelolaan data dan transaksi. Penggunaan metode *Sequential Search* yang efisien memungkinkan petugas koperasi untuk dengan cepat menemukan data anggota yang dibutuhkan, baik itu data simpanan, pinjaman, maupun informasi pribadi lainnya. Sementara itu, penerapan metode Waterfall dalam pengembangan sistem telah memastikan bahwa setiap tahap pengembangan dilakukan secara terstruktur dan sesuai dengan kebutuhan. Hasilnya, sistem ini tidak hanya mampu mencatat transaksi dengan lebih akurat, tetapi juga menghasilkan laporan keuangan yang lebih cepat dan terpercaya. Petugas koperasi dapat lebih fokus pada memberikan pelayanan yang prima kepada anggota, sehingga meningkatkan kepercayaan dan kepuasan anggota terhadap koperasi.

Dari ketiga jurnal, perbandingan dan kelebihan penelitian yaitu dari hasil yang telah di lakukan sistem yang dibuat sangat memudahkan dalam pencarian data anggota koperasi dengan cepat menggunakan metode *Sequential Search* dimana petugas koperasi lebih efisien menggunakan sistem ini dalam melakukan pekerjaannya, meningkatkan kinerja petugas koperasi, serta sistem ini sangat memudahkan dalam melakukan transaksi simpan pinjam para anggotannya yang dimana petugas koperasi dapat melakukan pekerjaannya dimana saja.

## Dasar Teori

Pada pembuatan sistem informasi ini, penyusun mempunyai bahan dan alat untuk pembuatan sistem informasi, maka penyusun mempelajari teori-teori yang ada sebagai acuan dalam implementasi sistem informasi yang akan penulis buat yaitu sebagai berikut:

### Website

*Website* merupakan sebuah kumpulan halaman-halaman *web* beserta *file*-*file* pendukungnya, seperti *file* gambar, *video*, dan *file* digital lainnya yang disimpan pada sebuah *web* server yang umumnya dapat diakses melalui internet. Atau dengan kata lain, *website* adalah sekumpulan *folder* dan *file* yang mengandung banyak perintah dan fungsi fungsi tertentu, seperti fungsi tampilan, fungsi menangani penyimpanan data (Departemen et al., 2023).

*World Wide Web* (www), juga dikenal sebagai web atau situs *web*, adalah *file* teks yang berisi tag pemformatan *Hypertext Markup Language* (HTML), tautan ke *file*, atau halaman *web* lainya, *File* ini disimpan di *server* situs *web* dan dapat dilihat oleh komputer lain yang terhubung ke *server*, baik melalui internet atau melalui jaringan area lokal (intranet). Dengan demikian dari pengertian diatas dapat disimpulkan bahwa *website* merupakan kumpulan halaman-halaman *web* dan *file-file* yang disimpan pada sebuah *web sercer*. Secara sederhana *website* dapat dianggap sebagai sekumpulan *folder* dan *file* yang memiliki berbagai perintah dan fungsi tertentu. (Arhandi et al., 2022)

### Koperasi Simpan Pinjam

Koperasi simpan pinjam yaitu suatu usaha yang bergerak dalam bidang pengumpulan modal dengan cara tabungan dan pinjaman dari anggotanya. Tujuan pengumpulan dana tersebut yaitu untuk memudahkan para anggotanya agar mendapatkan modal usaha yang pruduktif dan menambah kesejahteraan. (Isa & Hartawan, 2020)

Berdasarkan definisi tersebut, dapat disimpulkan bahwa koperasi simpan pinjam adalah suatu lembaga usaha yang bertujuan untuk mengumpulkan modal dari para anggotanya melalui tabungan dan pemberian pinjaman. Tujuan utama dari pengumpulan dana tersebut adalah untuk memberikan kemudahan kepada anggota dalam mendapatkan modal usaha yang produktif serta meningkatkan kesejahteraan mereka.

Berikut adalah beberapa poin penting tentang koperasi simpan pinjam :

Struktur Anggota

Koperasi simpan pinjam didirikan oleh sekelompok orang dengan kebutuhan dan kepentingan yang sama. Anggota biasanya terdiri dari masyarakat lokal atau kelompok tertentu yang ingin saling membantu dalam hal keuangan.

Simpanan Setiap anggota diharapkan untuk secara rutin menabung dalam koperasi. Simpanan ini tidak hanya memberikan keuntungan bagi anggota, tetapi juga menyediakan dana yang dapat dipinjamkan kepada anggota lain.

Pinjaman

Anggota yang membutuhkan dana dapat mengajukan pinjaman kepada koperasi. Pinjaman ini dapat digunakan untuk berbagai keperluan seperti pendidikan, usaha kecil, atau keperluan lainnya. Syarat dan ketentuan pinjaman biasanya ditetapkan oleh koperasi berdasarkan aturan yang telah disepakati bersama.

Bunga

Koperasi biasanya mengenakan bunga pada pinjaman yang diberikan kepada anggotanya. Namun, bunga yang dikenakan cenderung lebih rendah dibandingkan dengan lembaga keuangan konvensional. Keuntungan dari bunga tersebut seringkali digunakan untuk operasional koperasi dan pembagian sisa hasil usaha kepada anggota.

Regulasi

Koperasi simpan pinjam biasanya diatur oleh undang-undang dan regulasi pemerintah untuk memastikan transparansi, keamanan, dan keberlanjutan operasionalnya

### *Sequential Search*

*Sequential Search* adalah cara untuk pencarian data dalam satu array, data yang akan dicari dalam semua elemen dari awal sampai akhir. Pencarian ini merupakan fungsi untuk mengambil data pada suatu kumpulan data. (Wahyuni et al., 2022). *Sequential Search* metode pencarian berurutan, pencarian data dapat pada data yang terurut dan tidak terurut. Keuntungan dalam algoritma ini adalah jika data yang kita cari berada pada awal data maka langsung akan ditemukan.

Terdapat 2 kemungkinan yang akan terjadi dalam waktu pencarian data *Sequantial Search*, diantaranya yaitu :

*Best Case*

*Best case* kemungkinan terbaik akan terjadi apabila data yang dicari terletak pada index array yang paling depan, sehingga waktu yang dibutuhkan untuk mencari data sedikit.

*Worse Case*

*Worse case* kemungkinan terburuk akan terjadi apabila data yang dicari terletak pada *index array* yang paling akhir, sehingga waktu yang dibutuhkan untuk mencari data akan sangat lama.

Kelebihan *Sequential Search* diantaranya :

1. Sederhana

Algoritma pencarian berurutan adalah salah satu algoritma pencarian yang paling sederhana dan mudah dipahami. Ini membuatnya ideal untuk implementasi dalam situasi yang membutuhkan solusi cepat tanpa kompleksitas tambahan.

1. Fleksibilitas

Pencarian berurutan dapat diterapkan pada berbagai jenis data struktur, termasuk array, list, atau bahkan struktur data yang lebih kompleks.

1. Membutuhkan sedikit ruang

Algoritma ini hanya membutuhkan ruang untuk menyimpan data yang dicari, tanpa memerlukan struktur data tambahan. Ini membuatnya efisien dari segi penggunaan memori.

1. Mudah diimplementasikan

Karena sederhana, pencarian berurutan relatif mudah diimplementasikan, bahkan untuk pemrogram pemula (Wahyuni et al., 2022).

Kekurangan *Sequential Search* diantaranya :

1. Waktu eksekusi lama

Algoritma ini memiliki kompleksitas waktu rata-rata O(n), yang berarti waktu eksekusi meningkat secara linier dengan jumlah elemen dalam dataset. Oleh karena itu, untuk dataset yang besar, pencarian berurutan mungkin tidak efisien.

1. Tidak efisien untuk data terurut

Meskipun algoritma ini dapat digunakan pada data yang tidak terurut, ia tidak memanfaatkan fakta bahwa data telah diurutkan. Dalam kasus data terurut, algoritma pencarian biner akan jauh lebih efisien.

1. Performa tidak konsisten

Kinerja pencarian berurutan tidak selalu konsisten. Ini karena pencarian harus memeriksa setiap elemen dari awal hingga elemen yang dicari ditemukan atau sampai akhir dataset. Jika elemen yang dicari berada di posisi awal, pencarian akan lebih cepat; namun, jika berada di posisi terakhir, waktu eksekusi akan lebih lama.

Berikut adalah Langkah Langkah cara kerja *Sequential Search* :

1. Inisialisasi

Mulai dari awal himpunan data, atau indeks pertama, yang biasanya ditandai dengan indeks 0.

1. Periksa Elemen

Periksa setiap elemen satu per satu, dimulai dari elemen pertama (atau elemen pada indeks awal) dan terus berlanjut hingga elemen terakhir dari himpunan data.

1. Bandingkan

Saat memeriksa setiap elemen, bandingkan elemen yang sedang diperiksa dengan elemen yang ingin dicari.

1. Jika cocok

Jika elemen yang sedang diperiksa sama dengan elemen yang dicari, maka pencarian selesai dan indeks (atau posisi) elemen tersebut dikembalikan.

1. Jika tidak cocok

Jika elemen yang sedang diperiksa tidak sama dengan elemen yang dicari, lanjutkan pencarian ke elemen berikutnya dalam himpunan data.

1. Selesai   
   Jika telah memeriksa semua elemen dan tidak menemukan elemen yang dicari, pencarian dianggap selesai dan kembalikan nilai yang menandakan bahwa elemen tidak ditemukan (Herdiana, 2023).

Untuk meningkatkan efisiensi pencarian data pada Sequential Search dapat dilakukan dengan cara menghentikan looping dengan menggunakan break apabila data yang dicari sudah ketemu (Sonita & Sari, 2020)

Dengan demikian dari pengertian diatas dapat disimpulkan bahwa Sequential Search merupakan metode pencarian data dalam satu array dimana data yang dicari akan diuji satu persatu dari awal hingga akhir dan keuntungan algoritma ini adalah jika data yang dicari berada pada di awal data, maka pencarian akan cepat ditemukan.

### Database

Definisi dari *database* adalah kumpulan data yang terstruktur dan tersimpan secara sistematis dalam suatu sistem komputer, yang dapat diakses, dikelola, dan diperbarui secara efisien. Tujuan dari *database* adalah untuk menyediakan platform yang dapat menyimpan, mengelola, dan mengakses data dengan efisien, memberikan kemudahan dalam pencarian, manipulasi, dan analisis data, serta mendukung berbagai aplikasi dan proses bisnis dengan menyediakan informasi yang akurat dan konsisten. (’Afiifah et al., 2022)

### Perancangan Sistem

UML digunakan untuk menggambarkan perancangan awal dari sistem yang akan dibangun. UML memiliki banyak jenis permodelan, tetapi hanya beberapa yang digunakan saja yang akan dibahas. UML *(Unified Modeling Language)* adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkonsumsikan rancangan mereka dengan yang lain (Prihandoyo, 2020).

Berikut kelebihan Unified Modeling Language (UML):

1. Standar Industri

UML adalah standar industri yang diterima secara luas, sehingga memungkinkan komunikasi yang jelas dan efektif antara pengembang perangkat lunak, pemangku kepentingan, dan tim proyek

1. Peningkatan Komunikasi

UML menyediakan notasi grafis yang mudah dipahami, memungkinkan tim proyek untuk berkomunikasi dengan lebih baik dalam memahami dan merancang sistem perangkat lunak

1. Pemodelan yang komprehensif

UML mendukung berbagai jenis diagram yang dapat digunakan untuk memodelkan berbagai aspek sistem, termasuk struktur, perilaku, interaksi, dan arsitektur.

1. Analisis dan desain yang lebih baik

Dengan menggunakan UML, pengembang dapat melakukan analisis dan desain sistem dengan lebih terstruktur dan terperinci, sehingga menghasilkan solusi yang lebih baik.

1. Mendukung pengembangan berorientasi objek

UML dirancang untuk mendukung pemrograman berorientasi objek, sehingga memfasilitasi pemodelan sistem yang kompleks dan memungkinkan pengembang untuk fokus pada objek dan hubungan antara objek tersebut.

Berikut kekurangan Unified Modeling Language (UML):

1. Kompleksitas

UML memiliki banyak elemen dan konsep yang kompleks, yang dapat membuatnya sulit dipahami oleh pemula atau tim yang tidak terlatih.

1. Kelebihan Dokumen

Terkadang, terlalu banyaknya dokumentasi UML yang dihasilkan dapat mengarah pada kelebihan dokumentasi, yang menyebabkan dokumentasi yang tidak terkelola dengan baik atau tidak terkini.

1. Terlalu Umum

UML seringkali terlalu umum, sehingga beberapa konsep atau aspek dari sistem tertentu mungkin sulit untuk dimodelkan secara akurat dengan UML (Anggara et al., 2023).

*Use Case* Diagram

*Use Case* Diagram adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan. Menurut Shalahuddin dalam jurnal (Stinjak & Masya, 2021) mengungkapkan : “Diagram use case merupakan pemodelan untuk kelakuan (*behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.” Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2.2 Simbol *Use Case* Diagram

|  |  |  |
| --- | --- | --- |
| No | Simbol | Keterangan |
| 1 |  | Aktor yaitu mewakilkan tugas manusia, sistem maupun perangkat lain dalam mengkomunikasikan *use case*. |
| 2 |  | *Use case* yaituAbstraksi dan hubungan antara sistem dan aktor. |
| 3 |  | *Association* yaitu Abstraksi dari penghubung antara aktor dengan *use case*. |
| 4 | - - - - - - - -> | *Generalisasi* yaitu Menunjukkan spesialisasi aktor sehingga mereka dapat berpartisipasi dengan *use case*. |
| 5 | <<**include**>> | Menunjukkan bahwa *use case* sepenuhnya merupakan fungsi dari *use case* lain. |
| 6 | <<extend>> | Menunjukkan bahwa *use case* adalah pelengkap fungsional untuk *use case* lain jika suatu kondisi terpenuhi. |

*Class Diagram*

*Class Diagram* merupakan kumpulan objek-objek yang serupa. Setiap objek memiliki keadaan *(state)* dan perilaku *(behavior)*. State dari objek adalah kondisinya yang dijelaskan melalui atributnya. Perilaku objek menentukan bagaimana objek tersebut bertindak dan merespons situasi tertentu. Secara umum, class memiliki tiga area utama: nama *(stereotype)*, atribut, dan metode (operasi).

Menurut (Oktaviani et al., 2019) mengungkapkan: “Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas”.

Tabel 2.3 Simbol *Class* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Nama | Keterangan |
| 1 |  | *Generalization* | Yaitu hubungan objek anak (*descendent*) berbagi perilaku dan struktur data objek diatas objek induk (*ancestor)*. |
| 2 |  | *Nary Association* | Yaitu upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Yaitu kumpulan dari objek yang memiliki atribut dan fungsi yang sama. |
| 4 |  | *Collaboration* | Yaitu deskripsi urutan tindakan yang dilakukan oleh sistem yang menghasilkan hasil yang terukur bagi suatu aktor. |
| 5 | <- - - - - - | *Realization* | Yaitu tindakan yang sebenarnya dilakukan objek. |
| 6 | - - - - - -> | *Dependency* | Yaitu hubungan dimana perubahan elemen mandiri (*independent*) mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri. |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |

*Activity* Diagram

Menurut Rosa dalam jurnal (Handayani et al., 2023) mengungkapkan: “*Activity* Diagram menggambarkan work flow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem”.

*Activity* Diagram memiliki peran yang sangat penting dalam mengilustrasikan alur aktivitas dari system yang sedang berjalan. Dengan demikian, *Activity* diagram menjadi alat yang efektif dalam memahami dan mengkomunikasikan cara interaksi antara pengguna dan system aplikasi berlangsung dengan efisien.

Tabel 2.4 Simbol *Activity* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Keterangan |
| 1 |  | Status Awal | Yaitu sebuah diagram aktivitas memiliki keadaan awal. |
| 2 |  | Aktivitas | Yaitu aktivitas yang dilakukan oleh ocus, aktivitas biasanya dimulai dengan kata kerja. |
| 3 |  | Percabangan / Decision | Yaitu cabang ocus ada pilihan aktivitas yang lebih dari satu ocusar. |
| 4 |  | Percabangan / Join | Yaitu penggabungan ocus yang mana lebih dari satu kegiatan digabungkan menjadi satu. |
| 5 |  | Status Akhir | Yaitu keadaan akhir dari ocus, diagram aktivitas memiliki status akhir. |
| 6 |  | Swimlane | Yaitu memisahkan organisasi komersial yang bertanggung jawab atas aktivitas yang berlangsung. |

### *MYSQL*

*MySQL* adalah sebuah DBMS (*Database Management System*) menggunakan perintah *SQL* (*Structured Query Language*) yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis website. MySQL dibagi menjadi dua lisensi, pertama adalah *Free Software*dimana perangkat lunak dapat diakses oleh siapa saja. Dan kedua adalah *Shareware* dimana perangkat lunak berpemilik memiliki batasan dalam penggunaannya “MySQL adalah salah satu aplikasi *DBMS* yang sudah sangat banyak digunakan para pemrogram aplikasi web. Kelebihan dari *MySQL* adalah gratis, handal, selalu di-update dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi *DBMS* yang sering di bundling dengan web server sehingga proses instalasinya jadi lebih mudah” ( t bayu Kurniawan & Syarifuddin, 2020).

Berikut Kelebihan dari MySQL:

1. *Open Source*

*MySQL* tersedia sebagai perangkat lunak *open source*, yang berarti dapat digunakan secara gratis dan dikembangkan oleh komunitas pengembang yang besar.

1. Kinerja Tinggi

*MySQL* diketahui memiliki kinerja yang sangat baik, terutama untuk aplikasi web dengan jumlah pengguna yang besar atau tingkat transaksi yang tinggi

1. Dukungan Standar SQL

*MySQL* mendukung sebagian besar standar SQL, yang membuatnya mudah dipelajari dan diimplementasikan oleh para pengembang yang sudah terbiasa dengan bahasa kueri SQL.

1. Skalabilitas

MySQL dapat dengan mudah diintegrasikan dengan sistem yang lebih besar dan dapat diskalakan dengan baik, baik secara vertikal (dengan menambahkan lebih banyak sumber daya ke server tunggal) maupun horizontal (dengan menambahkan lebih banyak server).

1. Komunitas yang besar

*MySQL* memiliki komunitas pengguna yang besar, sehingga mudah untuk menemukan dukungan, tutorial, dan sumber daya lainnya secara online.

1. Berbagai pilihan Klien dan alat

*MySQL* mendukung berbagai klien dan alat pengelolaan basis data, termasuk *MySQL Workbench, phpMyAdmin*, dan banyak lagi.

1. Beragam Platform

MySQL dapat dijalankan di berbagai platform sistem operasi, termasuk Windows, Linux, dan macOS.

Berikut kekurangan dari *MySQL* :

1. Ketidakstabilan Versi Awal

Beberapa versi awal *MySQL* dapat dianggap kurang stabil dibandingkan dengan versi terbaru, meskipun kebanyakan masalah ini telah diperbaiki dengan rilis-rilis selanjutnya.

1. Keterbatasan Fitur

Beberapa fitur canggih atau kompleks yang tersedia di *DBMS* lain mungkin tidak didukung atau kurang optimal di *MySQL*

1. Manajemen Transaksi

Meskipun *MySQL* mendukung transaksi, manajemen transaksi dalam *MySQL* tidak sekuat sistem manajemen basis data relasional *(RDBMS)* yang lebih canggih seperti PostgreSQL.

1. Kesulitan Skalabilitas Horizontal

Meskipun *MySQL* dapat diskalakan secara horizontal, melakukan skala horizontal dengan benar memerlukan perencanaan yang baik dan pemahaman tentang konsep-konsep seperti replikasi dan partisi.

1. Kesulitan Manajemen Big Data

*MySQL* mungkin tidak menjadi pilihan terbaik untuk aplikasi yang memerlukan pemrosesan data dalam skala besar atau big data, seperti aplikasi data warehousing atau analisis data. Meskipun MySQL memiliki fitur untuk menangani data besar, terdapat DBMS lain yang lebih dioptimalkan untuk skenario tersebut, seperti Apache Hadoop atau Apache Spark.

### *PHP*

*PHP* (*Hypertext Prepocessor*) merupakan bahasa pemrograman berbasis *server–side* yang dapat melakukan parsing *script* php menjadi *script* web sehingga dari sisi client menghasilkan suatu tampilan yang menarik. PHP adalah PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *web* berbasis *server* (*server slide*) yang mampu memparsing kode PHP dari kode dengan ekstensi PHP sehingga menghasilkan tampilan website yang dinamis disisi *client* (Mubarak, 2019)*.*

Berikut Kelebihan dari PHP :

1. Mudah dipelajari

PHP dirancang untuk mudah dipelajari dan dimengerti, terutama bagi pemula dalam pengembangan web.

1. *Open Source*

PHP adalah perangkat lunak open source, yang berarti gratis untuk digunakan dan didukung oleh komunitas pengembang yang besar.

1. Platform Independen

PHP dapat dijalankan di berbagai platform sistem operasi, termasuk *Windows, Linux, macOS*, dan banyak lagi.

1. Integritas dengan database

PHP memiliki dukungan yang kuat untuk integrasi dengan berbagai sistem manajemen basis data, seperti *MySQL, PostgreSQL, SQLite*, dan lain-lain.

1. Fleksibel dan Dinamis

PHP adalah bahasa skrip yang sangat fleksibel dan dapat digunakan untuk membuat berbagai jenis aplikasi web, mulai dari situs web sederhana hingga aplikasi web yang kompleks.

Berikut Kekurangan pada PHP :

1. Tidak Terstruktur

PHP memiliki sejarah yang panjang, dan beberapa bagian dari bahasa ini mungkin terasa tidak terstruktur atau tidak konsisten.

1. Rentan terhadap Keamanan

PHP memiliki sejarah yang rumit dalam hal keamanan, dan beberapa kerentanannya telah dieksploitasi dalam masa lalu. Namun, dengan praktik pengkodean yang baik dan pembaruan reguler, banyak dari masalah ini dapat diatasi.

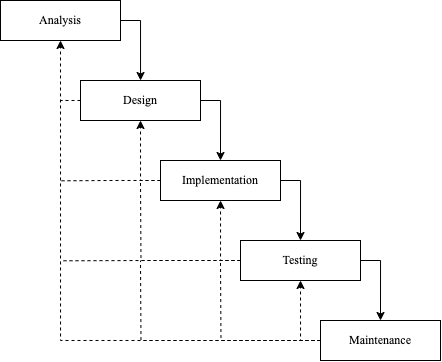
1. Kurangnya Alat Pengembangan yang kuat

Meskipun ada beberapa alat pengembangan yang baik untuk PHP, seperti PhpStorm dan Visual Studio Code, beberapa bahasa pemrograman lain memiliki alat pengembangan yang lebih kuat dan lengkap

1. Pengelolaan Memori yang buruk

PHP memiliki manajemen memori yang otomatis, yang membuatnya mudah digunakan tetapi juga dapat menyebabkan masalah kinerja pada aplikasi yang kompleks atau besar.

### Metode Waterfall

****

Gambar 2.1 Metode *Waterfall*(*Bassil*, 2012)

Model *Waterfall* adalah proses pengembangan perangkat lunak berurutan yang mana alur pengembangan mengalir semakin ke bawah (mirip dengan air terjun) melalui serangkaian fase yang harus dilaksanakan secara berurutan untuk berhasil membangun perangkat lunak komputer. Awalnya, model *Waterfall* diusulkan oleh *Winston W. Royce* pada tahun 1970 untuk menjelaskan praktik rekayasa perangkat lunak. Model *Waterfall* mendefinisikan beberapa fase yang harus diselesaikan berurutan, fase berikutnya dapat dikerjakan setelah fase sebelumnya selesai. (Bassil, 2012)

Metode *waterfall* merupakan pendekatan kerja yang menekankan urutan fase secara sistematis. Dinamakan *waterfall* karena prosesnya mengalir secara linear mirip dengan aliran air terjun. Pendekatan waterfall ini harus diikuti secara berurutan sesuai dengan tahapan yang telah ditentukan. (*Bassil*, 2012)

Berikut adalah tahap-tahap pengembangan dalam metode *waterfall*:

1. Analysis

Fase analisis melibatkan analis sistem dan bisnis untuk mengidentifikasi kebutuhan fungsional dan non-fungsional dari perangkat lunak. Kebutuhan fungsional adalah deskripsi mengenai apa saja fitur yang dapat dilakukan oleh sistem atau perangkat lunak yang dikembangkan, sedangkan kebutuhan non-fungsional adalah kebutuhan yang lebih berkaitan dengan bagaimana sistem tersebut dapat beroperasi.

1. *Design*

Fase ini melibatkan perencanaan dan perancangan berbagai aspek dari perangkat lunak, termasuk desain algoritma, arsitektur perangkat lunak, dan struktur database. Tujuannya adalah menghasilkan rencana yang jelas dan komprehensif untuk implementasi selanjutnya. Desain sistem dapat dibuat menggunakan berbagai alat seperti *Flowchart*, *Mind Map, Unified Modeling Language* (UML), atau *Entity Relationship Diagram* (ERD).

1. *Implementation*

Pada fase implementasi, desain perangkat lunak yang telah dirancang diimplementasikan menjadi kode yang dapat dieksekusi. Melibatkan proses menulis kode, mengkompilasi program, dan membuat basis data atau komponen perangkat lunak lainnya.

1. *Testing*

Setelah implementasi selesai, perangkat lunak diuji untuk memastikan bahwa perangkat lunak memenuhi persyaratan dan spesifikasi awal serta berfungsi sebagaimana mestinya. Pengujian ini melibatkan verifikasi dan validasi. Selama fase ini, *bug* dan masalah sistem lainnya diidentifikasi dan diperbaiki.

1. *Maintenance*

Fase terakhir adalah *maintenance* (pemeliharaan), dimana perangkat lunak disesuaikan dan diperbaiki setelah implementasi. Ini bisa meliputi perbaikan bug, peningkatan kinerja, atau penyesuaian dengan perubahan kebutuhan pengguna. Pemeliharaan juga memungkinkan adaptasi perangkat lunak terhadap lingkungan baru atau teknologi baru yang mungkin muncul setelah peluncuran. Perangkat lunak harus selalu dipantau untuk memastikan bahwa aplikasi tetap berfunngsi dengan baik. Perubahan atau perbaiakan dilakukan berdasarkan umpan balik pengguna atau jika ada masalah yang ditemukan.

### XAMPP

XAMPP adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: *Apache HTTP Server*, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), *Apache, MySQL*, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. Untuk mendapatkanya XAMPP anda dapat mendownload langsung dari web resminya. Dan berikut beberapa definisi program lainnya yang terdapat dalam XAMPP.

XAMPP ini merupakan project non-profit yang dikembangkan oleh Apache Friends yang didirikan Kai 'Oswalad' Seidler dan Kay Vogelgesang pada tahun 2002, project mereka ini bertujuan mempromosikan pengunaan Apache *web server* XAMPP adalah paket software yang didalamnya sudah terkandung *Web Server Apache, database MySQL*, dan PHP interpreter” (Rahmat et al., 2022)

Berikut kelebihan dari XAMPP :

1. Mudah di install dan digunakan

XAMPP memiliki installer yang mudah digunakan yang memungkinkan pengatur servel local dengan cepat.

Antarmuka pengguna yang ramah membuatnya cocok untuk pemula

1. Gratis dan *Open source*

XAMPP adalah perangkat lunak open source dan tersedia secara gratis, sehingga cocok untuk pengembang dengan anggaran terbatas.

1. Kompatibilitas *Multi-Platform*

Tersedia untuk *Windows*, *macOS*, dan *Linux*, sehingga pengembang dapat bekerja pada sistem operasi yang berbeda.

1. Paket Lengkap

XAMPP menyertakan *Apache*, *MariaDB* (atau *MySQL*), PHP, dan Perl dalam satu paket, yang menjadikannya solusi lengkap untuk pengembangan *web.*

1. Modular

Memungkinkan penambahan modul tambahan sesuai kebutuhan, seperti Tomcat untuk Java, *Mercury* untuk *mail server*, dan *FileZilla* *FTP Server*.

Berikut kekurangan pada XAMPP :

1. Keamanan

XAMPP tidak disarankan untuk digunakan di *server* produksi karena konfigurasi bawaannya yang kurang aman. Misalnya, beberapa fitur keamanan dinonaktifkan secara default untuk mempermudah pengembangan.

1. Penggunaan Memori

XAMPP dapat menggunakan banyak memori, terutama jika Anda menjalankan banyak modul secara bersamaan. Ini bisa menjadi masalah pada komputer dengan spesifikasi rendah.

1. Konfigurasi Manual

Beberapa fitur dan pengaturan mungkin memerlukan konfigurasi manual yang rumit, terutama jika Anda memiliki kebutuhan khusus yang tidak didukung oleh pengaturan default.

1. Kurangnya dukungan untuk lingkungan produksi

XAMPP tidak dirancang untuk lingkungan produksi, sehingga mungkin kurang efisien dan aman dibandingkan dengan solusi server yang dikonfigurasi secara profesional.

1. Perbedaan Versi

Versi perangkat lunak yang disertakan dalam XAMPP mungkin tidak selalu yang terbaru, dan ini bisa menyebabkan masalah kompatibilitas dengan aplikasi tertentu yang memerlukan versi yang lebih baru.

### *Laravel*

*Laravel* merupakan framework berbasis PHP yang dirilis oleh MIT yang menekankan kesederhanaan dan fleksibilitas pada desainnya. *Laravel* dibangun dengan konsep *MVC (Model-ControllerView),* kemudian dilengkapi juga dengan *command line tool* yang bisa digunakan untuk packaging bundle dan instalasi bundle melalui *command prompt.*

*Laravel* adalah kerangka kerja *(framework)* perangkat lunak sumber terbuka *(open-source)* yang ditulis dalam bahasa pemrograman PHP. Kerangka kerja ini dikembangkan oleh *Taylor Otwell* dan dirilis pertama kali pada tahun 2011. *Laravel* dirancang dengan tujuan menyederhanakan proses pengembangan aplikasi web dengan menyediakan seperangkat alat (tools) dan fitur-fitur yang kuat namun mudah digunakan. *Laravel* menonjol dengan keunggulan dalam mempermudah pengembangan aplikasi *web* melalui pendekatan yang menekankan kesederhanaan dan fleksibilitas. Dengan desain yang mengikuti konsep *Model-View-Controller (MVC)* (Mulyanto & Setiawan, 2020)

Fitur – fitur *framework* pun semakin relate dengan apa yang dibutuhkan oleh *developer*. Salah satunya *Laravel* yang merupakan *framework* PHP hasil pengembangan dari *framework Code Igniter*. Dengan adanya framework tentu sangat memudahkan para developer untuk menyederhanakan proses sehingga dapat menghemat waktu dan dapat menghasilkan performa yang maksimal. Penggunaan struktur MVC pada Laravel membuatnya mudah dipelajari dan mampu mempercepat proses pembuatan.

Beberapa fungsi *Laravel* meliputi :

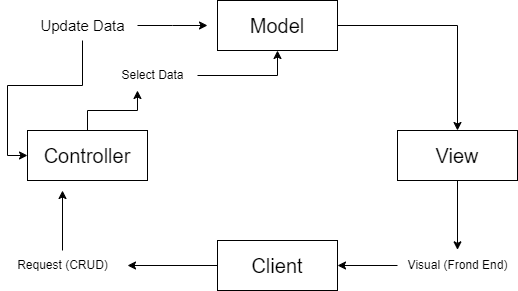
*Routing* yang Sederhana dan Kuat: *Laravel* menyediakan sistem routing yang mudah digunakan untuk mendefinisikan jalur *URL* dan mengarahkan permintaan HTTP ke pengendali (controller) yang sesuai.

ORM *(Eloquent)*: *Laravel* menggunakan Eloquent sebagai *Object-Relational Mapping (ORM)*, yang memungkinkan interaksi dengan database menggunakan model yang sesuai dengan tabel dalam *database*. *Eloquent* menyediakan cara yang elegan dan mudah untuk melakukan query dan manipulasi data.

*Blade Templating Engine: Laravel* memiliki Blade sebagai templating engine yang memungkinkan pengembangan halaman *web* dengan struktur dan logika yang bersih serta *reusable*

*Middleware*: *Laravel* mendukung middleware yang memungkinkan Anda menambahkan lapisan logika sebelum atau sesudah permintaan HTTP diproses oleh aplikasi. Ini sangat berguna untuk otentikasi, *logging*, dan manipulasi permintaan.

Artisan *CLI*: *Laravel* memiliki *command-line interface (CLI)* yang disebut Artisan. Artisan menyediakan berbagai perintah untuk membantu pengembang dalam tugas rutin seperti migrasi *database*, pembuatan model, *seeding*, dan lainnya.



Gambar 2.1 Arsitekruk *Laravel*

(Robby Yuli Endra, 2021)

*Laravel* menerapkan arsitektur MVC (Model-View-Controller) yang mana memisahkan antara tampilan *(HTML, CSS, JavaScript*) dan logika pemrograman (PHP*, MySQL*). Selanjutnya mari kita bedah apa itu MVC. Telah disinggung pada intro di atas bahwasanya MVC adalah arsitektur rancangan kode program yang dibuat terpisah-pisah.

### *Visual Studio Code*

*Visual Studio Code* merupakan software kode editor lintas platform yang ringan namun kuat yang dapat digunakan oleh siapa saja untuk membangun aplikasi Web. Muncul dengan builtin dukungan untuk JavaScript, naskah dan Node.js dan memiliki array beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C# , *Python*, dan PHP. *Visual Studio Code* menggunakan *open source* NET untuk memberikan dukungan ASP. *Visual Studio Code* dapat digunakan untuk berbagai bahasa pemrograman seperti JavaScript, HTML, CSS, PHP, *Python*, C++, dan masih banyak lagi. *Visual Studio Code* bekerja pada berbagai sistem operasi seperti *Windows*, *macOS*, dan *Linux* (Nanda Syarif et al., 2023)

Berikut kelebihan dari Visual Studio Code :

1. Gratis dan *Open Source*

*VS* *Code* adalah perangkat lunak gratis dan *open source*, yang berarti siapa saja bisa menggunakannya tanpa biaya dan berkontribusi pada pengembangannya.

1. Ringan dan Cepat

*VS* *Code* dirancang untuk menjadi ringan dan cepat, memungkinkan pengembang untuk memulai dan bekerja dengan proyek dengan cepat tanpa membebani sistem.

1. *Cross-Platform*

*VS Code* tersedia untuk berbagai platform, termasuk *Windows, macOS*, dan Linux, sehingga pengembang bisa bekerja di lingkungan yang mereka pilih.

1. Ekosistem Ekstensi yang Luas

*VS Code* memiliki ekosistem ekstensi yang sangat luas, memungkinkan pengembang untuk menambahkan fungsionalitas tambahan sesuai kebutuhan mereka, seperti dukungan untuk berbagai bahasa pemrograman, alat pengembangan, dan fitur tambahan.

1. *VS Code* memiliki integrasi Git bawaan yang memungkinkan pengembang untuk mengelola versi kode mereka secara langsung dari editor, termasuk commit, push, pull, dan banyak lagi.
2. Fitur *IntelliSense*

*VS Code* menyediakan fitur *IntelliSense* yang memberikan saran cerdas, autocompletion, dan dokumentasi *inline*, yang sangat membantu dalam meningkatkan produktivitas pengembangan.

1. *Debugging* yang kuat

*VS Code* memiliki dukungan debugging bawaan untuk banyak bahasa pemrograman, memungkinkan pengembang untuk menjalankan dan mendiagnosis kode mereka secara langsung dari editor.

1. Terminal terintegritas

*VS Code* memiliki terminal terintegrasi, sehingga pengembang bisa menjalankan perintah shell tanpa meninggalkan editor.

Berikut kekurangan *Visual Studio Code*

1. Kinerja pada Proyek Besar

Meskipun *VS Code* dirancang untuk menjadi ringan, beberapa pengguna melaporkan bahwa performanya bisa menurun saat bekerja dengan proyek yang sangat besar atau memiliki banyak ekstensi yang terpasang.

1. Kustomisasi yang Rumit

Meskipun *VS Code* sangat dapat dikustomisasi, proses kustomisasi bisa menjadi rumit dan memerlukan pengetahuan mendalam tentang file konfigurasi Meskipun *VS Code* sangat dapat dikustomisasi, proses kustomisasi bisa menjadi rumit dan memerlukan pengetahuan mendalam tentang file konfigurasi.

1. Penggunaan Memori

*VS Code* bisa mengkonsumsi banyak memori terutama jika banyak ekstensi yang diinstal dan digunakan secara bersamaan, yang dapat menjadi masalah pada komputer dengan spesifikasi rendah.

1. Tidak Semua Fitur IDF Penuh

Meskipun *VS Code* sangat kuat, itu masih lebih merupakan editor kode daripada IDE penuh. Beberapa fitur lanjutan yang ditemukan di IDE seperti *Visual Studio*, *IntelliJ IDEA*, atau *Eclipse* mungkin tidak tersedia atau memerlukan ekstensi tambahan.

1. Ketergantungan pada Ekstensi

Banyak fungsionalitas VS Code bergantung pada ekstensi. Jika ekstensi tidak dikembangkan atau dikelola dengan baik, ini bisa menyebabkan masalah kompatibilitas atau performa.

### Figma

Figma merupakan sebuah platform inovatif yang telah dirancang khusus untuk memfasilitasi proses desain grafis yang komprehensif. Dengan fokus utama pada pembuatan *wireframes*, *prototipe*, serta antarmuka pengguna, aplikasi ini telah menjadi pilihan utama bagi para profesional desain untuk menghasilkan produk digital yang memukau, baik itu aplikasi smartphone maupun situs *web* (Yaya Suharya, Nurul Imamah, 2023). Keunggulan utama dari Figma terletak pada kemampuannya untuk diakses melalui *browser*, memungkinkan kolaborasi yang mudah antara tim desain yang tersebar geografis. Dengan demikian, Figma tidak hanya menjadi alat desain yang andal, tetapi juga menjadi platform yang mempercepat proses pengembangan produk digital dengan kualitas yang optimal (R. Kurniawan & Budi, 2022)

Berikut kelebihan Figma :

1. Kolaborasi *Real-time*

Figma memungkinkan beberapa desainer untuk bekerja pada file yang sama secara bersamaan, mirip dengan cara *Google Docs* bekerja. Ini mempermudah kolaborasi tim, terutama pada proyek yang melibatkan banyak pemangku kepentingan.

1. Berbasis *Website* Karena Figma berbasis *web*, tidak perlu mengunduh dan menginstal perangkat lunak pada setiap komputer. Ini juga berarti file desain dapat diakses dari mana saja selama ada koneksi internet.
2. *Platform* *Independen*

Figma dapat berjalan di berbagai sistem operasi, termasuk *Windows*, *macOS*, dan *Linux*, karena hanya memerlukan *browser web*.

1. Versi Gratis

Figma menawarkan versi gratis dengan fitur yang cukup untuk banyak pengguna individu dan tim kecil. Ini memungkinkan pengguna mencoba dan menggunakan Figma tanpa investasi awal.

1. *Prototyping* dan Animasi

Figma memiliki alat prototyping dan animasi yang memungkinkan desainer untuk membuat dan menguji interaksi langsung dalam desain mereka tanpa memerlukan perangkat lunak tambahan.

1. Integritas dengan alat lainFigma mendukung integrasi dengan berbagai alat lain seperti *Slack*, *Zeplin*, dan JIRA, yang dapat meningkatkan produktivitas dan alur kerja tim.
   * 1. **Pengujian Blackbox**

*Black box* adalah suatu konsep yang mengacu pada suatu sistem atau perangkat yang dianggap sebagai entitas tunggal yang menerima input dan menghasilkan output tanpa memberikan penjelasan detail tentang bagaimana proses internalnya bekerja. Dalam konteks ini, meskipun input dan output diamati, pengetahuan tentang mekanisme kerja internal dari *black box* tidak tersedia atau tidak diungkapkan kepada pengguna eksternal. Dengan kata lain, *black box* berfungsi sebagai suatu "kotak hitam" di mana pengguna hanya dapat mengamati apa yang dimasukkan ke dalamnya dan apa yang keluar darinya, tanpa mengetahui secara spesifik bagaimana proses di dalamnya terjadi atau bekerja (Titi Komalasari, 2021).

* + 1. ***Technology Acceptance Model (TAM)***

TAM yang diperkenalkan pertama kali oleh Fred D. Davis pada tahun 1986 adalah adaptasi dari *TRA (Theory of Reasoned Action)* yang dibuat khusus untuk pemodelan penerimaan penggunaan terhadap sistem informasi. Menurut Davis, tujuan utama TAM adalah untuk memberikan dasar untuk penelusuran pengaruh faktor eksternal terhadap kepercayaan, sikap, dan tujuan pengguna. TAM menganggap bahwa 2 keyakinan individual, yaitu persepsi manfaat (*perceived usefulness*, disingkat PU) dan persepsi kemudahan penggunaan (*perceived easy of use*, disingkat PEOU), adalah pengaruh utama untuk perilaku penerimaan komputer. Dalam memformulasikan TAM, Davis menggunakan TRA sebagai grand teorinya namun tidak mengakomodasi semua komponen. Davis hanya memanfaatkan komponen *‘Belief’* dan *‘Attitude’* saja, sedangkan *Normative Belief* dan *Subjective Norms* tidak digunakannya.

Persepsi kegunaan *(perceived usefulness)* menggambarkan tingkat kepercayaan seseorang bahwa penggunaan sistem akan meningkatkan kinerjanya. Orang menggunakan teknologi informasi karena mempunyai keyakinan bahwa prestasi dan kinerja akan meningkat. Konsep ini menggambarkan ukuran dimana penggunaan suatu teknologi dipercaya akan mendatangkan manfaat bagi pemakainya. Kemudahan penggunaan yang dirasakan *(perceived ease of use)* menggambarkan tingkat kepercayaan seseorang bahwa penggunaan sistem informasi merupakan hal yang mudah dan tidak memerlukan usaha keras dari pemakainya. Kemudahan ini akan mengurangi tenaga, pikiran dan waktu yang digunakan untuk mempelajari dan menggunakan sistem informasi. Orang yang bekerja dengan sistem informasi, bekerja lebih mudah dibandingkan orang yang bekerja secara manual tanpa sistem informasi.

Sikap terhadap penggunaan *(attitude toward use)* merupakan kecenderungan tanggapan awal atas kondisi yang menyenangkan maupun tidak menyenangkan pada suatu objek tertentu. Secara teoritis, sikap merupakan refleksi perasaan seseorang atas objek dalam kondisi baik atau buruk, menguntungkan atau merugikan. Sikap muncul karena seseorang memiliki nilai yang ditentukan oleh kepercayaan atas objek tersebut.

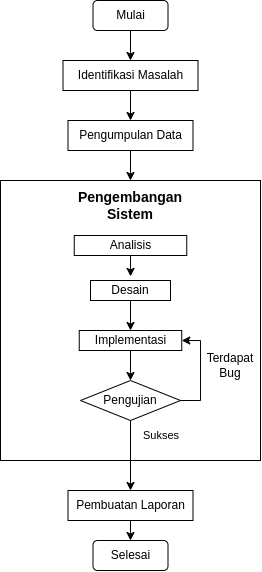
Pada kondisi lain, perilaku tertentu juga dapat mempengaruhi kepercayaan baru seseorang sehingga membawa perubahan pada sikap. Perilaku *(behavior)* dilakukan karena individu mempunyai niat atau keinginan untuk melakukan atau niat berperilaku akan menentukan perilakunya. Niat berperilaku *(behavioral intention)* adalah suatu keinginan seseorang untuk melakukan suatu perilaku tertentu atau kecenderungan seseorang untuk tetap menggunakan teknologi tertentu. Tingkat penggunaan teknologi seseorang dapat diprediksi dari sikap perhatiannya terhadap teknologi tersebut, misal motivasinya untuk tetap menggunakan maupun memotivasi penggunaan lain atau menambah perangkat pendukung.

Perilaku adalah penggunaan sesungguhnya *(actual usage)* teknologi itu sendiri atau kondisi nyata penggunaan sistem informasi. Perilaku atau penggunaan sesungguhnya sulit diobservasi dan diukur melalui daftar pertanyaan. Hasil penelitian TAM, menunjukkan bahwa penggunaan sistem informasi dapat diprediksi dengan baik dengan menggunakan variabel niat berperilaku *(behavioral intention)* (Rahayu, Budiyanto and Palyama, 2017).

# BAB III ANALISIS DAN PERANCANGAN

## Kerangka Pikir

Kerangka pikir merupakan jalur pemikiran yang dirancang berdasarkan kegiatan penyusun yang dilakukan. Berikut adalah kerangka pikir yang merupakan langkah-langkah yang akan dilakukan dalam penelitian ini:

****

Gambar . Kerangka Pikir

## Deskripsi

Deskripsi adalah prinsip yang terlibat dalam proses mengubah data menjadi format yang dapat dijelaskan dan disajikan dengen jelas dan akurat untuk mencapai tujuan tertentu. Hal ini memungkinkan informasi tersebut dapat dipahami dan dimengerti oleh pembaca dengan baik

### Identifikasi Masalah

Tahap pertama dalam merancang aplikasi sikibe (sistem informasi koperasi be7) dalam kerangka pikir penulis adalah mengidentifikasi masalah atau menetapkan perumusan masalah,. Dalam hal ini penulis meminta izin kepada pihak kepala koperasi untuk melakukan penelitian, serta melakukan observasi dan wawancara guna memahami masalah yang dihadapi oleh bagian petugas koperasi. Permasalah yang terdeteksi di koperasi SMK Negeri 7 adalah pencarian data anggota masih dilakukan secara manual, pencatatan transaksi simpan pinjam masih dilakukan secara manual dengan menulis di buku besar, dan pembuatan pelaporan transaksi simpan pinjam masih juga dilakukan dengan menulis di buku besar

### Pengumpulan Data

Dalam pengumpulan data, penulis menggunakan beberapa metode untuk mendapatkan data yang akurat yang diperlukan dalam pembuatan aplikasi dan penyusunan laporan, yaitu sebagai berikut:

1. Observasi

Yaitu dilakukan dengan cara mengamati langsung sistem informasi koperasi yang berjalan di SMKN 7 Baleendah dan penulis mendapatkan informasi dari petugas koperasi agar sistem yang sedang berjalan ini bisa di perbaiki dengan menerapkan digitalisasi.

1. Wawancara (Interview)

Yaitu dilakukan dengan proses tanya jawab dengan kepala koperasi SMKN 7 Baleendah mengenai sistem koperasi simpan pinjam pada yang sedang berjalan saat ini. Selain itu, kepala koperasi ibu tetet sopiah mengetahui kesulitan dan kebutuhan apa saja yang diperlukan dalam sistem yang akan dibangun oleh penulis.

1. Studi Pustaka

Yaitu dilakukan dengan mengumpulkan data dari beberapa buku, jurnal, dan sumber lainnya yang bersangkutan dengan aplikasi koperasi simpan pinjam menggunakan algoritma sequential search.

## Analisis Kebutuhan

Tahap selanjutnya adalah penulis merancang dan merencanakan aplikasi koperasi simpan pinjam yang dimulai dengan analisis kebutuhan perangkat lunak dan perangkat keras yang diperlukan untuk pengembangan aplikasi koperasi simpan pinjam yang bernama “Aplikasi Sikibe”. Pada tahap ini, dilakukan pengumpulan kebutuhan elemen-elemen pada tingkat perangkat lunak.

Pengumpulan Data

1. Input data
2. Validasi data

Penyimpanan Data

1. Manajemen Database
2. Keamanan Data

Pemrosesan Data

1. Tranformasi Data
2. Integrasi Data

Pengambilan Data

1. Query dan Retrieval
2. API untuk akses data

Analisis dan Pelaporan Data

1. Analitik Data
2. Visualisasi Data
3. Pelaporan

Pemeliharaan Data

1. Backup dan Recovery
2. Versi dan Audit Trail

Akses dan Pengendalian

1. Manajemen Akses Pengguna
2. Log Aktivitas

Keberlanjutan dan Skalabilitas

1. Scalability
2. Performance monitoring

Kepatuhan dan Standar

1. Regulasi Data
2. Data Quality Management

### Desain

Setelah menyelesaikan analisis kebutuhan, tahap desain dilaksanakan untuk merancang arsitektur sistem dan antarmuka pengguna Aplikasi Sikibe. Pada langkah ini, dilakukan perencanaan basis data, perancangan struktur program, dan merancang antarmuka pengguna (UI) yang mudah dipahami dan *user-friendly*.

Dengan melakukan proses ini secara hati-hati, tahap desain dapat menghasilkan panduan yang kokoh untuk mengembangkan Aplikasi Sikibe yang sesuai dengan kebutuhan yang telah diidentifikasi sebelumnya.

Perancangan Diagram

Pada tahap ini dilakukan dengan membuat desain menggunakan bahasa pemodelan *Unified Modelling Language* (UML). UML yang digunakan dalam penelitian terdiri dari:

* 1. *Use case* diagram

digunakan untuk menggambarkan secara umum hubungan antara aktor dan kasus pengguna dalam suatu sistem aplikasi. Diagram ini membantuk dalam pemahaman tentang interaksi antara aktor (entitas yang berinteraksi dengan sistem) dan berbagai fungsi (*use case*) yang ada dalam sistem tersebut. Dalam konteks ini, terdapat satu aktor yaitu “Admin” dan kasus pengguna (use case) yang mencakup :

1. Halaman *Login* berfungsi untuk memungkinkan pengguna (admin dan anggota) untuk masuk ke aplikasi menggunakan nama pengguna dan kata sandi.
2. Halaman *Register* berfungsi untuk memungkinkan calon anggota untuk membuat akun baru dengan mengisi informasi pribadi seperti nama, alamat, nomor kontak, dan informasi lain yang diperlukan. Halaman ini memastikan bahwa hanya anggota terdaftar yang dapat mengakses layanan koperasi.
3. Halaman Beranda berfungsi untuk menampilkan ringkasan informasi penting seperti total simpanan, pinjaman, notifikasi terbaru, dan tautan cepat ke fitur utama lainnya.
4. Halaman Pengguna berfungsi untuk memungkinkan admin untuk melihat, menambah, mengedit, dan menghapus data pengguna. Halaman ini membantu admin dalam mengelola akun pengguna dengan efektif dan memastikan data pengguna selalu *up-to-date*.
5. Halaman Anggota berfungsi untuk menyediakan informasi lengkap mengenai setiap anggota, termasuk profil, riwayat transaksi, dan status keanggotaan. Admin dapat menggunakan halaman ini untuk memantau aktivitas anggota dan menjaga komunikasi yang efektif.
6. Halaman Simpan berfungsi untuk memungkinkan anggota untuk melihat detail simpanan mereka, menambah simpanan baru, dan memantau perkembangan simpanan. Admin juga dapat mengelola catatan simpanan dan menghasilkan laporan terkait simpanan anggota.

Halaman Pinjam berfungsi untuk memungkinkan anggota untuk mengajukan pinjaman, melihat status pinjaman, dan memantau pembayaran angsuran. Admin dapat menggunakan halaman ini untuk memproses aplikasi pinjaman, memantau status pembayaran, dan menghasilkan laporan pinjaman.

* 1. *Activity* diagram

digunakan untuk memodelkan rangkain aktivitas dalam suatu sistem, mencakup representasi visual dari urutan aktivitas yang dilakukan oleh pengguna atau entitas lain dalam aplikasi. Diagram ini memberikan pandangan keseluruhan tentang bagaimana aktivitas berlangsung, berinteraksi, dan mengalir dalam sistem. Penulis akan membuat beberapa diagram aktivitas yang berkaitan dengan sistem aplikasi diantaranya:

1. Proses akses login
2. Proses akses menu beranda
3. Proses akses menu Simpanan
4. Proses akses menu Pinjaman (Angsuran)
5. Proses akses menu Anggota
   1. Perancangan *Interface*

Pada tahap perancangan antarmuka, desain aplikasi dibuat sederhana menggunakan Figma, namun tetap mampu mengilustrasikan aplikasi yang dibangun dan memperhatikan interaksi antara pengguna dan sistem, dalam proses ini, desain aplikasi dibuat dengan menggunakan Figma yang lebih canggih, tetapi masih mampu menggambarkan fungsionalitas aplikasi serta menunjukan bagaimana interaksi antara pengguna dan sistem akan terjadi di dalamnya.

### Implementasi

Setelah melakukan perancangan diagram dan antarmuka, tahap selanjutnya adalah melakukan implementasi perancangan yang telah dibuat selanjutnya dengan cara menerjemahkan kedalam bentuk pengkodean secara nyata memanfaatkan bahasa pemrograman PHP, *Visual Studio Code* (VS Code) dan database menggunakan MySQL serta *framework Laravel*.

### Pengujian

Pada tahap ini, pengujian dilakukan untuk memastikan bahwa hasilnya sesuai dengan desain yang diinginkan. Jika terdapat masalah, koreksi dilakukan kembali pada tahap implementasi hingga semua berjalan dengan baik sesuai yang diharapkan. Pengujian sistem menggunakan metode *Black Box Testing*, sementara pengujian perangkat menggunakan metode fungsionalitas, kinerja, dan keamanan. Apabila terdapat kekurangan, kembali ke tahap implementasi untuk diperbaiki sehingga sesuai dengan desain yang diinginkan.

Rencana pengujian sistem merupakan hal penting dalam proses pembangunan sistem, hal ini bertujuan untuk menemukan kesalahan serta kekurangan yang terdapat pada sistem yang akan diuji. Adapun rencana pengujian dari sistem yang penulis teliti yaitu dengan menggunakan pengujian *alpha* (*alpha testing*) dan pengujian *beta* (beta testing).

Pada pengujian *Alpha*, pengujian akan dilakukan oleh peneliti dengan cara *Black box testing*, pengujian *alpha* atau pengujian fungsional dilakukan terhadap sistem untuk memastikan yang telah dibangun sesuai dengan kebutuhan dan tujuan yang telah ditetapkan. Untuk menentukan pengujian alpha ini dilakukan dengan cara skenario pengujian *alpha*.

Rencana Pengujian *beta* ini akan dilakukan dengan menyebarkan kuesioner yang terdiri dari lima variable dan 18 pernyataan yang terbagi menjadi lima yaitu dari segi kemudahan penggunaan (*perceived ease of use*), manfaat penggunaan (*perceived usefulness*), perilaku pengguna (*attitude toward using*), dan niat perilaku untuk tetap menggunakan (*behavioral intention use*). Adapun yang akan melakukan pengujian beta pada Sistem Informasi adalah kepala koperasi, guru, dan staff tata usaha. Penentuan sampel dalam penelitian ini dapat menggunakan teknik penarikan sampel acak sederhana, dimana cara pengambilan sampel dari anggota populasi dengan menggunakan acak tanpa memberikan strata (tingkatan) dalam anggota populasi tersebut.

Maka dari itu, dapat disimpulkan bahwa dalam penelitian ini, jumlah sampel yang akan diteliti adalah 70 orang, dengan menggunakan *margin of error* sebesar 10% atau 0,1 (e). Pengujian beta ini dilakukan dengan menyebarkan kuesioner kepada 70 responden, dan hasil dari kuesioner ini akan diolah menggunakan aplikasi IBM SPSS.

### Pelaporan

Langkah terakhir adalah menyusun laporan skripsi yang merupakan satu-satunya komponen yang dinilai sesuai dengan kriteria yang telah ditetapkan dalam formulir pengajuan proposal, dan juga merupakan syarat untuk mencapai kelulusan di Fakultas Teknologi dan Informasi Universitas Bale Bandung. Laporan skripsi ini dipersiapkan sesuai dengan panduan penulisan skripsi yang diberikan oleh fakultas. Laporan ini adalah hasil akhir dari proses penelitian yang menunjukkan kualitas ilmiah dan ketaatan terhadap standar yang telah ditetapkan dalam panduan tersebut.

# BAB IV ANALISIS DAN PERANCANGAN

## Analisis

Pada tahap ini, analisis yang dilakukan dalam proses penelitian ini meliputi beberapa tahapan yaitu Analisis masalah, analisis *software*, analisis pengguna, *User interface*, fitur-fitur, analisis data serta analisis biaya.

### Analisis Masalah

Berdasarkan observasi dan wawancara ditemukan bahwa, koperasi SMK Negeri 7 Baleendah masih menghadapi beberapa permasalahan utama. Proses pencarian data anggota masih dilakukan secara konvensional, pencatatan transaksi simpan pinjam tidak bisa diakses di mana saja, pencatatan pelaporan transaksi simpan pinjam masih disajikan secara konvensional. Masalah ini menyebabkan lamanya pengambilan keputusan, kesulitan dalam pencarian data konsumen, dan ketidakakuratan informasi yang diberikan kepada konsumen. Selain itu, pelaporan transaksi memakan waktu lama karena pencarian data dari arsip buku yang tersimpan secara manual. Dengan jumlah anggota yang banyak dan petugas koperasi yang terbatas, hal ini mengakibatkan lambannya kinerja koperasi. Hingga kini, koperasi belum memanfaatkan digitalisasi atau aplikasi khusus untuk mempermudah pengelolaan data dan dokumen, sehingga semua arsip konsumen masih dicatat secara manual dalam bentuk kertas.

### Analisis Software

Berdasarkan analisis kebutuhan maka untuk memenuhi kebutuhan dalam pengembangan aplikasi SIKIBE (Sistem Informasi Koperasi BE7)memerlukan software sebagai alat penunjang pengembangan aplikasi SIKIBE (Sistem Informasi Koperasi BE7). Adapun *software* yang dibutuhkan sebagai alat penunjang pada penelitian ini adalah sebagai berikut:

1. Sistem operasi windows 10.
2. *Framework laravel* digunakan untuk mengembangkan sistem dengan mudah dan cepat.
3. *Visual Studio Code* sebagai *text editor* digunakan untuk melakukan pengkodean dalam pembuatan aplikasi SIKIBE.
4. *XAMPP* digunakan untuk *web server*.
5. *Web browser* digunakan untuk melihat tampilan sistem yang sudah dibuat.
6. *Draw.Io* digunakan untuk membuat atau merancang diagram UML.
7. *Figma* digunakan untuk membuat rancangan antarmuka sistem aplikasi SIKIBE.
8. *Domain sikibe.mazalio.com*
9. *Hosting biznet neo lite*

### Analisis Pengguna

Analisis pengguna adalah yang berkaitan dengan yang akan memakai Aplikasi SIKIBE. Terdapat dua kelompok pengguna utama, yaitu kepala koperasi selaku admin dan masyarakat sekolah, khususnya guru dan jajarannya, sebagai user. Kepala koperasi memiliki akses penuh terhadap aplikasi untuk mengelola data konsumen, transaksi, dan laporan keuangan. Sebagai admin, kepala koperasi bertanggung jawab untuk menambah, mengedit, dan menghapus data, serta memantau status simpan pinjam. Aplikasi ini juga memungkinkan kepala koperasi untuk membuat laporan keuangan dengan cepat dan akurat, sehingga mendukung pengambilan keputusan yang lebih efisien dan efektif.

Karyawan sekolah, khususnya guru dan staf, menggunakan aplikasi sebagai user untuk mengakses informasi tentang simpan pinjam mereka. Pengguna dapat melihat saldo simpanan, status pinjaman, dan riwayat transaksi mereka melalui aplikasi. Dengan akses mudah ke informasi ini, guru dan staf dapat lebih efektif dalam mengelola keuangan pribadi mereka terkait koperasi. Aplikasi ini dirancang untuk meningkatkan transparansi, efisiensi, dan kemudahan akses informasi bagi semua anggota koperasi di lingkungan sekolah.

### *User interface*

*User interface* dari aplikasi SIKIBE sangat berpengaruh untuk kenyamanan user saat menggunakannya. Hal ini mencakup perangkat yang dibuat ditunjukan untuk menghasilkan informasi yang cepat, maka *User interface* juga harus menyesuaikan dengan kebutuhan user berupa tampilan yang mudah dipahami. Berikut adalah *User interface* aplikasi SIKIBE (Sistem Informasi Koperasi BE7), meliputi:

1. *User interface* halaman *login*
2. *User interface* halaman *register*
3. *User interface* halaman beranda
4. *User interface* halaman pengguna
5. *User interface* halaman anggota
6. *User interface* halaman simpan
7. *User interface* halaman pinjam

### Fitur-fitur

Fitur-fitur yang digunakan dalam aplikasi SIKIBE ini dibuat agar pengguna dapat dengan mudah mendapatkan informasi dan mengetahui detail simpan pinjam mereka. Berikut adalah fitur-fitur yang disediakan dalam aplikasi SIKIBE:

1. Halaman *Login* berfungsi untuk memungkinkan pengguna (admin dan anggota) untuk masuk ke aplikasi menggunakan nama pengguna dan kata sandi.
2. Halaman *Register* berfungsi untuk memungkinkan calon anggota untuk membuat akun baru dengan mengisi informasi pribadi seperti nama, alamat, nomor kontak, dan informasi lain yang diperlukan. Halaman ini memastikan bahwa hanya anggota terdaftar yang dapat mengakses layanan koperasi.
3. Halaman Beranda berfungsi untuk menampilkan ringkasan informasi penting seperti total simpanan, pinjaman, notifikasi terbaru, dan tautan cepat ke fitur utama lainnya.
4. Halaman Pengguna berfungsi untuk memungkinkan admin untuk melihat, menambah, mengedit, dan menghapus data pengguna. Halaman ini membantu admin dalam mengelola akun pengguna dengan efektif dan memastikan data pengguna selalu *up-to-date*.
5. Halaman Anggota berfungsi untuk menyediakan informasi lengkap mengenai setiap anggota, termasuk profil, riwayat transaksi, dan status keanggotaan. Admin dapat menggunakan halaman ini untuk memantau aktivitas anggota dan menjaga komunikasi yang efektif.
6. Halaman Simpan berfungsi untuk memungkinkan anggota untuk melihat detail simpanan mereka, menambah simpanan baru, dan memantau perkembangan simpanan. Admin juga dapat mengelola catatan simpanan dan menghasilkan laporan terkait simpanan anggota.
7. Halaman Pinjam berfungsi untuk memungkinkan anggota untuk mengajukan pinjaman, melihat status pinjaman, dan memantau pembayaran angsuran. Admin dapat menggunakan halaman ini untuk memproses aplikasi pinjaman, memantau status pembayaran, dan menghasilkan laporan pinjaman.

### Analisis Data

Untuk mendukung perancangan aplikasi SIKIBE, penulis memerlukan beberapa data sebagai bahan penelitian dan perancangan aplikasi. Adapun data-data yang digunakan dan diolah pada aplikasi SIKIBE adalah sebagai berikut:

1. Data Simpanan

Analisis data simpanan akan memberikan detail mengenai jumlah simpanan anggota koperasi, termasuk tanggal simpanan, jumlah uang yang disimpan, dan jenis simpanan. Dengan analisis ini, anggota koperasi dapat memantau simpanan mereka, sehingga memudahkan dalam mengambil keputusan terkait penambahan atau penarikan simpanan.

1. Data Peminjaman

Analisis data peminjaman akan memberikan informasi mengenai detail pinjaman yang diajukan oleh anggota koperasi. Informasi ini meliputi jumlah pinjaman, jangka waktu pengembalian, bunga pinjaman, dan status pembayaran. Admin koperasi dapat menggunakan data ini untuk menyusun laporan pinjaman dan memantau pembayaran angsuran, sehingga dapat mengambil keputusan berdasarkan data yang akurat dan terbaru.

1. Data Anggota

Analisis data anggota mencakup informasi mengenai identitas dan status keanggotaan setiap anggota koperasi. Data ini meliputi nama, alamat, nomor kontak, dan riwayat transaksi simpan pinjam. Dengan data ini, admin koperasi dapat mengelola keanggotaan dengan lebih efektif, memastikan semua data anggota selalu terkini, serta memudahkan dalam melakukan komunikasi dan penyampaian informasi kepada anggota.

### Analisis Biaya

Pada penelitian kali ini ada beberapa rincian biaya yang dibutuhkan dalam proses pengerjaan penelitian diantaranya adalah:

Tabel 4.1 Analisis Biaya

|  |  |  |
| --- | --- | --- |
| No | Jenis Kebutuhan | Biaya |
| 1 | Biaya ATK | Rp. 1.500.000 |
| 2 | Internet | Rp. 1.500.000 |
| 3 | Analisis | Rp. 2.000.000 |
| 4 | *Programming* | Rp. 5.500.000 |
| 5 | Biaya Bahan Bakar | Rp. 1.200.000 |
| Jumlah | | Rp. 11.700.000 |

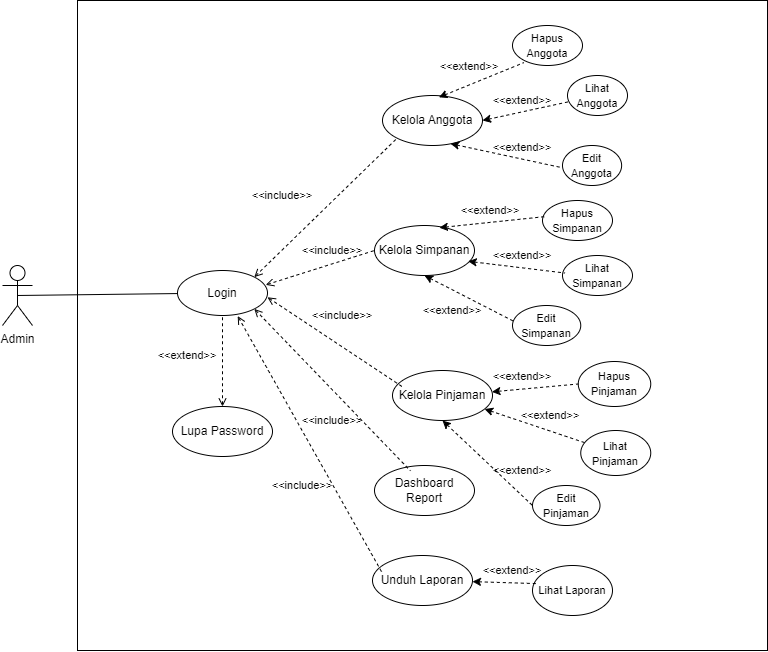
## Perancangan

Sebelum masuk dalam pembuatan aplikasi dibutuhkan perancangan yang bertujuan untuk memudahkan pengguna agar mendapatkan informasi mengenai aplikasi SIKIBE (Sistem Informasi Koperasi BE7) yang diantaranya yaitu merancang *software* dalam bentuk *Unified Modelling Language* (UML) yang terdiri dari *use case* diagram, *Activity* diagram, dan *class* diagram.

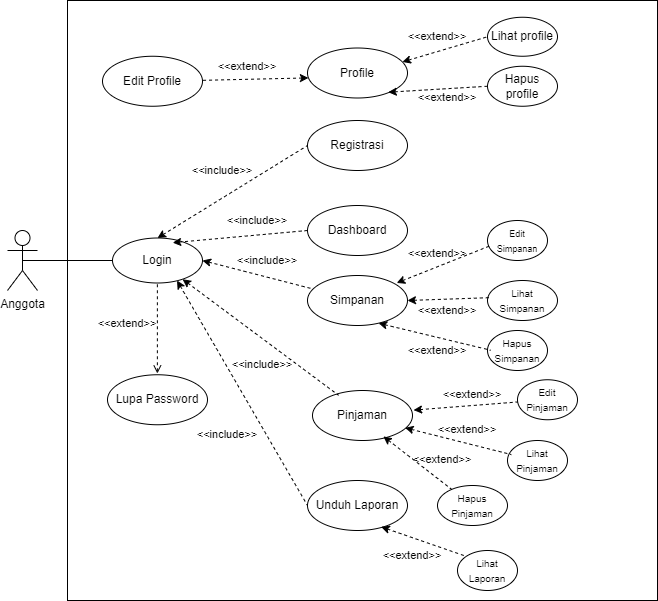
### *Unified Modeling Language* (UML)

1. *Usecase* Diagram

Pada rancangan *use case* diagram bertujuan untuk memberikan gambaran interaksi aktor dengan aplikasi SIKIBE (Sistem Informasi Koperasi BE7). Yang terdiri dari beberapa modul utama, seperti management anggota, transaksi simpanan, transaksi pinjaman, dan laporan keuangan. Diagram ini membantu dalam memahami alur kerja sistem dan memastikan bahwa semua proses di dukung oleh sistem yang baik. Berikut adalah penjelasan Use Case diagram Aplikasi SIKIBE:

1. *Usecase* Diagram

Gambar . *Usecase* Diagram Admin



Gambar . *Usecase* Diagram Anggota

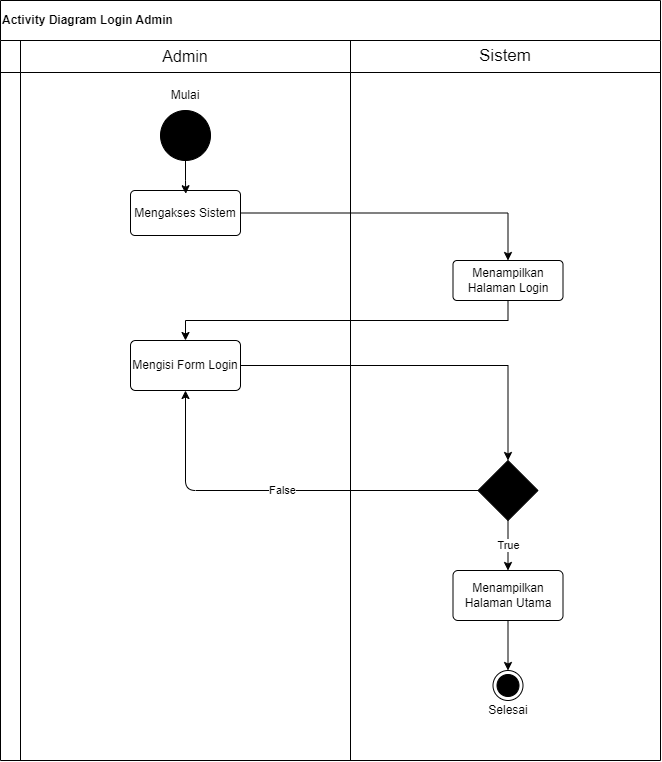
1. Deskripsi Aktor

Tabel 4.2 Deskripsi Aktor

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Admin | Admin merupakan pengguna yang memiliki akses penuh terhadap aplikasi SIKIBE. Admin dapat melakukan berbagai operasi seperti menambah, mengedit, dan menghapus data pengguna, data anggota, data simpanan, data pinjaman, dan data angsuran. Admin juga bertanggung jawab untuk melihat dan mencetak laporan keuangan serta memantau aktivitas keseluruhan koperasi. |
| 2 | Anggota | Anggota merupakan pengguna yang terdaftar dalam koperasi. Anggota dapat mengakses aplikasi untuk melihat informasi simpanan dan pinjaman mereka, mengajukan pinjaman, melakukan pembayaran angsuran, serta mengelola profil pribadi mereka. Anggota juga dapat melihat riwayat transaksi mereka dan menerima notifikasi terkait aktivitas koperasi. |

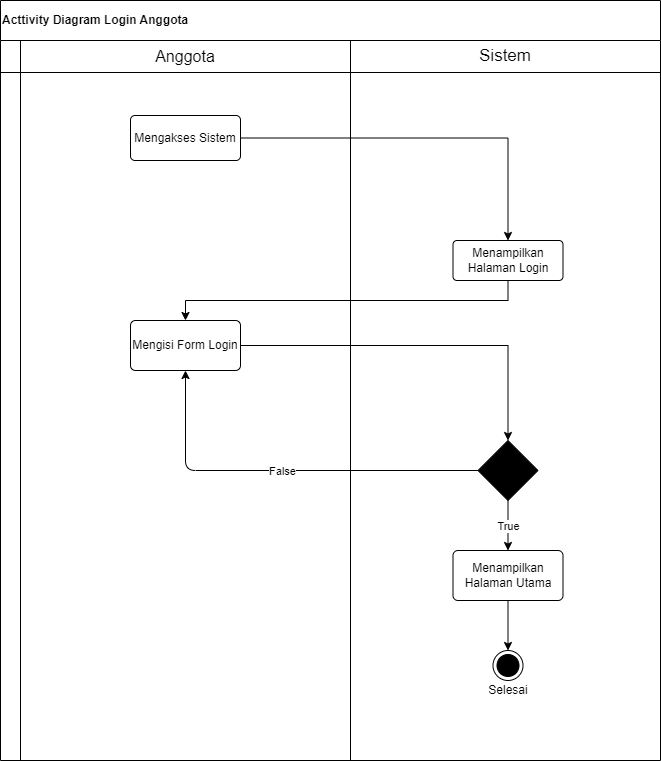
1. *Activity* Diagram

*Activity* Diagram menggambarkan urutan aktivitas proses pada sebuah sistem. Berikut adalah *Activity* diagram pada aplikasi SIKIBE.

1. *Activity* Diagram *Login* Admin

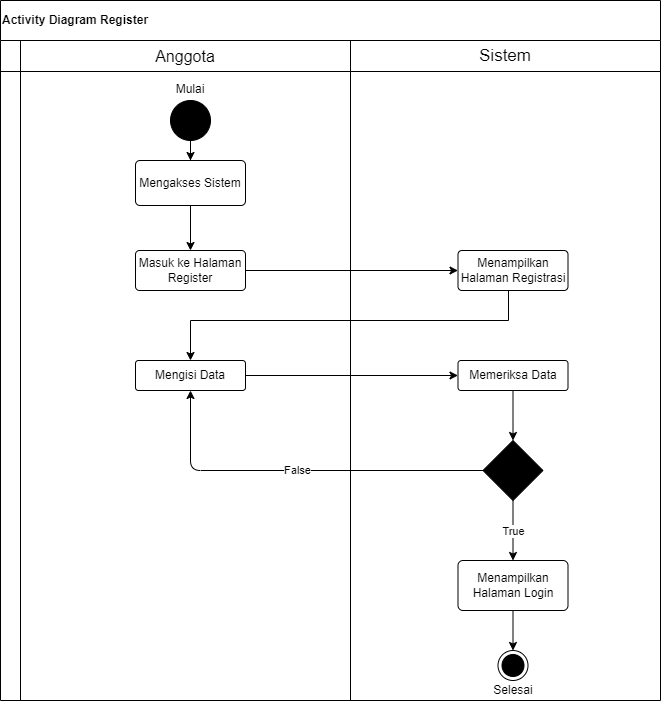
Gambar . *Activity* Diagram *Login* Admin

Proses login admin melibatkan admin memasukkan *username* dan *password* yang kemudian diverifikasi oleh sistem. Jika data valid, admin berhasil *login*; jika tidak, sistem menampilkan pesan *error*.

1. *Activity* Diagram *Login* Anggota

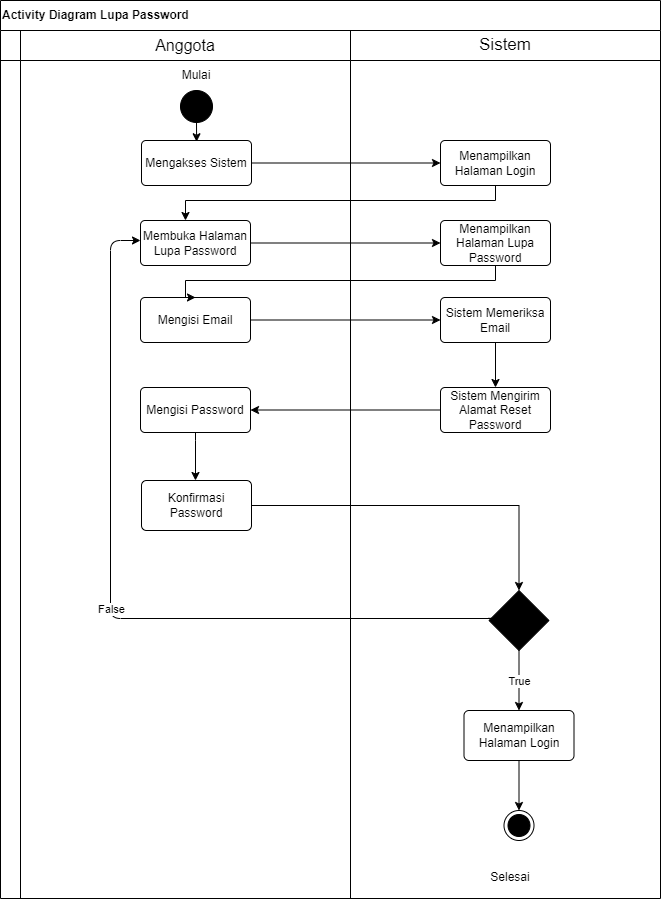
Gambar . *Activity* Diagram *Login* Anggota

Proses *login* anggota mirip dengan *login* admin, di mana anggota memasukkan *username* dan *password*. Setelah diverifikasi, anggota diberikan akses ke beranda anggota atau ditampilkan pesan *error* jika data tidak valid.

1. *Activity* Diagram *Register*

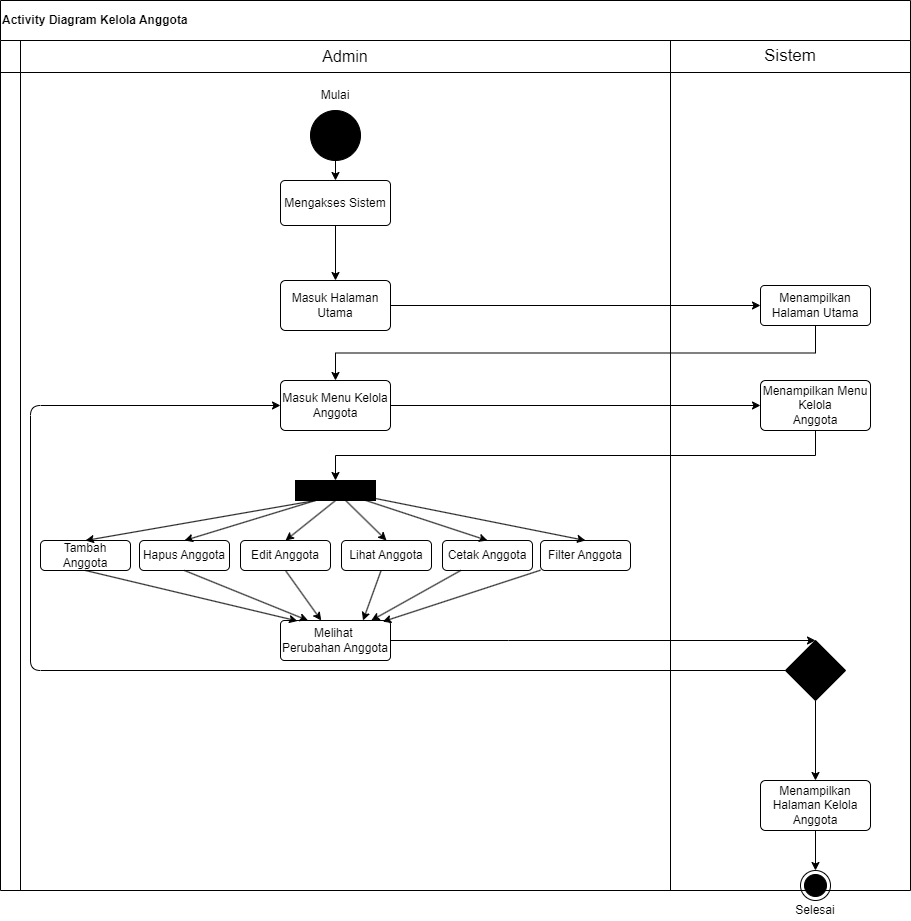
Gambar . *Activity* Diagram *Register*

Proses registrasi memungkinkan pengguna baru untuk membuat akun dengan memasukkan informasi yang diperlukan. Setelah data valid, akun dibuat dan pengguna diarahkan ke halaman *login*.

1. *Activity* Diagram Lupa *Password*

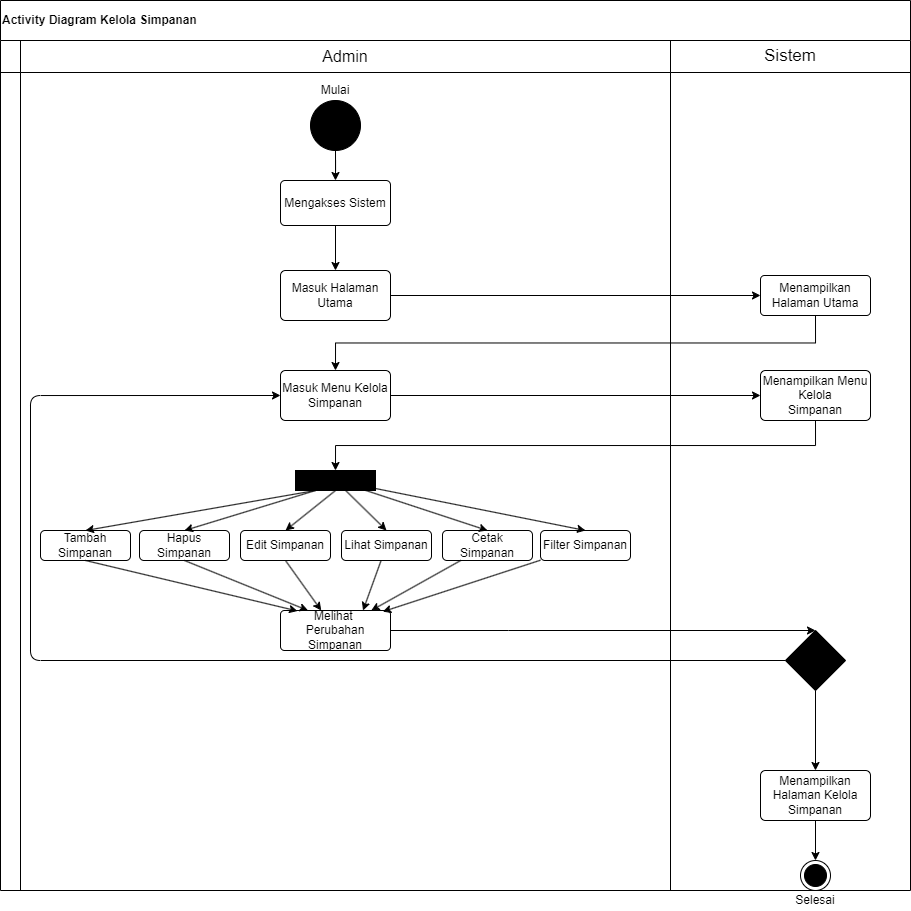
Gambar . *Activity* Diagram Lupa *Password*

Proses lupa *password* melibatkan pengguna memasukkan email terdaftar untuk menerima tautan *reset password*. Setelah email diverifikasi, pengguna dapat mengatur ulang *password* melalui tautan yang diterima.

1. *Activity* Diagram Kelola Anggota

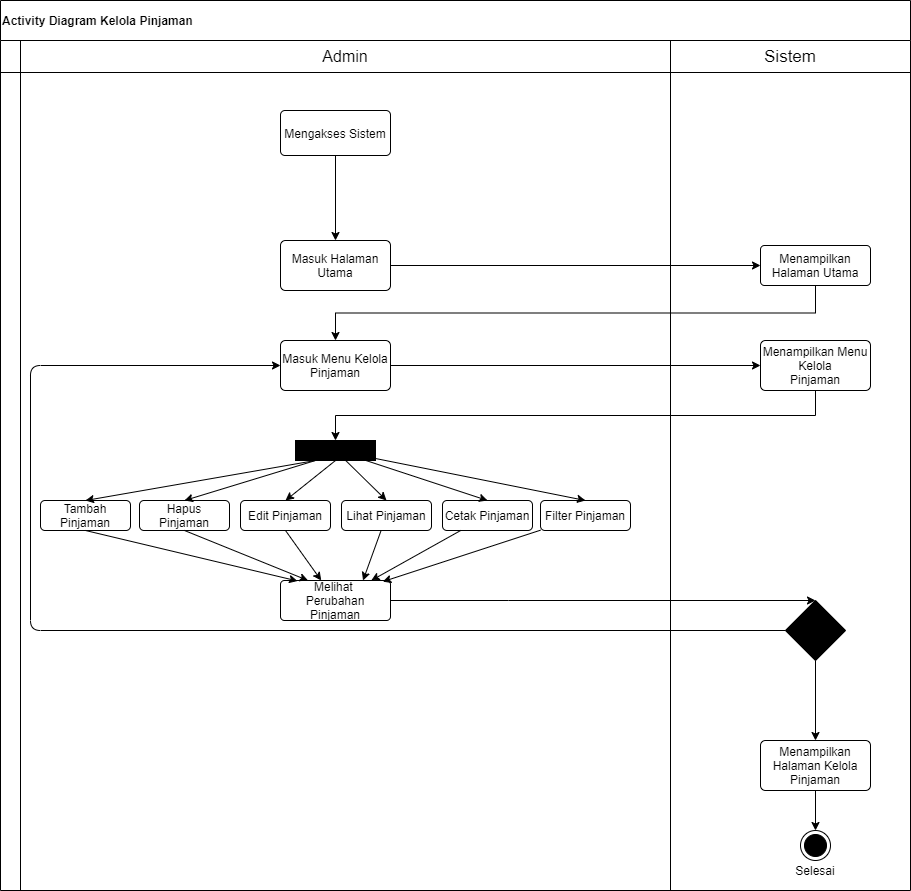
Gambar . *Activity* Diagram Kelola Anggota

Admin dapat mengelola anggota dengan menambah, mengedit, atau menghapus data anggota. Setiap perubahan diverifikasi dan disimpan dalam *database*.

1. *Activity* Diagram Kelola Simpanan

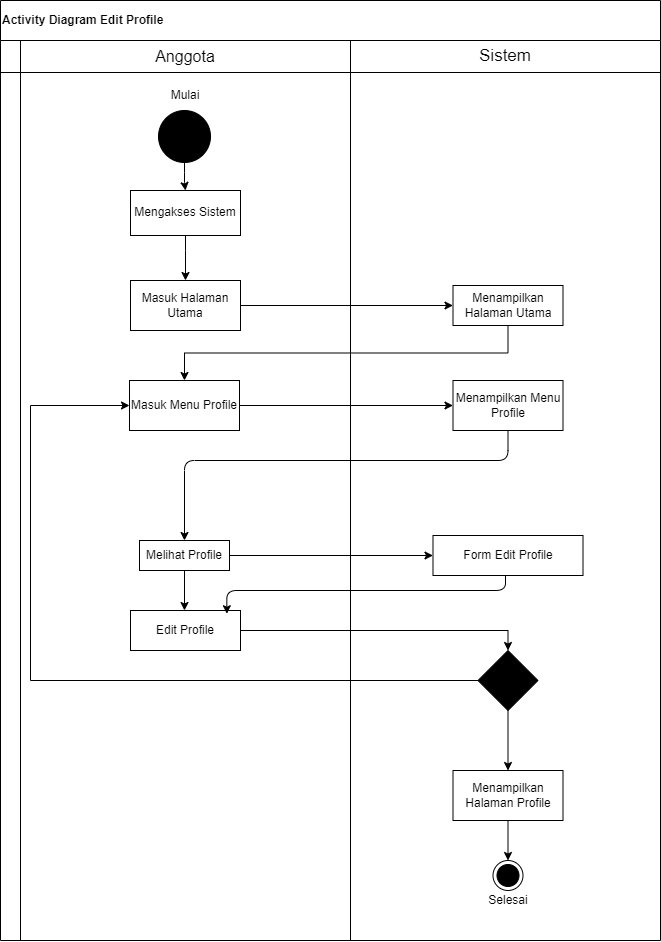
Gambar . *Activity* Diagram Kelola Simpanan

Admin mengelola simpanan anggota dengan menambah, mengedit, atau menghapus data simpanan. Setiap transaksi diverifikasi dan disimpan dalam *database*.

1. *Activity* Diagram Kelola Pinjaman

Gambar . *Activity* Diagram Kelola Pinjaman

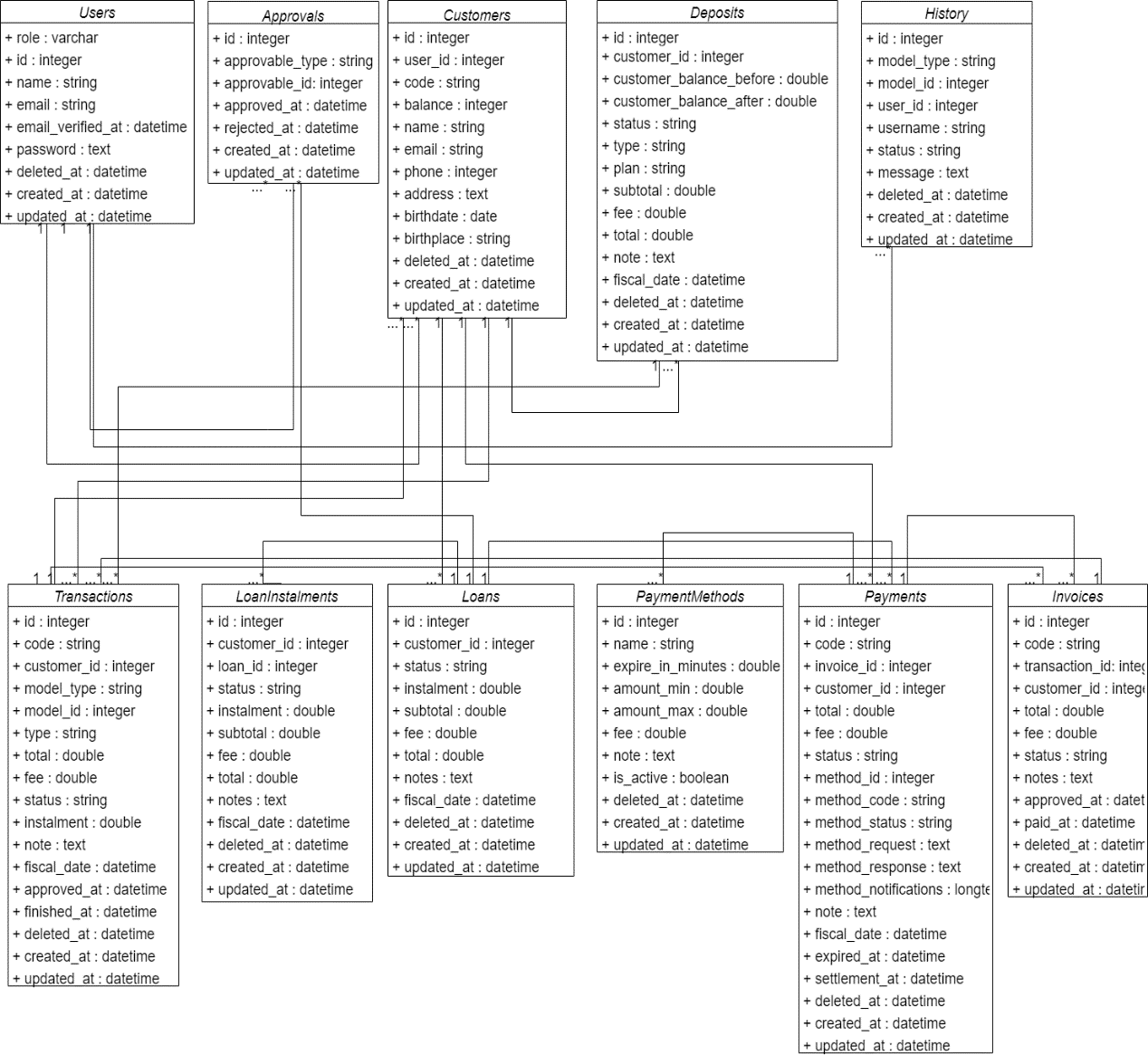
Admin mengelola pinjaman anggota dengan menambah, mengedit, atau menghapus data pinjaman. Setiap transaksi diverifikasi dan disimpan dalam *database*.

1. *Activity* Diagram *Edit* *Profile*

Gambar . *Activity* Diagram *Edit Profile*

Pengguna dapat mengedit *profile* mereka dengan memasukkan informasi baru yang valid. Setelah perubahan diverifikasi, *profile* diperbarui dalam *database*.

1. *Class* Diagram

*Class* Diagram pada aplikasi SIKIBE (Sistem Informasi Koperasi BE7)

Gambar . *Class* DiagramStruktur Tabel

Struktur Tabel-tabel yang terdapat dalam basis data yang digunakan dalam aplikasi SIKIBE adalah sebagai berikut:

1. Perancangan tabel users

Tabel 4.3 Tabel *Users*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik pengguna |
| role | varchar | 50 | - | Peran pengguna |
| name | varchar | 50 | - | Nama pengguna |
| email | email | 50 | - | Email pengguna |
| email\_verified\_at | date\_time | timestamp | - | Waktu verifikasi email |
| password | text | 50 | - | Kata sandi pengguna |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

1. Perancangan tabel approvals

Tabel 4.4 Tabel *Approvals*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik persetujuan |
| approvable\_type | varchar | 50 | - | Tipe yang disetujui |
| approvable\_id | integer | 10 | *Foreign key* | Id yang disetujui |
| approved\_at | date\_time | timestamp | - | Waktu persetujuan |
| rejected\_at | date\_time | timestamp | - | Waktu penolakan |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

1. Perancangan tabel customers

Tabel 4.5 Tabel *Customers*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik pelanggan |
| user\_id | integer | 10 | *Foreign key* | Id pengguna |
| code | varchar | 50 | - | Kode pelanggan |
| balance | integer | 11 | - | Saldo pelanggan |
| name | varchar | 50 | - | Nama pelanggan |
| email | varchar | 50 | - | Email pelanggan |
| phone | int | 11 | - | Nomor telepon pelanggan |
| address | text | 50 | - | Alamat pelanggan |
| birthdate | date | time | - | Tanggal lahir pelanggan |
| birthplace | varchar | 50 | - | Tempat lahir pelanggan |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

1. Perancangan tabel deposits

Tabel 4.6 Tabel *Deposits*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik deposit |
| customer\_id | integer | 11 | *Foreign key* | Id pelanggan |
| customer\_balance\_before | double | 15,2 | - | Saldo sebelum deposit |
| customer\_balance\_after | double | 15,2 | - | Saldo setelah deposit |
| status | varchar | 50 | - | Status deposit |
| type | varchar | 50 | - | Tipe deposit |
| plan | varchar | 50 | - | Rencana deposit |
| subtotal | double | 15,2 | - | Subtotal deposit |
| fee | double | 15,2 | - | Biaya deposit |
| total | double | 15,2 | - | Total deposit |
| notes | text | 50 | - | Catatan deposit |
| fiscal\_date | date\_time | timestamp | - | Tanggal fiskal |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

1. Perancangan tabel histories

Tabel 4.7 Tabel *Histories*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik riwayat |
| model\_type | varchar | 50 | - | Tipe model |
| model\_id | integer | 10 | *Foreign key* | Id model |
| user\_id | integer | 10 | *Foreign key* | Id pengguna |
| username | varchar | 50 | - | Nama pengguna |
| status | varchar | 50 | - | Status riwayat |
| message | text | 50 | - | Pesan riwayat |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

1. Perancangan tabel invoices

Tabel 4.8 Tabel *Invoices*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik invoice |
| code | varchar | 50 | - | Kode invoice |
| transaction\_id | integer | 10 | *Foreign key* | Id transaksi |
| customer\_id | integer | 10 | *Foreign key* | Id pelanggan |
| total | double | 15,2 | - | Total invoice |
| fee | double | 15,2 | - | Biaya invoice |
| status | varchar | 50 | - | Status invoice |
| notes | text | 50 | - | Catatan invoice |
| approved\_at | date\_time | timestamp | - | Waktu persetujuan invoice |
| paid\_at | date\_time | timestamp | - | Waktu pembayaran invoice |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

1. Perancangan tabel loans

Tabel 4.9 Tabel *Loans*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik pinjaman |
| customer\_id | integer | 11 | *Foreign key* | Id pelanggan |
| status | varchar | 50 | - | Status pinjaman |
| instalment | double | 15,2 | - | Angsuran pinjaman |
| subtotal | double | 15,2 | - | Subtotal pinjaman |
| fee | double | 15,2 | - | Biaya pinjaman |
| total | double | 15,2 | - | Total pinjaman |
| notes | text | 50 | - | Catatan pinjaman |
| fiscal\_date | date\_time | timestamp | - | Tanggal fiskal |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

1. Perancangan tabel loan\_instalments

Tabel 4.10 Tabel *Loan\_Instalments*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik angsuran pinjaman |
| customer\_id | integer | 11 | *Foreign key* | Id pelanggan |
| loan\_id | integer | 11 | *Foreign key* | Id pinjaman |
| status | varchar | 50 | - | Status angsuran |
| instalment | double | 15,2 | - | Jumlah angsuran |
| subtotal | double | 15,2 | - | Subtotal angsuran |
| fee | double | 15,2 | - | Biaya angsuran |
| total | double | 15,2 | - | Total angsuran |
| notes | text | 50 | - | Catatan angsuran |
| fiscal\_date | date\_time | timestamp | - | Tanggal fiskal |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

1. Perancangan tabel payments

Tabel 4.11 Tabel *Payments*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik pembayaran |
| code | varchar | 50 | *Foreign key* | Kode pembayaran |
| invoice\_id | integer | 10 | *Foreign key* | Id invoice |
| customer\_id | integer | 10 | *Foreign key* | Id pelanggan |
| total | double | 15,2 | - | Total pembayaran |
| fee | double | 15,2 | - | Biaya pembayaran |
| status | varchar | 50 | - | Status pembayaran |
| method\_id | integer | 10 | *Foreign key* | Id metode pembayaran |
| method\_code | varchar | 50 | - | Kode metode pembayaran |
| method\_status | varchar | 50 | - | Status metode pembayaran |
| method\_request | text | 50 | - | Permintaan metode pembayaran |
| method\_response | text | 50 | - | Respon metode pembayaran |
| method\_notifications | longtext | 50 | - | Notifikasi metode pembayaran |
| note | text | 50 | - | Catatan pembayaran |
| fiscal\_date | date\_time | timestamp | - | Tanggal fiskal |
| expired\_at | date\_time | timestamp | - | Waktu kadaluwarsa pembayaran |
| settelment\_at | date\_time | timestamp | - | Waktu penyelesaian pembayaran |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

1. Perancangan tabel payment\_methods

Tabel 4.12 Tabel *Payment\_Methods*

| **Field** | **Type** | **Size** | **Indeks** | | **Deskripsi** |
| --- | --- | --- | --- | --- | --- |
| id | integer | 20 | | *Primary key* | Id unik metode pembayaran |
| name | varchar | 50 | | - | Nama metode pembayaran |
| expire\_in\_minutes | double | 15,2 | | - | Waktu kadluwarsa (menit) |
| amount\_min | double | 15,2 | | - | Jumlah minimum |
| amount\_max | double | 15,2 | | - | Jumlah maksimum |
| fee | double | 15,2 | | - | Biaya metode pembayaran |
| note | text | 50 | | - | Catatan metode pembayaran |
| is\_active | boolean | 1 | | - | Status aktif metode pembayaran |
| deleted\_at | date\_time | timestamp | | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | | - | Waktu pembaruan terakhir |

1. Perancangan tabel transactions

Tabel 4.13 Tabel *Transactions*

| **Field** | **Type** | **Size** | **Indeks** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| id | integer | 20 | *Primary key* | Id unik transaksi |
| code | varchar | 50 | *Foreign key* | Kode transaksi |
| customer\_id | integer | 10 | *Foreign key* | Id pelanggan |
| model\_type | varchar | 50 | - | Tipe model |
| model\_id | integer | 11 | *Foreign key* | Id model |
| type | varchar | 50 | - | Tipe transaksi |
| total | double | 15,2 | - | Total transaksi |
| fee | double | 15,2 | - | Biaya transaksi |
| status | varchar | 50 | - | Status transaksi |
| instalment | double | 15,2 | - | Angsuran transaksi |
| note | text | 50 | - | Catatan transaksi |
| fiscal\_date | date\_time | timestamp | - | Tanggal fiskal |
| approved\_at | date\_time | timestamp | - | Waktu persetujuan transaksi |
| finished\_at | date\_time | timestamp | - | Waktu penyelesaian transaksi |
| deleted\_at | date\_time | timestamp | - | Waktu penghapusan data |
| created\_at | date\_time | timestamp | - | Waktu pembuatan data |
| updated\_at | date\_time | timestamp | - | Waktu pembaruan terakhir |

### Desain sistem

1. *User Interface* Halaman *Login Admin*



Gambar . *User Interface* Halaman *Login Admin*

Sebelum memasuki menu aplikasi SIKIBE, admin diharuskan untuk melakukan login dengan memasukkan *username* dan *password* yang terdaftar di aplikasi SIKIBE.

1. *User Interface* Halaman Login *User*



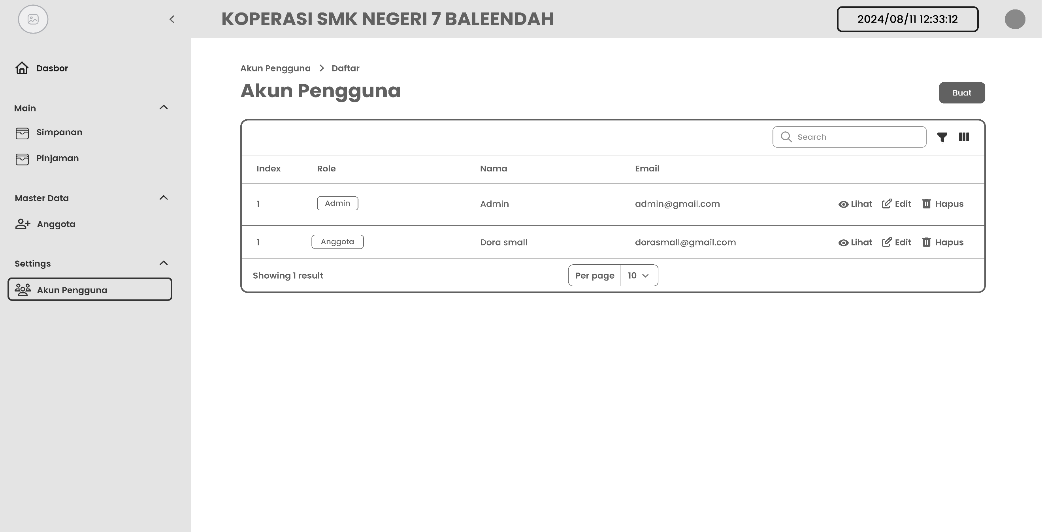
Gambar 4.13 *User Interface* Halaman *Login User*

Sebelum memasuki menu aplikasi SIKIBE, *user* diharuskan untuk melakukan login dengan memasukkan *username* dan *password* yang terdaftar di aplikasi SIKIBE.

1. *User Interface* Halaman Beranda

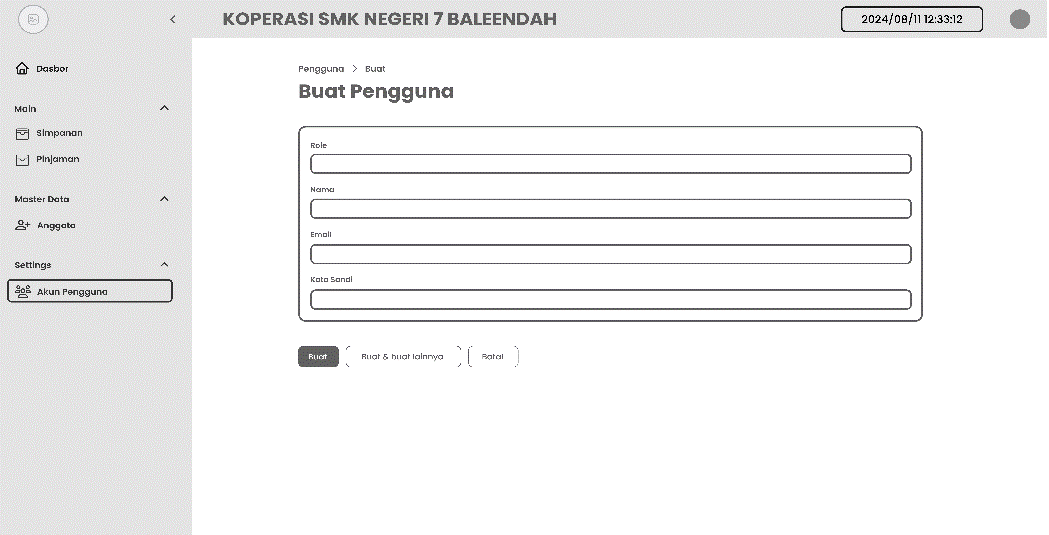
Gambar .14 *User Interface* Halaman Beranda

Setelah admin atau user mengisi *form login* dengan benar, tampilan pertama yang muncul adalah halaman beranda sebagai halaman utama. Pada halaman beranda aplikasi SIKIBE, pengguna dapat melihat ringkasan informasi penting seperti total simpanan, pinjaman, notifikasi terbaru, dan statistik koperasi.

1. *User Interface* Halaman Pengguna

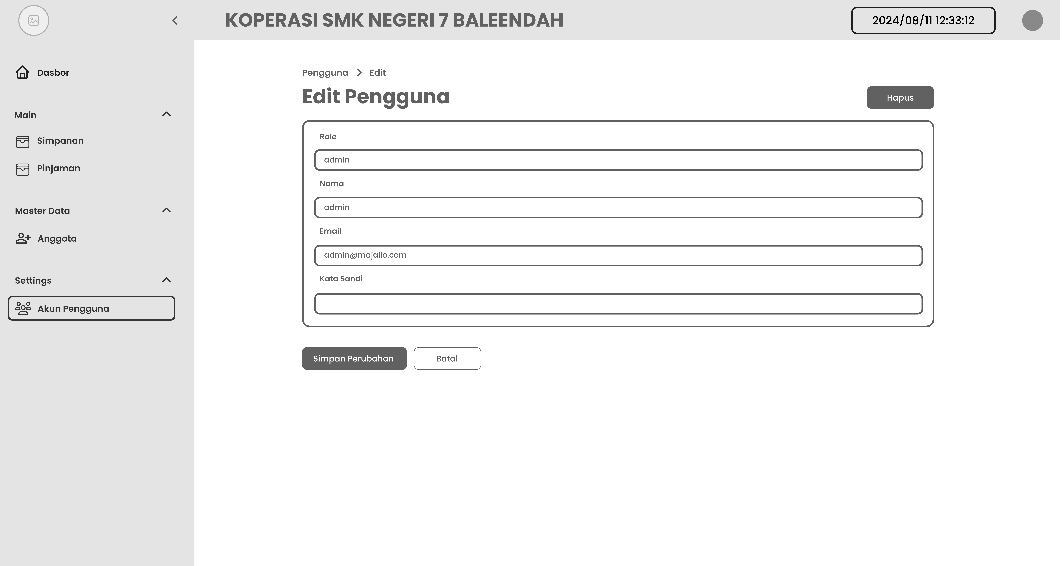
Gambar .15 *User Interface* Halaman Pengguna

Pada desain halaman pengguna, admin dapat menambah, mengedit, dan menghapus data pengguna seperti admin dan anggota koperasi. Halaman ini memudahkan pengelolaan data pengguna secara efektif dan efisien.

1. *User Interface* Halaman Tambah Pengguna

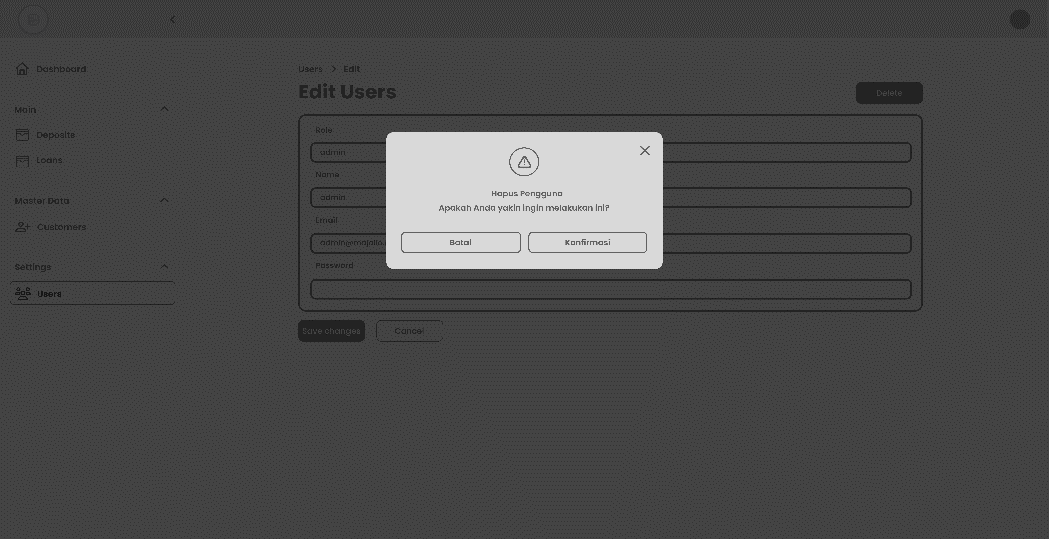
Gambar .16 *User Interface* Halaman Tambah Pengguna

Pada desain halaman tambah pengguna, admin dapat melakukan tambah pegguna dengan cara mengklik tombol tambah pengguna kemudian sistem akan menampilkan form tambah pengguna.

1. *User Interface* Halaman Ubah Pengguna

Gambar .17 *User Interface* Halaman Ubah Pengguna

Pada desain halaman ubah pengguna, admin dapat melakukan ubah pegguna dengan cara mengklik tombol ubah kemudian sistem akan menampilkan form ubah pengguna.

1. *User Interface* Halaman Hapus Pengguna

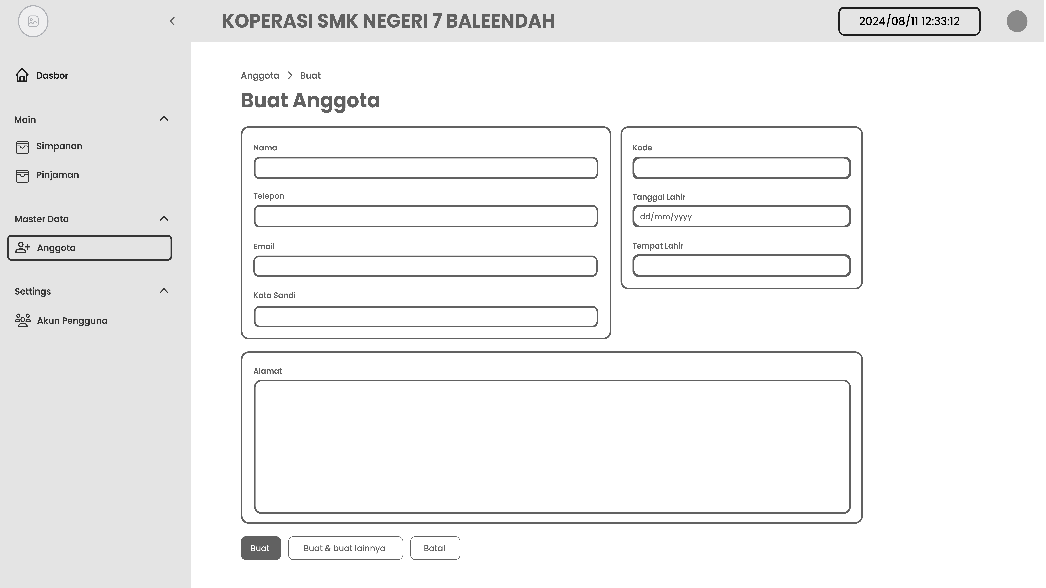
Gambar .18 *User Interface* Halaman Hapus Pengguna

Pada desain halaman hapus pengguna, admin dapat melakukan hapus pegguna dengan cara mengklik tombol hapus kemudian sistem akan menampilkan peringatan sekaligus validasi untuk pengapusan data pengguna.

1. *User Interface* Halaman Anggota

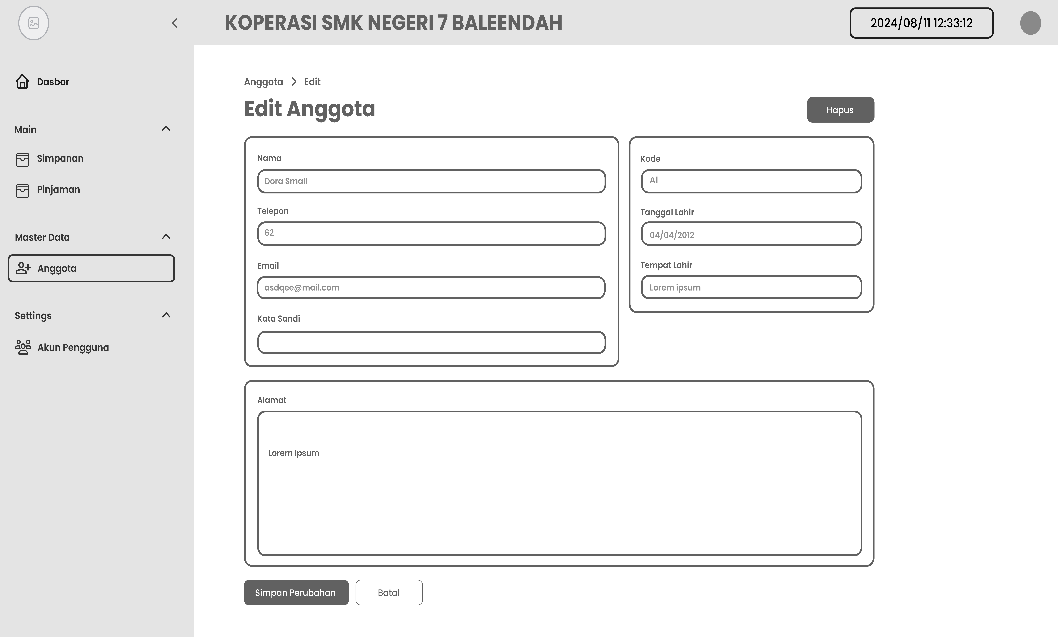
Gambar . *User Interface* Halaman Anggota

Pada desain halaman anggota, admin dapat menambah anggota baru, melihat detail anggota, mengubah data anggota, dan menghapus data anggota. Halaman ini mencakup informasi lengkap mengenai setiap anggota, termasuk profil, riwayat transaksi, dan status keanggotaan.

1. *User Interface* Halaman Tambah Anggota

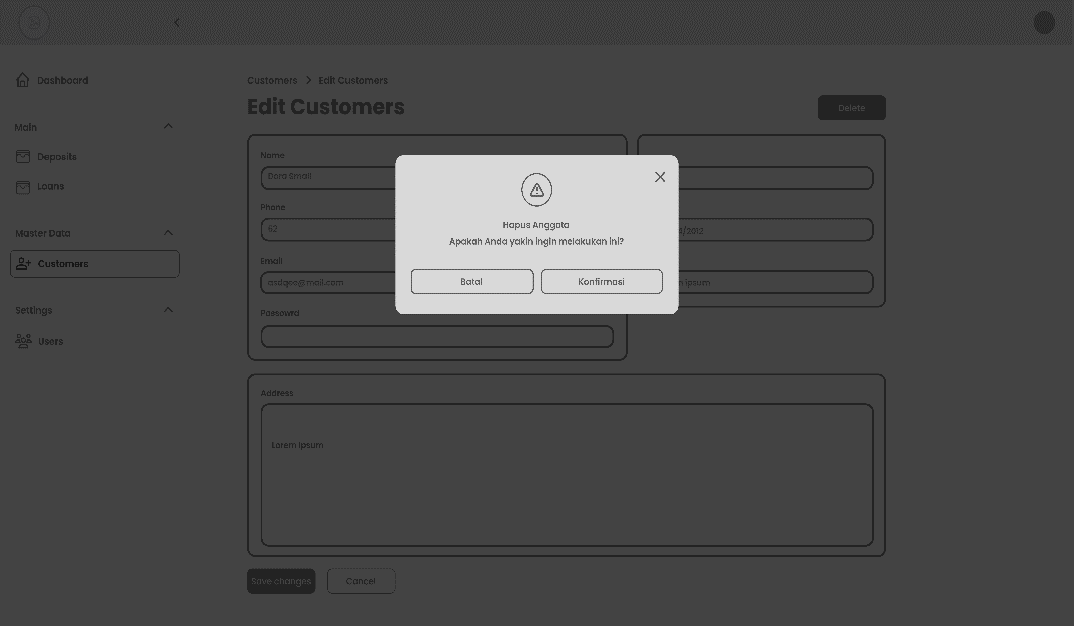
Gambar .20 *User Interface* Halaman Tambah Anggota

Pada desain halaman tambah anggota, admin dapat melakukan tambah anggota dengan cara mengklik tombol tambah anggota kemudian sistem akan menampilkan form tambah anggota.

1. *User Interface* Halaman Ubah Anggota

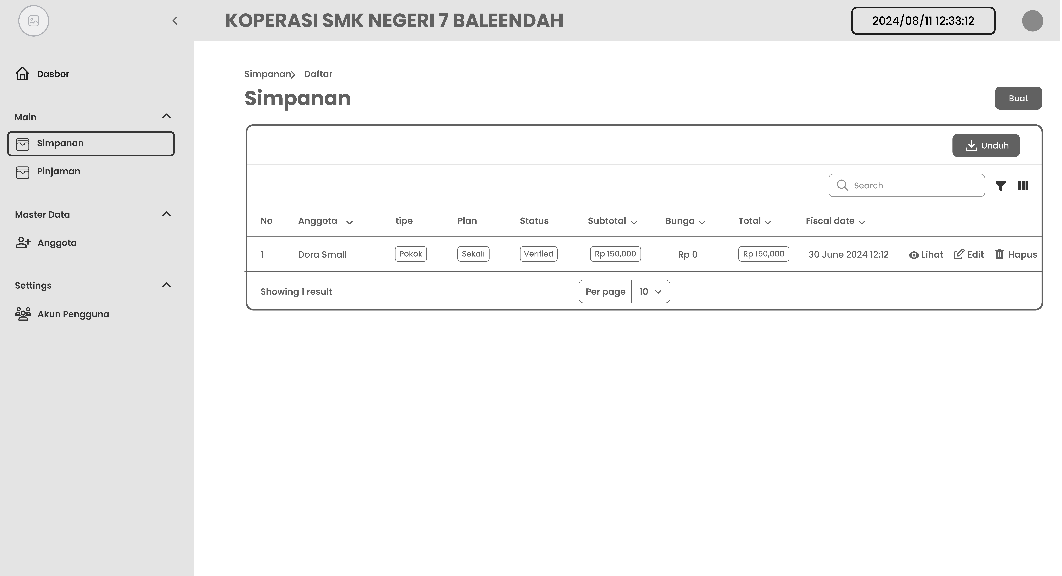
Gambar . *User Interface* Halaman Ubah Pengguna

Pada desain halaman ubah anggota, admin dapat melakukan ubah anggota dengan cara mengklik tombol ubah kemudian sistem akan menampilkan form ubah anggota.

1. *User Interface* Halaman Hapus Anggota

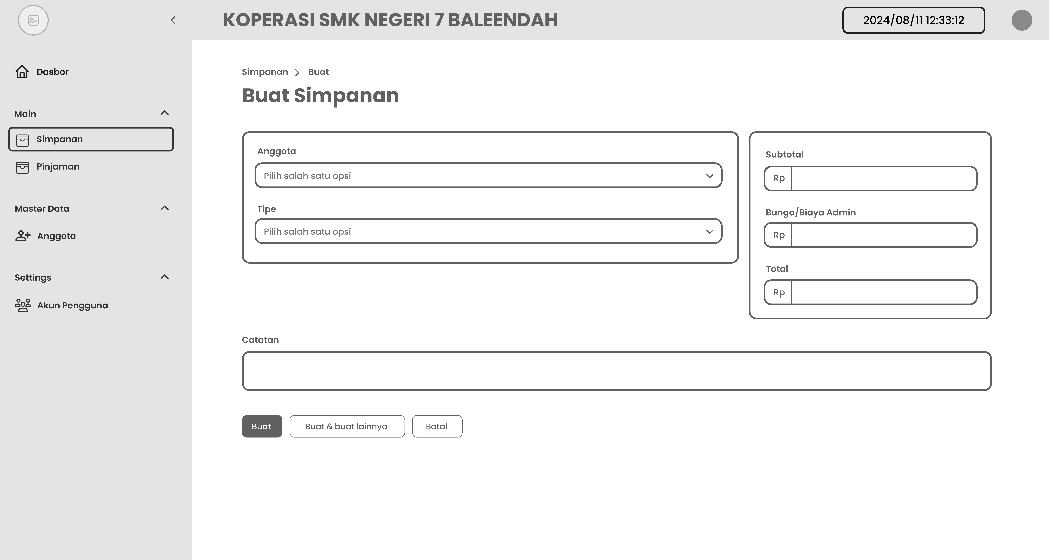
Gambar .22 *User Interface* Halaman Hapus Anggota

Pada desain halaman hapus anggota, admin dapat melakukan hapus anggota dengan cara mengklik tombol hapus kemudian sistem akan menampilkan peringatan sekaligus validasi untuk pengapusan data anggota.

1. *User Interface* Halaman Simpanan

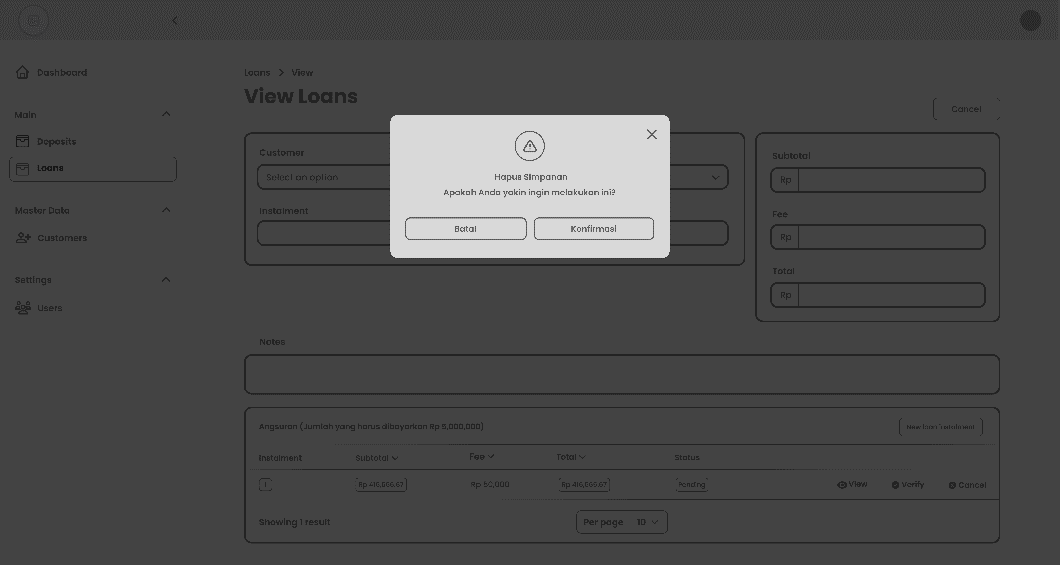
Gambar .23 *User Interface* Halaman Simpanan

Pada desain halaman simpan, admin dapat melakukan pencatatan simpanan baru, melihat detail simpanan, dan menghapus data simpanan. Anggota koperasi juga dapat melihat informasi mengenai simpanan mereka secara rinci di halaman ini.

1. *User Interface* Halaman Buat Simpanan

Gambar .24 *User Interface* Halaman Buat Simpanan

Pada desain halaman buat simpan, admin dapat melakukan buat simpan dengan cara mengklik tombol buat simpan kemudian sistem akan menampilkan form buat simpan.

1. *User Interface* Halaman Batalkan Simpana

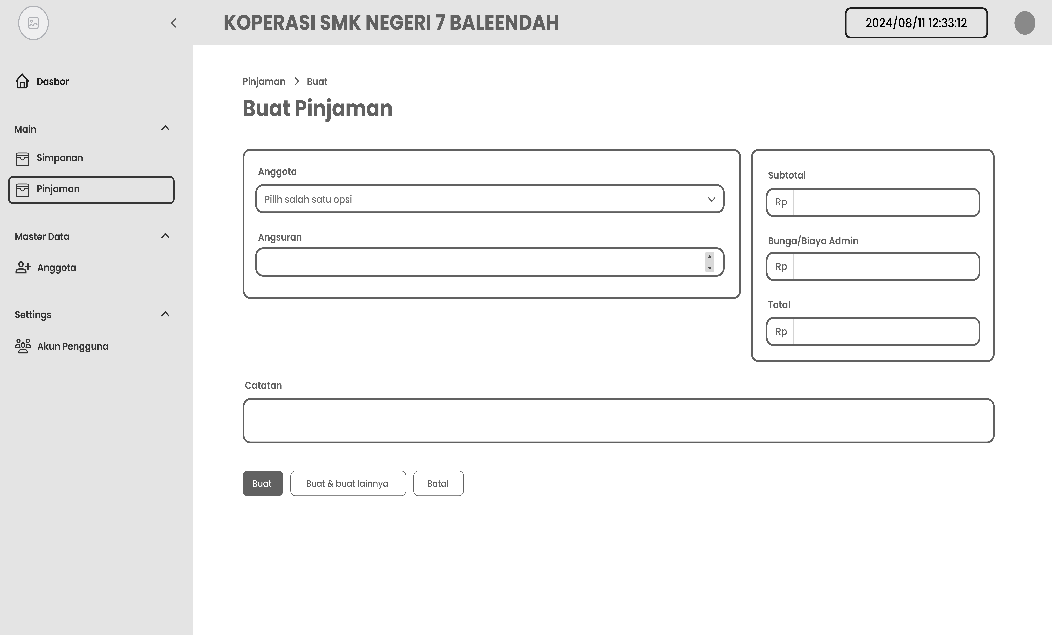
Gambar . *User Interface* Halaman Batalkan Simpanan

Pada desain halaman batalkan simpan, admin dapat melakukan batalkan simpan dengan cara mengklik tombol batalkan simpan kemudian sistem akan menampilkan peringatan sekaligus validasi untuk pembatalan data simpan.

1. *User Interface* Halaman Pinjaman

Gambar .26 *User Interface* Halaman Pinjaman

Pada desain halaman pinjam, admin dapat mencatat pinjaman baru, melihat detail pinjaman, dan menghapus data pinjaman. Anggota koperasi dapat menggunakan halaman ini untuk mengajukan pinjaman dan memantau status pinjaman mereka.

1. *User Interface* Halaman Buat Pinjaman

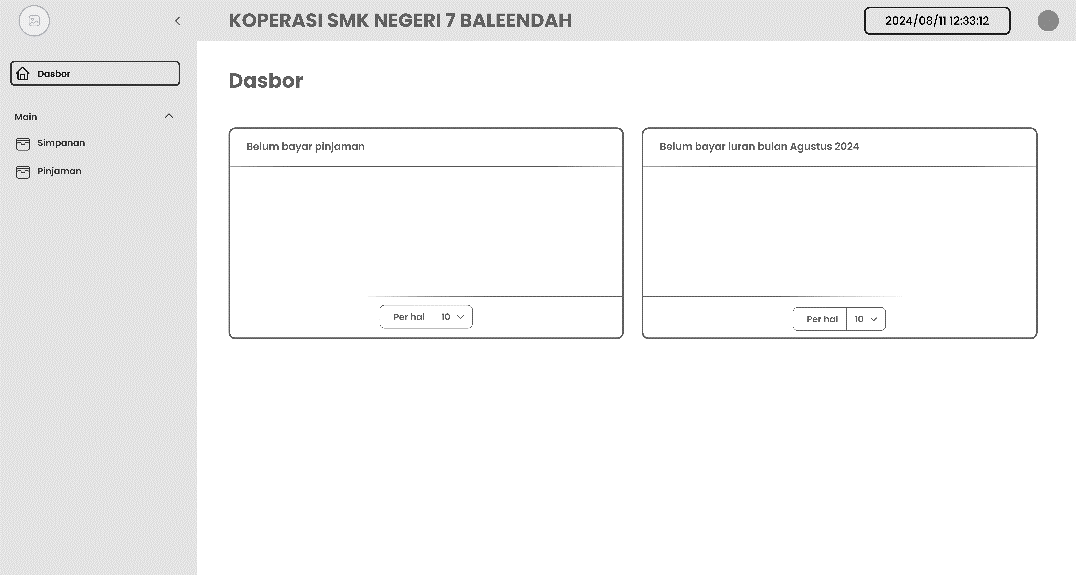
Gambar .27 *User Interface* Halaman Buat Pinjam

Pada desain halaman buat pinjam, admin dapat melakukan buat pinjam dengan cara mengklik tombol buat pinjam kemudian sistem akan menampilkan form buat pinjam.

1. *User Interface* Halaman Batalkan Pinjaman

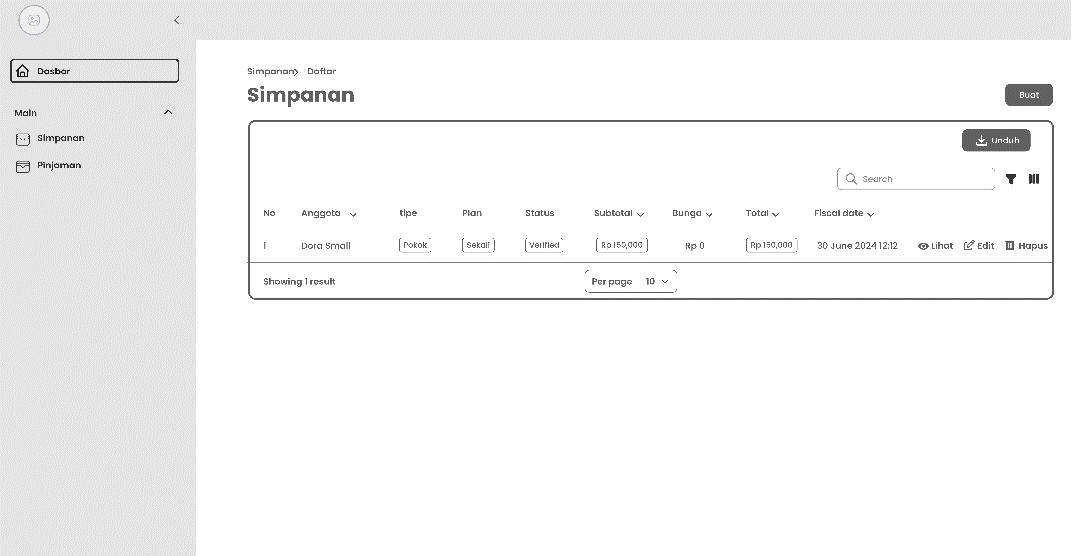
Gambar .28 *User Interface* Halaman Batalkan Pinjam

Pada desain halaman batalkan pinjam, admin dapat melakukan batalkan pinjam dengan cara mengklik tombol batalkan pinjam kemudian sistem akan menampilkan peringatan sekaligus validasi untuk pembatalan data pinjam.

1. Halaman Beranda *User*

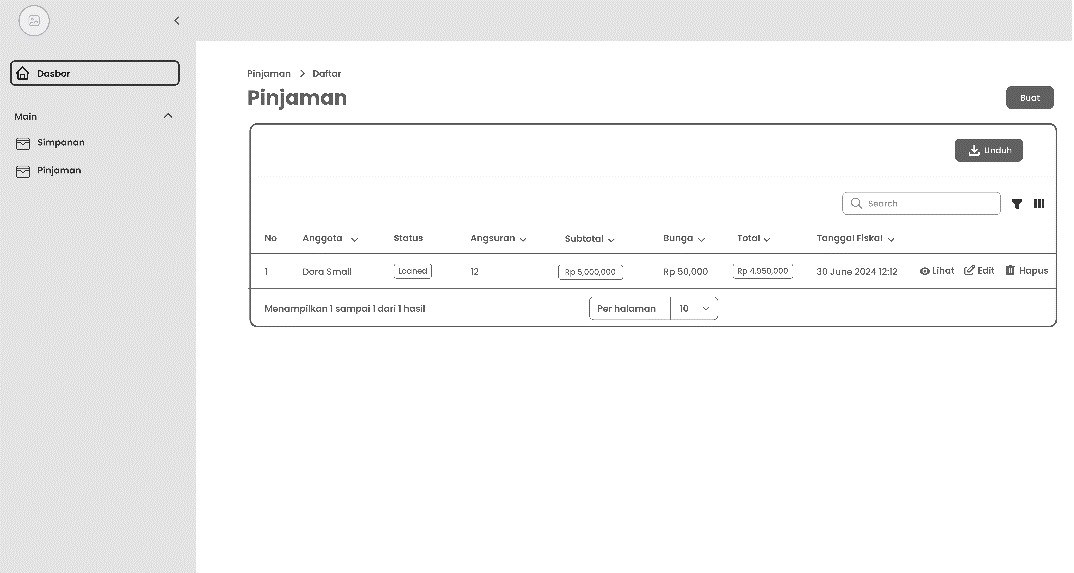
Gambar 4.29 Halaman Beranda *User*

Pada desain halaman Beranda, *User* bisa melihat dan memonitoring transaksi simpanan dan pinjaman

1. Halaman Simpanan

Gambar 4.30 Halaman Simpanan *User*

Pada desain halaman Simpanan, *User* dapat melihat dan memonitoring transaksi di menu simpanan

1. Halaman Pinjaman *User*

Gambar 4.31 Halaman Pinjaman *User*

Pada desain halaman pinjaman, *User* dapat melihat dan memonitoring transaksi di menu pinjaman

# BAB V IMPLEMENTASI DAN PENGUJIAN

## Implementasi

Setelah melaksanakan analisis dan perancangan maka selanjutnya adalah pengimplementasian untuk menjalankan analisis dan perancangan yang sudah dibuat ke dalam bentuk aplikasi.

### *Listing* Program

1. *Listing* Program *login*

|  |
| --- |
| public function authenticate(): ?LoginResponse  {  try {  $this->rateLimit(5);  } catch (TooManyRequestsException $exception) {  Notification::make()  ->title(\_\_('filament-panels::pages/auth/login.notifications.throttled.title', [  'seconds' => $exception->secondsUntilAvailable,  'minutes' => ceil($exception->secondsUntilAvailable / 60),  ]))  ->body(array\_key\_exists('body', \_\_('filament-panels::pages/auth/login.notifications.throttled') ?: []) ? \_\_('filament-panels::pages/auth/login.notifications.throttled.body', [  'seconds' => $exception->secondsUntilAvailable,  'minutes' => ceil($exception->secondsUntilAvailable / 60),  ]) : null)  ->danger()  ->send();  return null;  }  $data = $this->form->getState();  if (! Filament::auth()->attempt($this->getCredentialsFromFormData($data), $data['remember'] ?? false)) {  $this->throwFailureValidationException();  }  $user = Filament::auth()->user();  if (  ($user instanceof FilamentUser) &&  (! $user->canAccessPanel(Filament::getCurrentPanel()))  ) {  Filament::auth()->logout();  $this->throwFailureValidationException();  }  session()->regenerate();  return app(LoginResponse::class);  } |

1. *Listing* Program register

|  |
| --- |
| public function register(): ?RegistrationResponse  {  try {  $this->rateLimit(2);  } catch (TooManyRequestsException $exception) {  Notification::make()  ->title(\_\_('filament-panels::pages/auth/register.notifications.throttled.title', [  'seconds' => $exception->secondsUntilAvailable,  'minutes' => ceil($exception->secondsUntilAvailable / 60),  ]))  ->body(array\_key\_exists('body', \_\_('filament-panels::pages/auth/register.notifications.throttled') ?: []) ? \_\_('filament-panels::pages/auth/register.notifications.throttled.body', [  'seconds' => $exception->secondsUntilAvailable,  'minutes' => ceil($exception->secondsUntilAvailable / 60),  ]) : null)  ->danger()  ->send();  return null;  }  $user = $this->wrapInDatabaseTransaction(function () {  $this->callHook('beforeValidate');  $data = $this->form->getState();  $this->callHook('afterValidate');  $data = $this->mutateFormDataBeforeRegister($data);  $this->callHook('beforeRegister');  $user = $this->handleRegistration($data);  $this->form->model($user)->saveRelationships();  $this->callHook('afterRegister');  return $user;  });  event(new Registered($user));  $this->sendEmailVerificationNotification($user);  Filament::auth()->login($user);  session()->regenerate();  return app(RegistrationResponse::class);  } |

1. *Listing* Program beranda

|  |
| --- |
| class Dashboard extends Page  {  protected static string $routePath = '/';  protected static ?int $navigationSort = -2;  protected static string $view = 'filament-panels::pages.dashboard';  public static function getNavigationLabel(): string  {  return static::$navigationLabel ??  static::$title ??  \_\_('filament-panels::pages/dashboard.title');  } |

1. *Listing* Program pengguna

|  |
| --- |
| class UserResource extends Resource  {  protected static ?string $model = User::class;  protected static ?string $navigationIcon = 'heroicon-o-user-group';  public static function table(Table $table): Table      {          return $table              ->columns([                  Tables\Columns\TextColumn::make('index')->state(                      static function ($livewire, $rowLoop): string {                          return (string) (                              $rowLoop->iteration +                              ($livewire->getTableRecordsPerPage() \* (                                  $livewire->getTablePage() - 1                              ))                          );                      }                  ),                  Tables\Columns\TextColumn::make('role')                      ->badge()                      ->searchable(),                  Tables\Columns\TextColumn::make('name')                      ->searchable(),                  Tables\Columns\TextColumn::make('email')                      ->icon('heroicon-m-envelope')                      ->searchable(),                  Tables\Columns\TextColumn::make('created\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),                  Tables\Columns\TextColumn::make('updated\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),              ])              ->filters([                  //              ])              ->actions([                  Tables\Actions\ViewAction::make(),                  // Tables\Actions\EditAction::make(),              ])              ->bulkActions([              ])              ->emptyStateActions([                  Tables\Actions\CreateAction::make(),              ]);      } |

1. *Listing* Program anggota

|  |
| --- |
| class CustomerResource extends Resource  {  protected static ?string $model = Customer::class;  protected static ?string $navigationIcon = 'heroicon-o-user-plus';  public static function table(Table $table): Table  {  return $table  ->columns([  Tables\Columns\TextColumn::make('no')  ->rowIndex(),  Tables\Columns\TextColumn::make('name')  ->description(fn ($record) => str($record?->code)->limit(20))  ->wrap()  ->searchable(),  Tables\Columns\TextColumn::make('email')  ->searchable(),  Tables\Columns\TextColumn::make('phone')  ->numeric()  ->searchable(),  Tables\Columns\TextColumn::make('total\_deposits')  ->prefix('Rp ')  ->badge()  ->color('success')  ->numeric(),  Tables\Columns\TextColumn::make('total\_loans')  ->prefix('Rp ')  ->badge()  ->color('danger')  ->numeric(),  Tables\Columns\TextColumn::make('birthdate')  ->date()  ->toggleable(isToggledHiddenByDefault: true)  ->sortable(),  Tables\Columns\TextColumn::make('birthplace')  ->toggleable(isToggledHiddenByDefault: true)  ->searchable(),  Tables\Columns\TextColumn::make('deleted\_at')  ->dateTime()  ->sortable()  ->toggleable(isToggledHiddenByDefault: true),  Tables\Columns\TextColumn::make('created\_at')  ->dateTime()  ->sortable()  ->toggleable(isToggledHiddenByDefault: true),  Tables\Columns\TextColumn::make('updated\_at')  ->dateTime()  ->sortable()  ->toggleable(isToggledHiddenByDefault: true),  ])  ->filters([  Tables\Filters\TrashedFilter::make(),  ])  ->actions([  Tables\Actions\ViewAction::make(),  // Tables\Actions\EditAction::make(),  ])  ->bulkActions([]);  } |

1. *Listing* Program simpan

|  |
| --- |
| class DepositResource extends Resource  {  protected static ?string $model = Deposit::class;  protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';  public static function table(Table $table): Table  {  return $table  ->defaultSort('fiscal\_date', 'desc')  ->columns([  Tables\Columns\TextColumn::make('no')  ->rowIndex(),  Tables\Columns\TextColumn::make('customer.name')  ->numeric()  ->sortable(),  Tables\Columns\TextColumn::make('type')  ->badge()  ->tooltip(fn ($record): string => match ($record->type) {  DepositType::Pokok => 'Iuran wajib anggota 1x',  DepositType::Wajib => 'Setor wajib anggota',  DepositType::Sukarela => 'Setor simpanan',  })  ->searchable(),  Tables\Columns\TextColumn::make('plan')  ->badge()  ->searchable(),  Tables\Columns\TextColumn::make('status')  ->badge()  ->searchable(),  Tables\Columns\TextColumn::make('subtotal')  ->numeric()  ->badge()  ->tooltip('Total nilai masuk')  ->color('success')  ->prefix('Rp ')  ->sortable(),  Tables\Columns\TextColumn::make('fee')  ->numeric()  ->color('danger')  ->prefix('Rp ')  ->sortable(),  Tables\Columns\TextColumn::make('total')  ->prefix('Rp ')  ->numeric()  ->badge()  ->color('warning')  ->tooltip('Total tagihan harus dibayar')  ->sortable(),  Tables\Columns\TextColumn::make('fiscal\_date')  ->dateTime('d F Y H:i')  ->sortable()  ->toggleable(isToggledHiddenByDefault: false),  Tables\Columns\TextColumn::make('created\_at')  ->dateTime()  ->sortable()  ->toggleable(isToggledHiddenByDefault: true),  Tables\Columns\TextColumn::make('updated\_at')  ->dateTime()  ->sortable()  ->toggleable(isToggledHiddenByDefault: true),  Tables\Columns\TextColumn::make('deleted\_at')  ->dateTime()  ->sortable()  ->toggleable(isToggledHiddenByDefault: true),  ])  ->filters([  Tables\Filters\TrashedFilter::make(),  Tables\Filters\SelectFilter::make('type')  ->searchable()  ->preload()  ->options(fn () => DepositType::class),  Tables\Filters\SelectFilter::make('plan')  ->searchable()  ->preload()  ->options(fn () => DepositPlan::class),  Tables\Filters\SelectFilter::make('status')  ->searchable()  ->preload()  ->options(fn () => DepositStatus::class),  ])  ->actions([  Tables\Actions\ViewAction::make(),  ])  ->bulkActions([]);  } |

1. *Listing* Program pinjam

|  |
| --- |
| class LoanResource extends Resource  {  protected static ?string $model = Loan::class;  protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';  public static function table(Table $table): Table  {  return $table  ->defaultSort('fiscal\_date', 'desc')  ->columns([  Tables\Columns\TextColumn::make('no')  ->rowIndex(),  Tables\Columns\TextColumn::make('customer.name')  ->numeric()  ->sortable(),  Tables\Columns\TextColumn::make('status')  ->badge()  ->searchable(),  Tables\Columns\TextColumn::make('instalment')  ->numeric()  ->sortable(),  Tables\Columns\TextColumn::make('subtotal')  ->prefix('Rp ')  ->numeric()  ->badge()  ->color('success')  ->tooltip('Total pinjaman')  ->sortable(),  Tables\Columns\TextColumn::make('fee')  ->prefix('Rp ')  ->numeric()  ->color('danger')  ->sortable(),  Tables\Columns\TextColumn::make('total')  ->prefix('Rp ')  ->numeric()  ->badge()  ->color('warning')  ->tooltip('Total tagihan harus dibayar')  ->sortable(),  Tables\Columns\TextColumn::make('fiscal\_date')  ->dateTime('d F Y H:i')  ->sortable()  ->toggleable(isToggledHiddenByDefault: false),  Tables\Columns\TextColumn::make('created\_at')  ->dateTime()  ->sortable()  ->toggleable(isToggledHiddenByDefault: true),  Tables\Columns\TextColumn::make('updated\_at')  ->dateTime()  ->sortable()  ->toggleable(isToggledHiddenByDefault: true),  Tables\Columns\TextColumn::make('deleted\_at')  ->dateTime()  ->sortable()  ->toggleable(isToggledHiddenByDefault: true),  ])  ->filters([  Tables\Filters\TrashedFilter::make(),  Tables\Filters\SelectFilter::make('status')  ->searchable()  ->preload()  ->options(fn () => LoanStatus::class),  ])  ->actions([  Tables\Actions\ViewAction::make(),  ])  ->bulkActions([]);  } |

Souce Code lengkap ada di Lampiran 7

### Implementasi Sistem

Implementasi sistem adalah tahap penerapan sistem yang akan dilakukan jika sistem telah disetujui termasuk program yang telah dibuat pada tahap perancangan sistem agar siap untuk dioperasikan. Adapun waktu dan tempat penerapan sistem yang sudah dibuat sebagai berikut:

1. Waktu dan Tempat Implementasi

Tempat : SMKN 7 Baleendah

Alamat : Jalan Siliwangi KM.15 RT 04 RW 17 Desa Manggahang Kecamatan Baleendah Kabupaten bandung jawa barat

### Spesifikasi Sistem

Spesifikasi sistem akan menjelaskan tentang spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam pengimplementasian aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Spesifikasi Perangkat Keras

Di bawah ini merupakan spesifikasi perangkat keras yang digunakan sebagai berikut:

Tabel 5.1 Spesifikasi Perangkat Keras

|  |  |
| --- | --- |
| Processor | AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx (4 CPUs), 2.6GHz |
| RAM | 4 GB |
| SSD | 128 GB |

1. Spesifikasi Perangkat Lunak

Di bawah ini merupakan spesifikasi perangkat lunak yang digunakan sebagai berikut:

Tabel 5.2 Spesifikasi Perangkat Lunak

|  |  |
| --- | --- |
| Sistem Operasi | Windows 10 |
| Database | MYSQL |
| Bahasa Pemrograman | PHP, HTML, CSS |

### Installasi Sistem

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam aplikasi SIKIBE adalah sebagai berikut:

1. Installasi Aplikasi
2. XAMPP

XAMPP ini dibutuhkan untuk menjalankan aplikasi dengan memanggil localhost dan juga untuk melakukan *running MySQL* yang berada di dalam XAMPP.

Berikut di bawah ini merupakan tahapan-tahapan instalasi XAMPP:

1. Unduh *installer* aplikasi XAMPP melalui *web* *apachefriends.org*
2. Lakukan double klik pada file XAMPP yang sudah diunduh
3. Klik *next* pada jendela *installer*
4. Pilih komponen yang akan diinstal atau di biarkan default untuk menginstal keseluruhan. Disarankan untuk menginstal keseluruhan
5. Pilih folder instalasi, lalu klik next
6. Pilih Bahasa yang ingin digunakan, lalu klik next
7. Jalankan instalasi dengan mengklik next pada jendela berikutnya
8. Tunggu hingga proses instalasi selesai
9. Setelah instalasi selesai, klik finish dan XAMPP siap digunakan.
10. Browser (Google Chrome)

Browser ini berguna untuk mengaktifkan dan menjalankan aplikasi SIKIBE yang sudah dibuat.

Berikut di bawah ini merupakan tahapan-tahapan instalasi Browser Google Chrome:

* 1. Unduh installer Google Chrome
  2. Jika diminta, klik jalankan atau simpan
  3. Jika memilih simpan maka lakukan klik dua kali pada tombol download untuk memulai proses instalasi
  4. Tunggu hingga proses instalasi selesai
  5. Setelah instalasi selesai, Google Chrome sudah bisa digunakan.

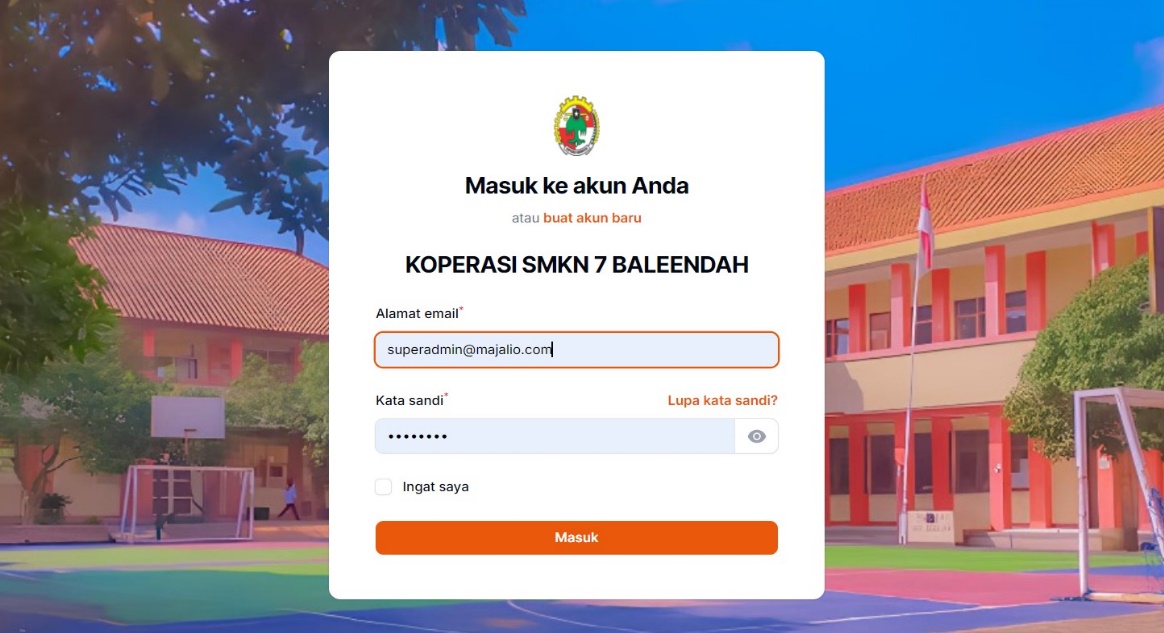
1. *Installasi Database*

Berkaitan dengan database yang digunakan pada aplikasi toko online merupakan MySQL dan MySQL itu sendiri sudah otomatis terdapat di dalam aplikasi XAMPP maka yang harus dilakukan untuk menjalankan database MySQL hanyalah dengan mengaktifkan XAMPP dan memanggil PHPMyAdmin di browser.

### Menjalankan Sistem

Pada bagian ini akan dijelaskan bagaimana cara-cara menjalankan aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

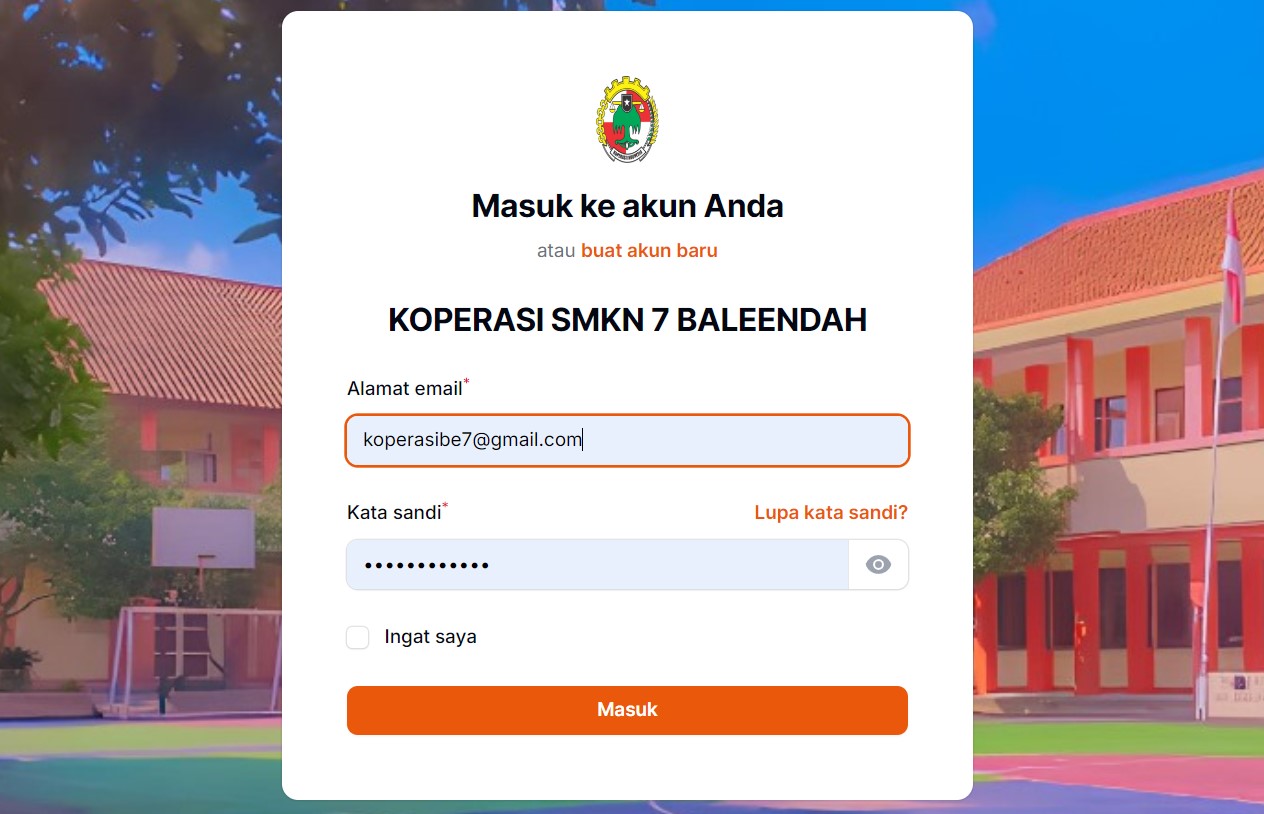
1. Halaman *login* Admin



Gambar . Halaman *Login* Admin

Untuk bisa memasuki aplikasi SIKIBE (Sistem Informasi Koperasi BE7) *Admin* diwajibkan untuk melakukan proses autentikasi dengan cara memasukkan *username* serta *password* yang telah diatur sebelumnya.

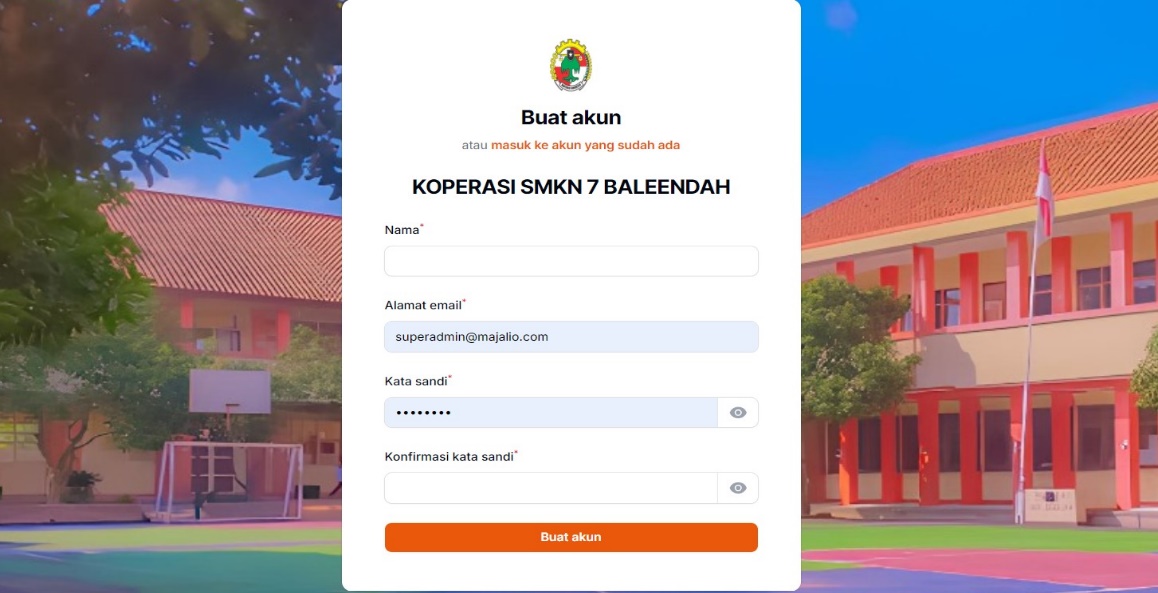
1. Halaman *login User*



Gambar 5.2 Halaman *Login User*

*User* melakukan proses autentikasi sama seperti *login* admin

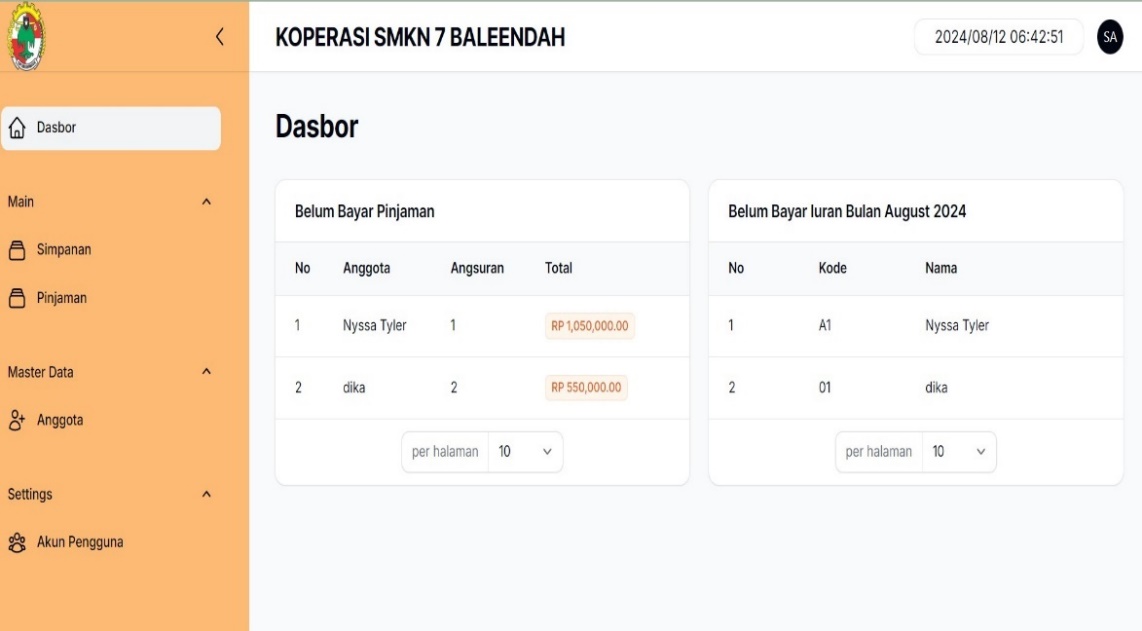
1. Halaman *register*



Gambar 5.3 Halaman *Register*

Jika *user* belum mempunyai *account* maka *user*  diwajibkan untuk mendaftaar terlebih dahulu di halaman *register* yang disediakan di dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

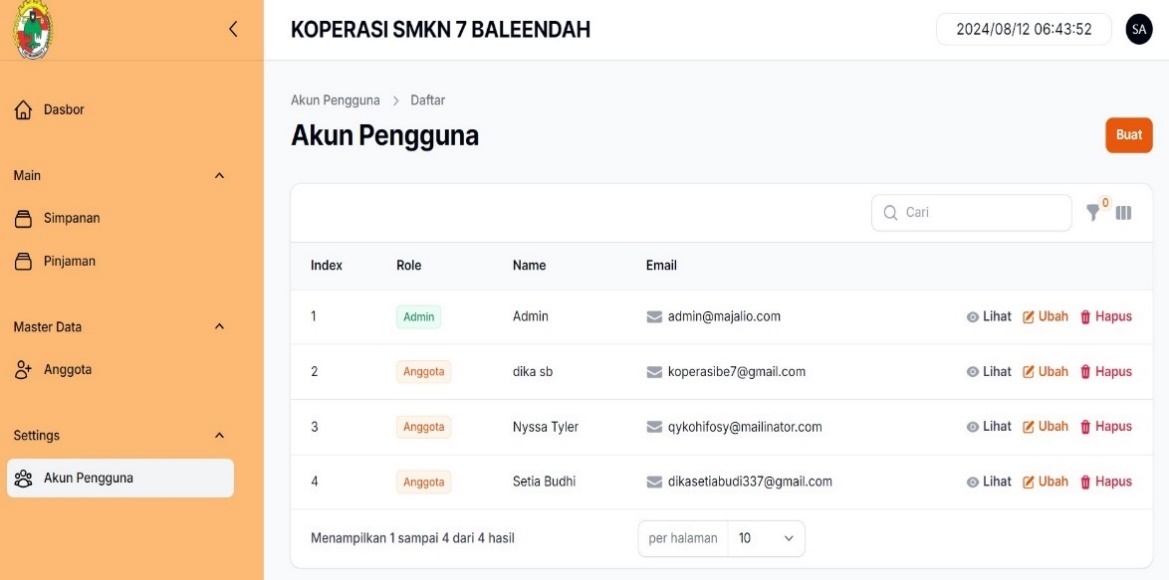
1. Halaman beranda



Gambar 5.4 Halaman Beranda

Setelah berhasil melakukan proses autentikasi *Admin* diarahkan ke halaman beranda.

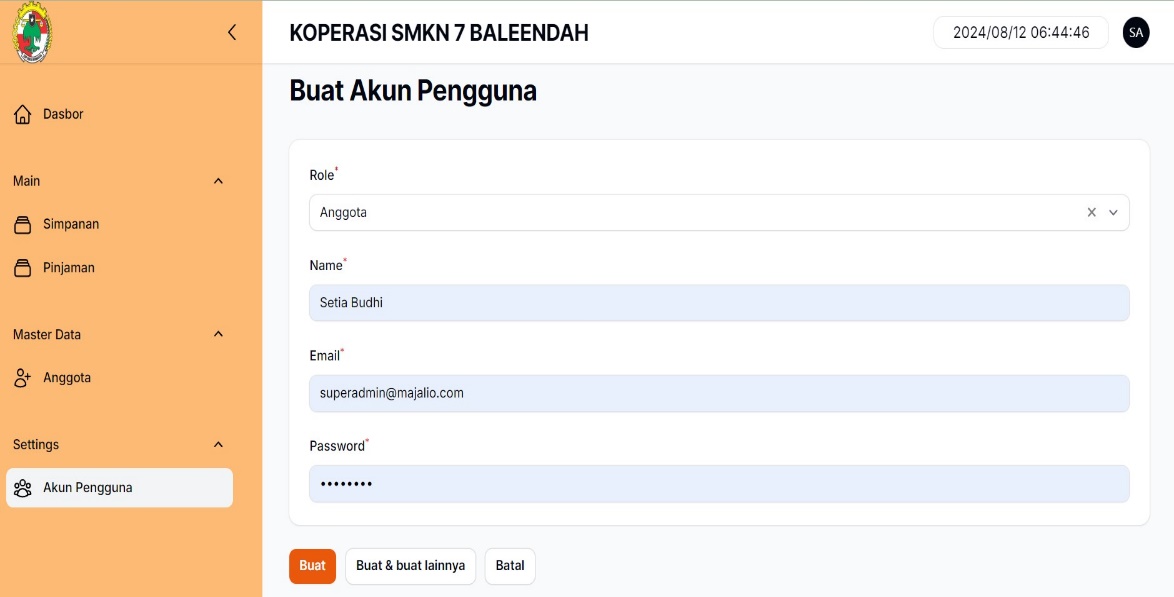
1. Halaman pengguna



Gambar .5 Halaman Pengguna

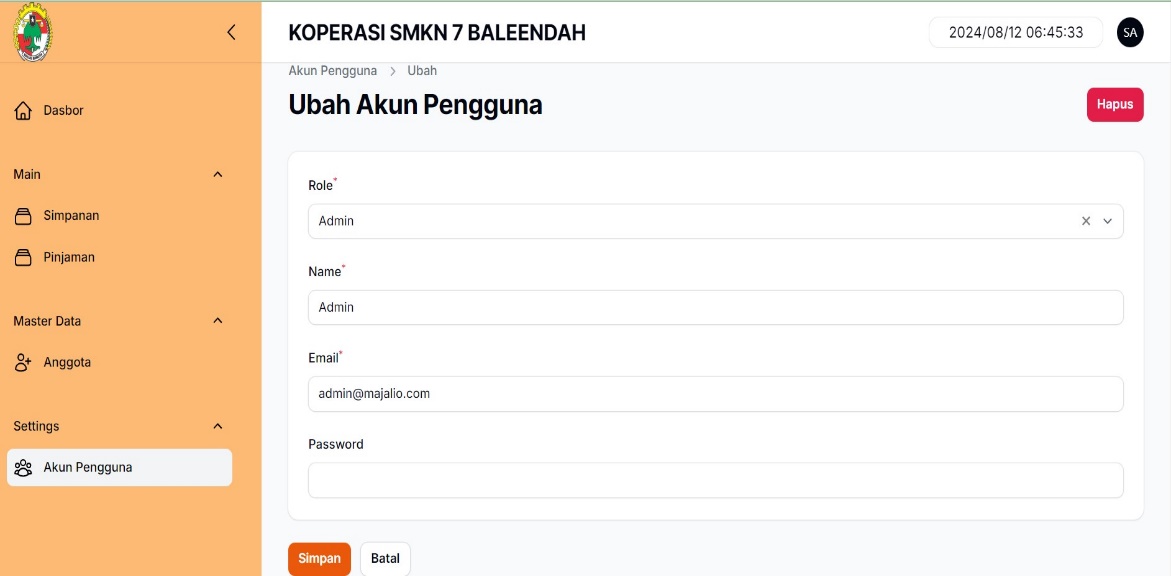
Halaman pengguna merupakan halaman yang digunakan untuk melihat daftar pengguna yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman tambah pengguna



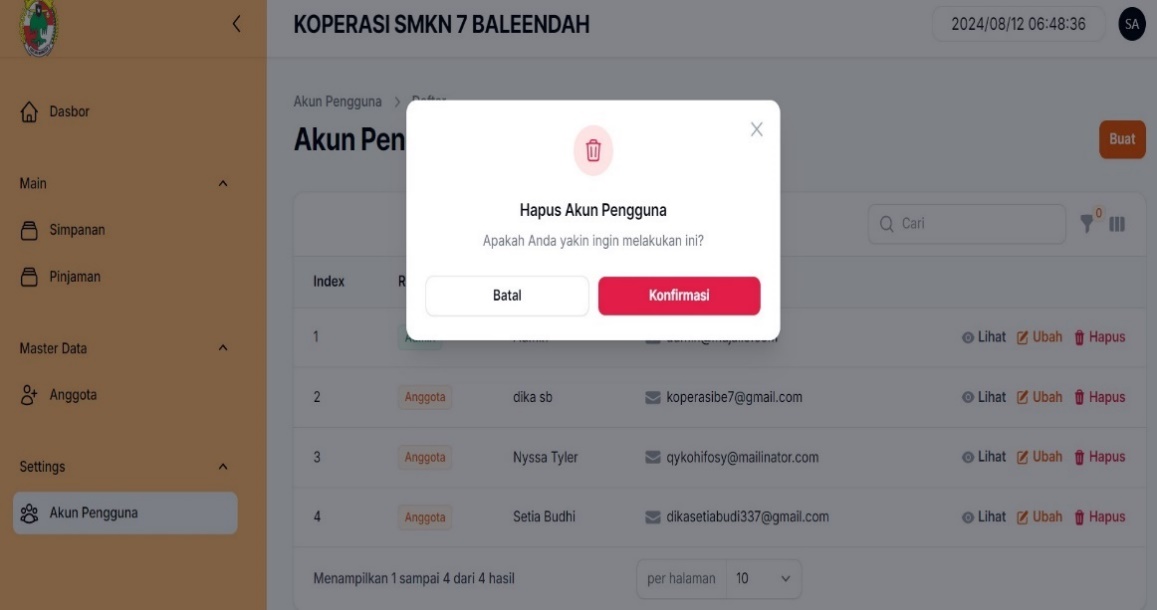
Gambar .6 Halaman Tambah Pengguna

Halaman tambah pengguna merupakan halaman yang digunakan untuk menambah pengguna yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman ubah pengguna

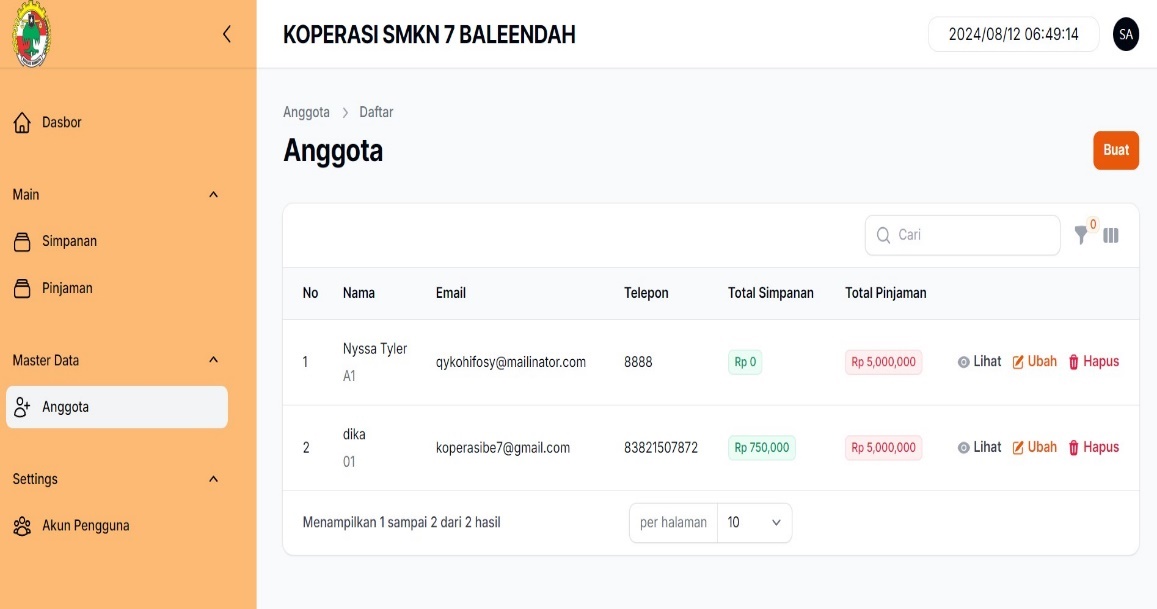
Gambar .7 Halaman Ubah Pengguna

Halaman tambah pengguna merupakan halaman yang digunakan untuk mengubah pengguna yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman hapus pengguna

Gambar .8 Halaman Hapus Pengguna

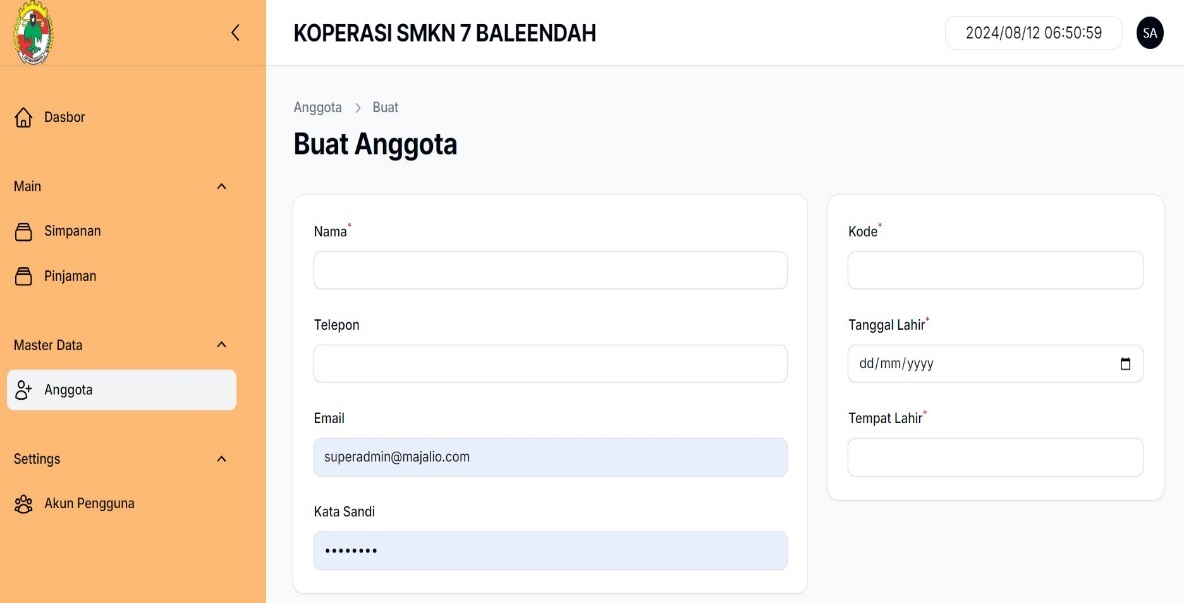
Halaman tambah pengguna merupakan halaman yang digunakan untuk menghapus pengguna yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman anggota

Gambar .9 Halaman Anggota

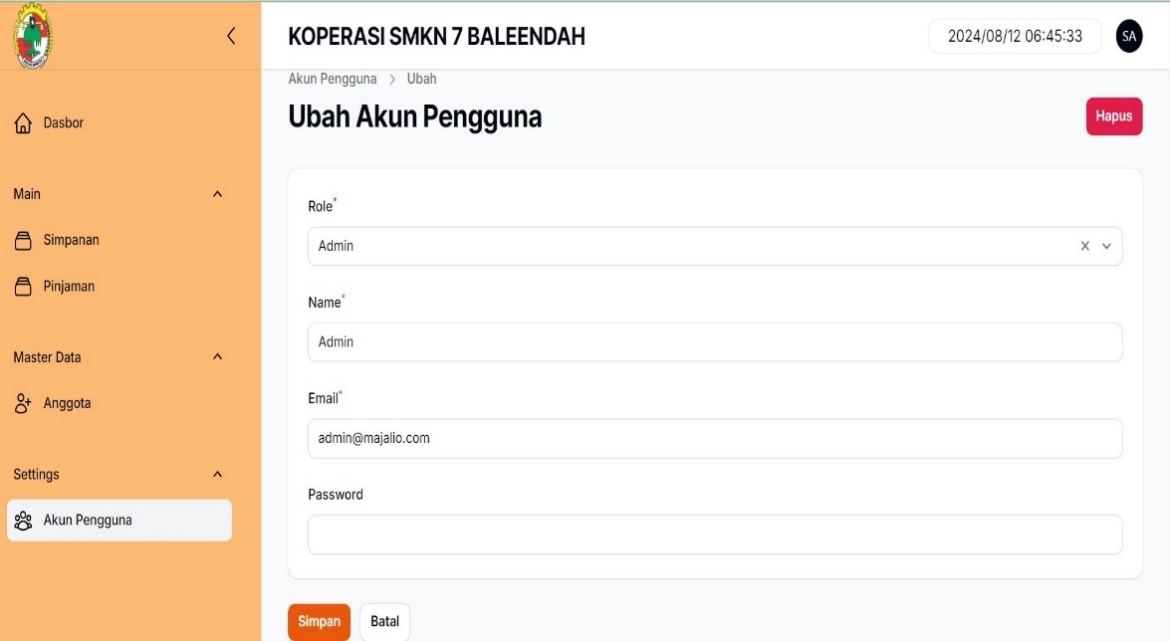
Halaman anggota merupakan halaman yang digunakan untuk melihat daftar anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman tambah anggota



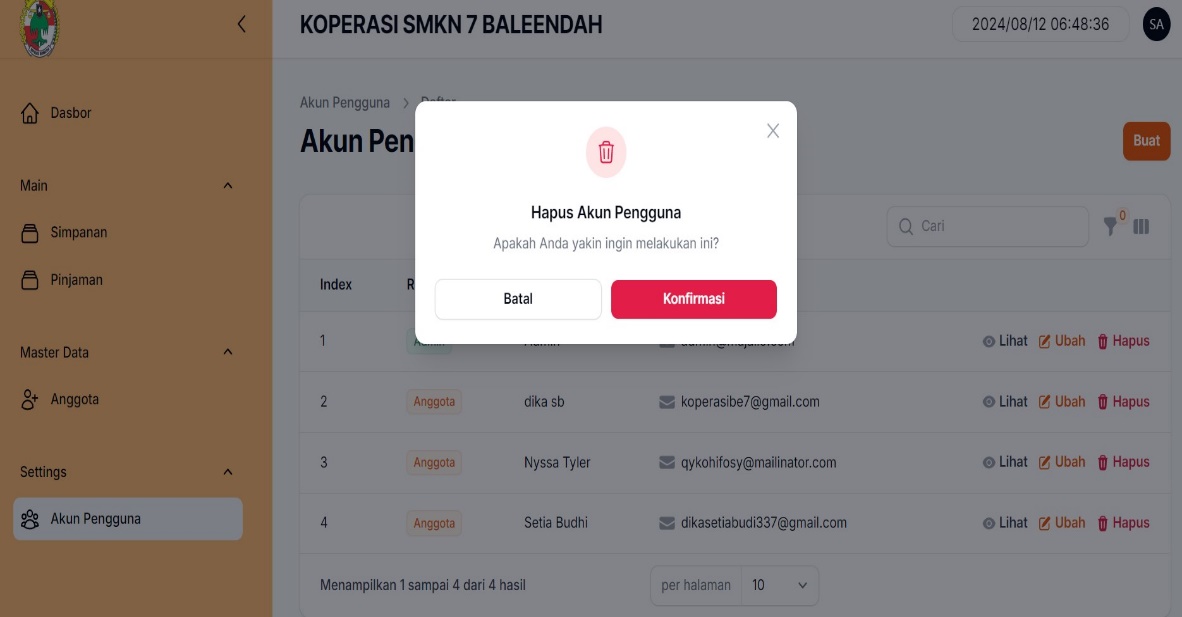
Gambar .10 Halaman Tambah Anggota

Halaman tambah anggota merupakan halaman yang digunakan untuk menambah anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman ubah anggota

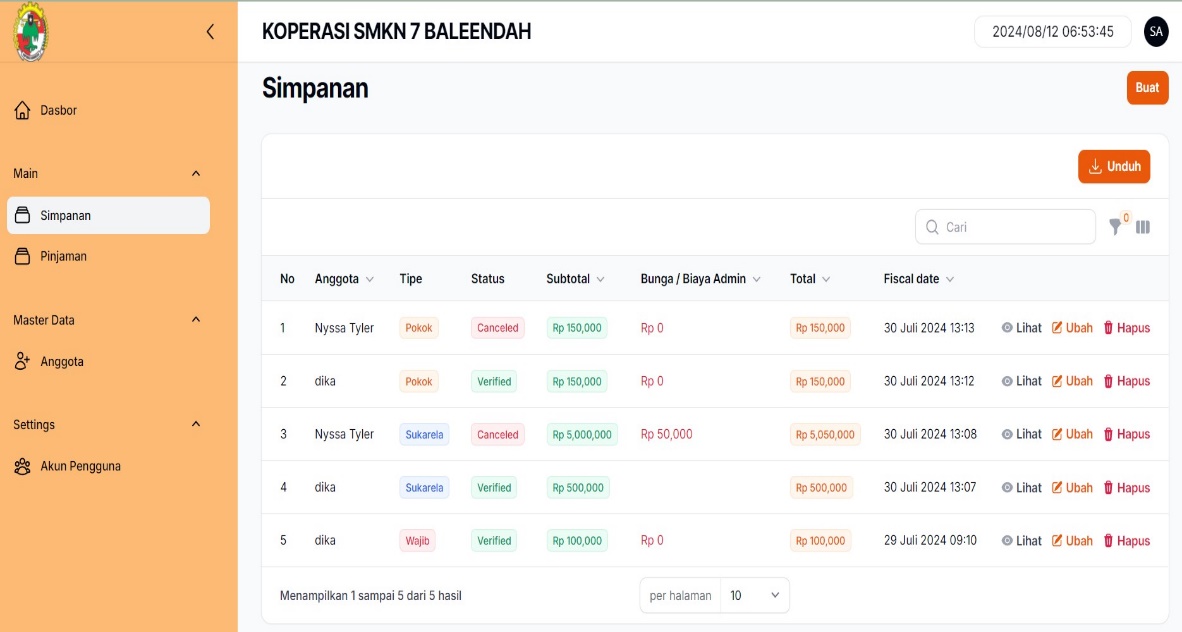
Gambar .11 Halaman Ubah Pengguna

Halaman anggota merupakan halaman yang digunakan untuk mengubah anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman hapus anggota

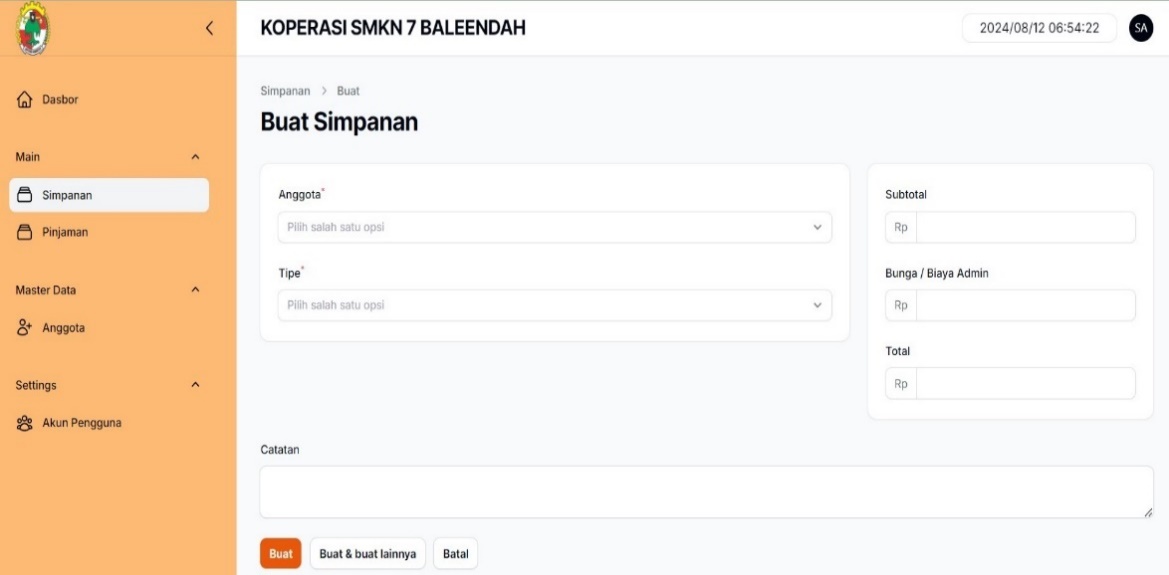
Gambar .1 Halaman Hapus Anggota

Halaman anggota merupakan halaman yang digunakan untuk menghapus anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman simpanan

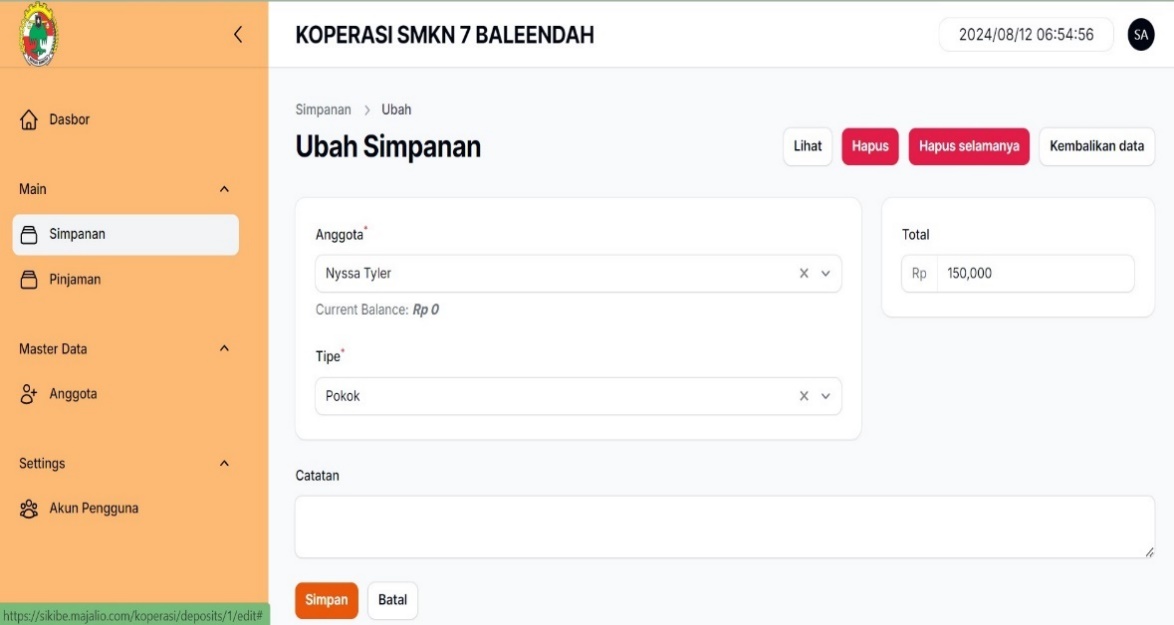
Gambar .13 Halaman Simpanan

Halaman simpan merupakan halaman yang digunakan untuk melihat daftar simpanan para anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman buat simpanan

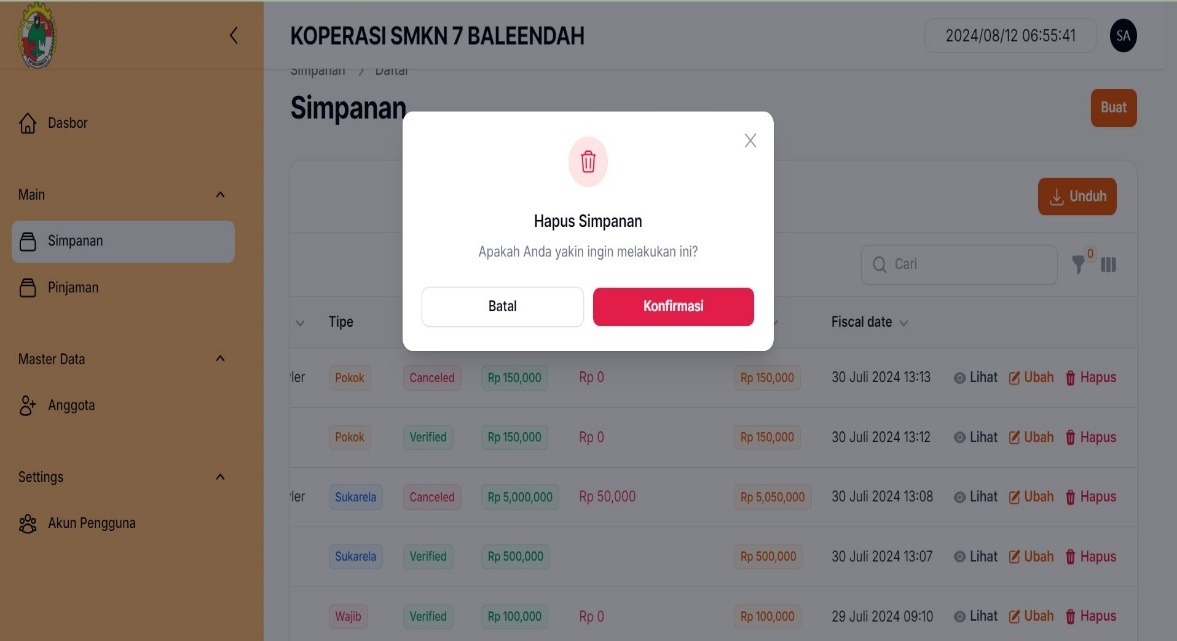
Gambar .14 Halaman Buat Simpanan

Halaman buat simpan merupakan halaman yang digunakan untuk membuat simpanan para anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman ubah simpanan

Gambar .15 Halaman Ubah Simpanan

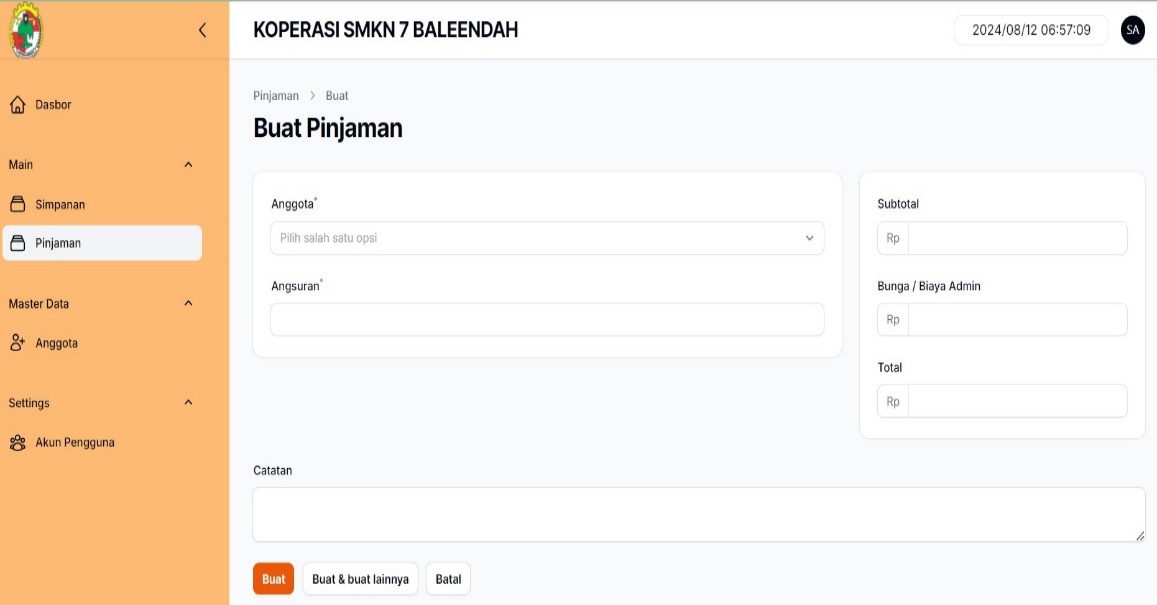
Halaman simpan merupakan halaman yang digunakan untuk membatalkan simpanan para anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman hapus simpanan

Gambar .16 Halaman Hapus Simpanan

Halaman hapus simpanan merupakan halaman yang digunakan untuk menghapus daftar simpanan para anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman Pinjaman



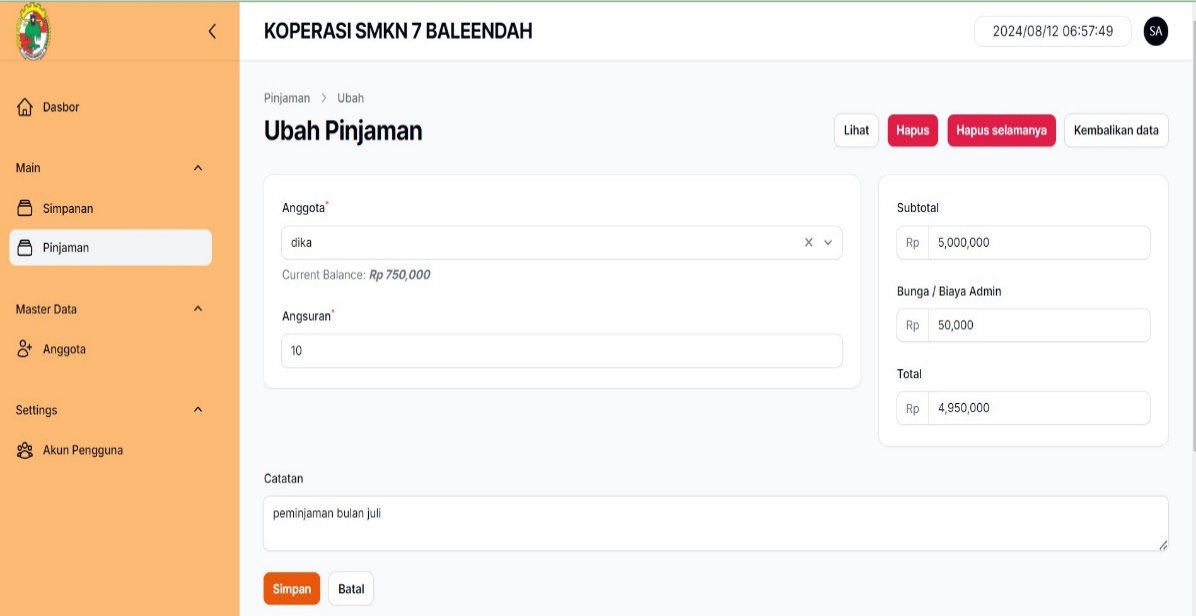
Gambar .17 Halaman Pinjaman

Halaman pinjaman merupakan halaman yang digunakan untuk melihat daftar pinjaman para anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman tambah pinjaman

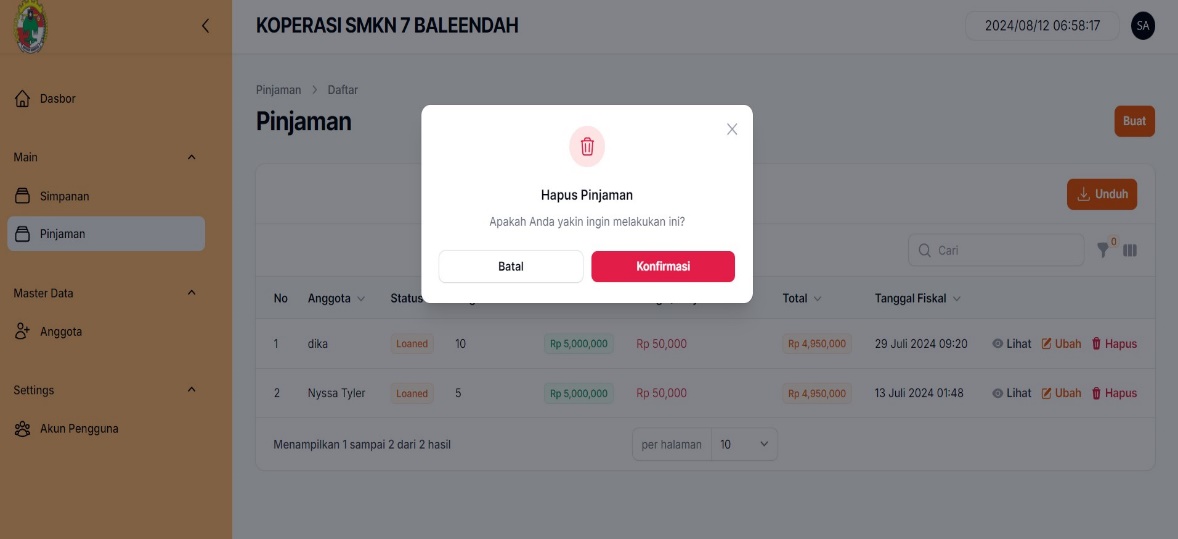
Gambar .18 Halaman Tambah Pinjaman

Halaman tambah pinjaman merupakan halaman yang digunakan untuk menambahkan pinjaman para anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman ubah pinjaman

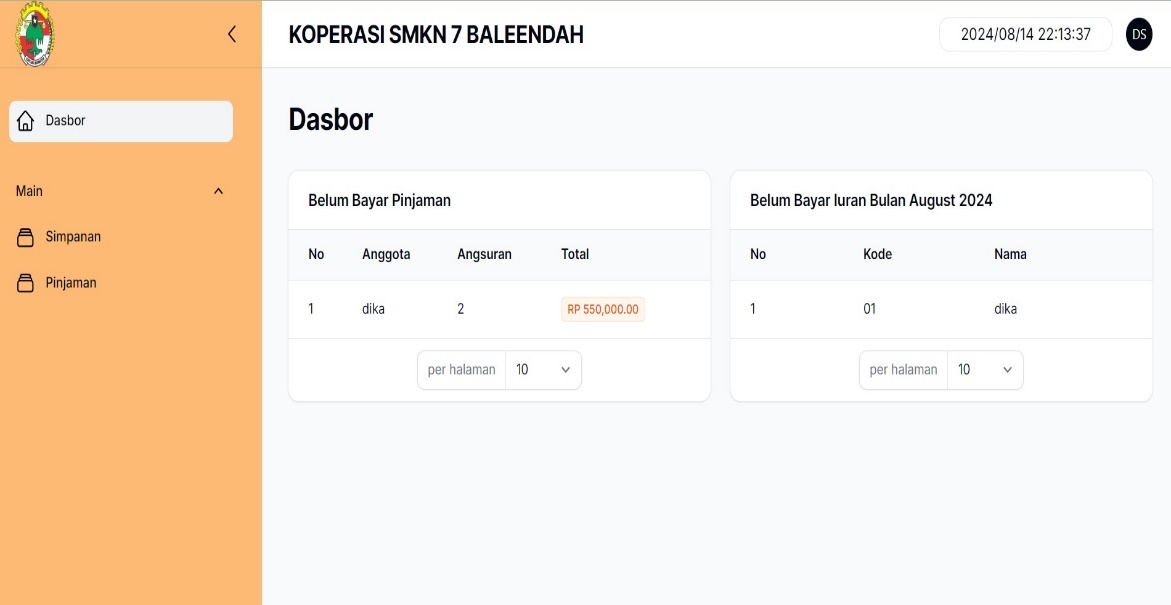
Gambar 5.19 Halaman Ubah Pinjaman

Halaman ubah pinjaman merupakan halaman yang digunakan untuk mengubah pinjaman para anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman hapus pinjaman

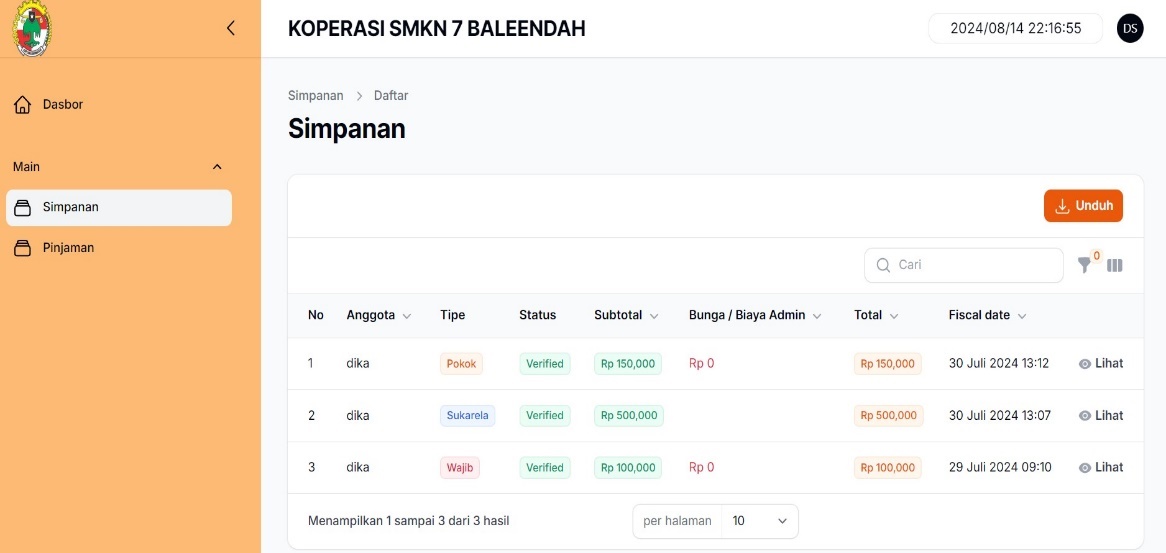
Gambar 5.20 Halaman Hapus Pinjaman

Halaman hapus pinjaman merupakan halaman yang digunakan untuk menghapus pinjaman para anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman *Dasboard User*

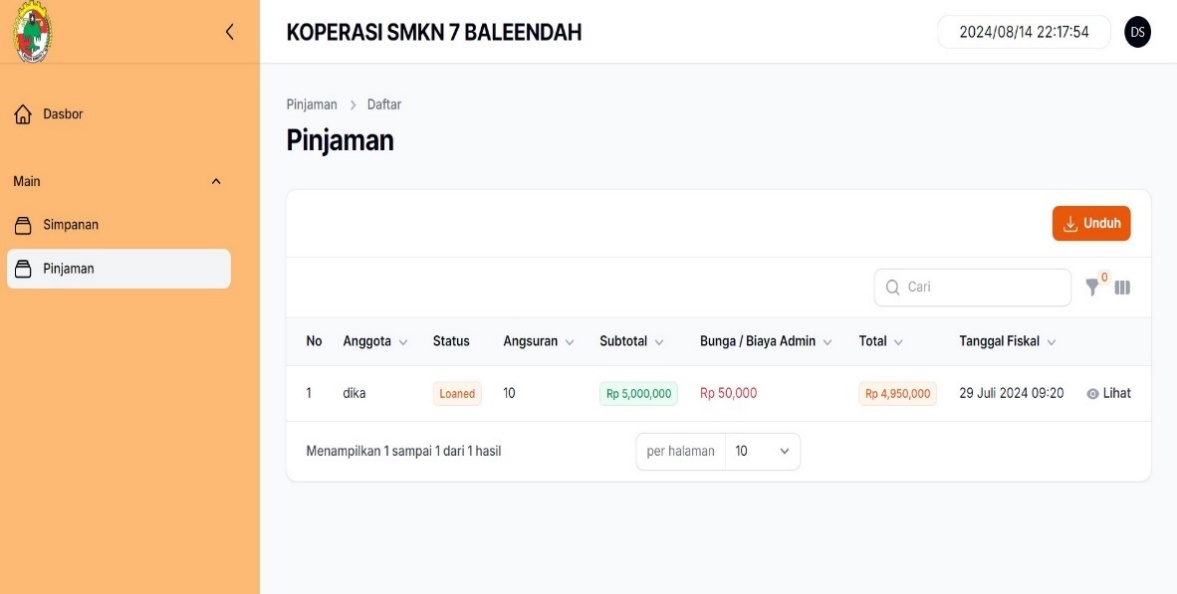
Gambar 5.21 Halaman Beranda *User*

Setelah berhasil melakukan proses autentikasi *User* diarahkan ke halaman beranda.

1. Halaman Simpanan *User*

Gambar 5.22 Halaman Simpanan *User*

Halaman simpanan merupakan halaman yang digunakan untuk melihat daftar simpanan anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

1. Halaman Pinjaman *User*

Gambar 5.23 Halaman Pinjaman *User*

Halaman pinjaman merupakan halaman yang digunakan untuk melihat daftar pinjaman anggota yang ada dalam aplikasi SIKIBE (Sistem Informasi Koperasi BE7).

## Pengujian

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan dan kekurangan pada sistem yang telah dibangun, pengujian bermaksud untuk mengetahui sistem dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan atau tidak. Pengujian pada penelitian ini melalui dua tahapan yaitu pengujian *alpha* yang akan dilakukan dengan menggunakan metode *black box* dan tahapan pengujian *beta* menggunakan model TAM.

### Pengujian Alpha

Pengujian *alpha* pada sistem ini menggunakan pengujian Unit (*Unit Testing*), pengujian ini menggunakan metode *blackbox testing*, proses pengujian awal yang dilakukan oleh tim penguji internal tanpa melihat ke dalam kode atau struktur dari sistem, pengujian ini berfokus pada aspek fungsionalitas, dimana penguji hanya mengevaluasi apakah sistem ini berfungsi sesuai dengan spesifikasi dan kebutuhan pengguna. berikut hasil pengujian yang telah dilakukan.

Tabel 5.3 Hasil Pengujian Aplikasi Dengan Metode *Blackbox*

| No | Item uji | Skenario Pengujian | Hasil Yang Diharapkan | Hasil | |
| --- | --- | --- | --- | --- | --- |
| Sesuai | Tidak Sesuai |
| 1. | Halaman *Login* | *Login* dengan *username* dan *password* yang *valid* | Sistem akan menampilkan halaman *dashboard* sesuai hak akses | Sesuai |  |
| 2. | Halaman *User* | *Login* dengan *username* dan *password* yang *valid* | Sistem akan menampilkan halaman *dashboard* sesuai hak akses | Sesuai |  |
| 3. | Halaman *Register* | Daftar menggunakan nama, *alamat email*, dan *password* | Sistem akan menampilkan halaman *login* setelah pengguna berhasil mendaftar | Sesuai |  |
| 4. | Halaman *Dashboard* | Masuk ke halaman *Dashboard* | Sistem akan menampilkan halaman *dashboard* | Sesuai |  |
| 5. | Halaman Pengguna | Masuk ke halaman Pengguna | Sistem akan menampilkan halaman daftar Pengguna | Sesuai |  |
| 6. | Halaman Anggota | Masuk ke halaman Anggota | Sistem akan menampilkan halaman daftar Anggota | Sesuai |  |
| ­7. | Halaman Simpan | Masuk ke halaman Simpan | Sistem akan menampilkan halaman daftar Simpan | Sesuai |  |
| 8. | Halaman Pinjam | Masuk ke halaman Pinjam | Sistem akan menampilkan halaman daftar Pinjam | Sesuai |  |
| 9. | *Logout* | *Logout* dari sistem | Dapat *logout* dari sistem | Sesuai |  |

Berdasarkan tabel 5.3 yang menampilkan hasil pengujian aplikasi dengan metode *blackbox*, maka dapat disimpulkan bahwa hasil pengujian aplikasi sudah berhasil dan sesuai dengan perancangan dan analisis yang dilakukan, semua fitur berjalan dengan baik sesuai dengan fungsi yang seharusnya dan penulis inginkan.

### Pengujian Beta

Pengujian *Beta* pada sistem ini menggunakan pengujian Unit (*Unit Testing*), pengujian ini menggunakan metode *blackbox testing*, berikut hasil pengujian yang telah dilakukan.

Pengujian beta dilakukan dengan mengisi kuesioner yang disiapkan oleh penulis dalam bentuk formulir *online* menggunakan *Google Formulir*, kuesioner diisi kepala koperasi, guru, dan staff tata usaha di SMK Negeri 7 Baleendah yang berjumlah 70 Orang. Pilihan Jawaban yang tersedia yaitu “Sangat Setuju”, “Setuju”, “Ragu-Ragu”, “Tidak Setuju”, dan “Sangat Tidak Setuju”. Setiap pilihan jawaban memiliki skor masing-masing sebesar:

1. Sangat Setuju (SS): 5
2. Setuju (ST): 4
3. Ragu-Ragu (RG): 3
4. Tidak Setuju (TS): 2
5. Sangat Tidak Setuju (STS): 1

Pada pengujian beta, kuesioner ditujukan untuk sarana prasarana, guru, dan staff tata usaha di SMK Negeri 7 Baleendah yang berjumlah 70 orang melalui perhitungan persamaan slovin dengan populasi 70 orang dan margin error 10%. dengan bentuk pertanyaan sebagai berikut:

Tabel 5.4 Pertanyaan Pengujian Beta

| Kategori | Pertanyaan | Variabel |
| --- | --- | --- |
| *Perceived*  *Usefulness* | Apakah Aplikasi SIKIBE sangat membantu untuk mempercepat pencarian data anggota koperasi? | PU1 |
| Apakah Aplikasi SIKIBE sangat membantu bagian koperasi dalam merekap data anggota koperasi? | PU2 |
| Apakah Aplikasi SIKIBE bermanfaat untuk menyimpan data anggota koperasi agar tetap aman? | PU3 |
| Apakah Aplikasi SIKIBE bermanfaat sebagai laporan bagian koperasi dalam pendataan anggota koperasi? | PU4 |
| *Perceived Ease*  *of*  *Use* | Apakah Aplikasi SIKIBE ini mudah digunakan? | PEU1 |
| Apakah Aplikasi SIKIBE ini mudah dimengerti dan dipahami? | PEU2 |
| Apakah Aplikasi SIKIBE ini mempunyai tampilan yang mudah dipahami? | PEU3 |
| Apakah Aplikasi SIKIBE ini mudah diakses? | PEU4 |
| *Attitude Toward*  *Using* | Apakah anda merasa senang untuk menggunakan Aplikasi SIKIBE ini? | ATU1 |
| Apakah anda menikmati alur dari Aplikasi SIKIBE ini? | ATU2 |
| Apakah anda berencana untuk menggunakan Aplikasi SIKIBE ini? | ATU3 |
| Apakah Aplikasi SIKIBE ini mempunyai tampilan yang menarik? | ATU4 |
| *Behavioral*  *Intention* | Apakah anda berniat menggunakan Aplikasi SIKIBE ini? | BI1 |
| Apakah anda akan selalu menggunakan Aplikasi SIKIBE ini untuk melakukan pendataan anggota? | BI2 |
| Apakah anda akan menggunakan Aplikasi SIKIBE ini kapan saja saat butuh untuk pendataan anggota? | BI3 |
| Apakah anda akan tetap menggunakan Aplikasi SIKIBE ini untuk mendapatkan informasi pendataan anggota? | BI4 |
| *Actual System*  *Use* | Secara keseluruhan apakah anda merasa puas dengan kinerja Aplikasi SIKIBE ini? | AU1 |
| Apakah anda menghabiskan waktu cukup lama dalam menggunakan Aplikasi SIKIBE ini? | AU2 |

1. **Pengujian Validitas**

Pengujian validitas dilakukan untuk menentukan instrumen kuesioner yang dinyatakan valid atau tidak valid. Instrumen dinyatakan valid jika nilai korelasi *(pearson correlation)* r Hitung lebih besar dari r Tabel. Berikut adalah uji validitas kuesioner pada SPSS:

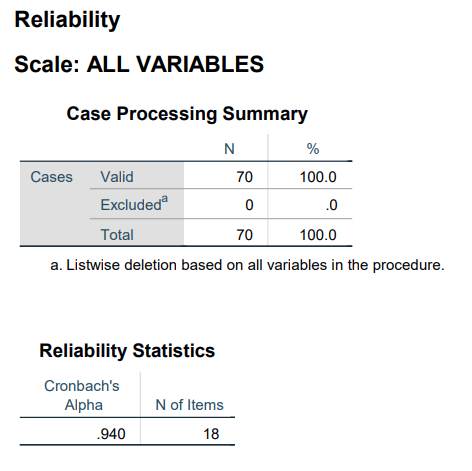
Tabel 5.5 Pengujian Validitas

| Variabel | rHitung | rTabel df=70 α = 0.05 | Pertanyaan |
| --- | --- | --- | --- |
| PU1 | .805 | .235 | Apakah Aplikasi SIKIBE sangat membantu untuk mempercepat pencarian data anggota koperasi? |
| PU2 | .785 | .235 | Apakah Aplikasi SIKIBE sangat membantu bagian koperasi dalam merekap data anggota koperasi? |
| PU3 | .757 | .235 | Apakah Aplikasi SIKIBE bermanfaat untuk menyimpan data anggota koperasi agar tetap aman? |
| PU4 | .822 | .235 | Apakah Aplikasi SIKIBE bermanfaat sebagai laporan bagian koperasi dalam pendataan anggota koperasi? |
| PEU1 | .771 | .235 | Apakah Aplikasi SIKIBE ini mudah digunakan? |
| PEU2 | .824 | .235 | Apakah Aplikasi SIKIBE ini mudah dimengerti dan dipahami? |
| PEU3 | .821 | .235 | Apakah Aplikasi SIKIBE ini mempunyai tampilan yang mudah dipahami? |
| PEU4 | .823 | .235 | Apakah Aplikasi SIKIBE ini mudah diakses? |
| ATU1 | .856 | .235 | Apakah anda merasa senang untuk menggunakan Aplikasi SIKIBE ini? |
| ATU2 | .850 | .235 | Apakah anda menikmati alur dari Aplikasi SIKIBE ini? |
| ATU3 | .331 | .235 | Apakah anda berencana untuk menggunakan Aplikasi SIKIBE ini? |
| ATU4 | .289 | .235 | Apakah Aplikasi SIKIBE ini mempunyai tampilan yang menarik? |
| BI1 | .817 | .235 | Apakah anda berniat menggunakan Aplikasi SIKIBE ini? |
| BI2 | .791 | .235 | Apakah anda akan selalu menggunakan Aplikasi SIKIBE ini untuk melakukan pendataan anggota? |
| BI3 | .752 | .235 | Apakah anda akan menggunakan Aplikasi SIKIBE ini kapan saja saat butuh untuk pendataan anggota? |
| BI4 | .778 | .235 | Apakah anda akan tetap menggunakan Aplikasi SIKIBE ini untuk mendapatkan informasi pendataan anggota? |
| AU1 | .794 | .235 | Secara keseluruhan apakah anda merasa puas dengan kinerja Aplikasi SIKIBE ini? |
| AU2 | .485 | .235 | Apakah anda menghabiskan waktu cukup lama dalam menggunakan Aplikasi SIKIBE ini? |

Berdasarkan analisa diatas maka dapat disimpulkan bahwa 18 item r Hitung lebih besar dari r Tabel, maka dapat dinyatakan valid.

1. **Pengujian Reliabilitas**

Pengujian Reliabilitas dilakukan untuk menentukan bahwa instrumen kuesioner memberikan hasil *reliable* atau handal. Suatu instrumen dapat dikatakan *reliable* apabila memiliki nilai *Cronbach Alpha* lebih besar dari 0.6.



Gambar .24 Pengujian Realibilitas

Berdasarkan hasil pengujian reliabilitas diatas, diketahui angka *Cronbach’s Alpha* adalah sebesar 0.940, jadi angka tersebut lebih besar dari nilai minimal *Cronbach Alpha* 0,6. Oleh karena itu dapat disimpulkan bahwa kuesioner ini dapat dinyatakan *reliable*.

# BAB VI KESIMPULAN

## Kesimpulan

Berdasarkan penelitian yang telah dilakukan penulis melalui beberapa tahapan yang dilakukan pada bab sebelumnya, maka penulis mengambil kesimpulan bahwa:

Implementasi algoritma *sequential search* telah terbukti mempermudah proses pencarian data anggota koperasi. Dengan sistem ini, petugas koperasi dapat dengan cepat menemukan informasi yang dibutuhkan tanpa harus mencari secara manual melalui berkas fisik.

Sistem digitalisasi yang dibangun untuk pencatatan transaksi anggota koperasi telah memberikan kemudahan dan mengurangi kesalahan yang mungkin terjadi pada pencatatan manual. Transaksi anggota kini dapat dicatat secara *real-time*, sehingga data yang dihasilkan lebih akurat dan *up-to-date*.

Sistem pembuatan laporan digital telah mengurangi waktu yang dibutuhkan untuk menyusun laporan koperasi. Dengan fitur pencetakan dan pengunduhan dokumen, petugas koperasi dapat menghasilkan laporan dengan lebih efisien dan mudah diakses.

Hasil Pengujian Aplikasi SIKIBE sudah digunakan oleh petugas koperasi SMK Negeri 7 baleendah, sampai saat ini Aplikasi SIKIBE sangat membantu petugas koperasi dalam transaksi koperasi simpan pinjam, dan membantu kinerja petugas koperasi menjadi lebih efisien dibandingkan dengan sebelumnya.

Hasil dari pengujian menggunakan *alpha* dan *beta*, Sistem Aplikasi SIKIBE semua unit dan menu yang diuji sudah sesuai dengan skenario yang telah di tentukan. angka Cronbach’s Alpha adalah sebesar 0.940, jadi angka tersebut lebih besar dari nilai minimal Cronbach Alpha 0,6. Oleh karena itu dapat disimpulkan bahwa kuesioner ini dapat dinyatakan reliable.

## Saran

Untuk memaksimalkan hasil dari penggunaan Aplikasi SIKIBE, beberapa saran berikut diharapkan dapat membantu meningkatkan kinerja dan efektivitas aplikasi:

Meskipun algoritma *sequential search* sudah memadai untuk pencarian data, disarankan untuk mengeksplorasi dan mengimplementasikan algoritma pencarian lain seperti *binary search* atau *hash table* untuk meningkatkan kecepatan dan efisiensi pencarian data, terutama jika jumlah data terus bertambah.

Untuk memastikan sistem digitalisasi berjalan optimal, disarankan untuk mengadakan pelatihan bagi petugas koperasi. Pelatihan ini bertujuan agar petugas terbiasa menggunakan sistem baru dan dapat memaksimalkan fitur-fitur yang tersedia.

Untuk meningkatkan efisiensi operasional koperasi, disarankan agar sistem ini diintegrasikan dengan sistem lain yang ada di SMK Negeri 7 Baleendah, seperti sistem keuangan sekolah atau sistem administrasi lainnya, guna menciptakan ekosistem digital yang lebih terintegrasi dan efisien.

Agar sistem informasi koperasi selalu optimal maka disarankan agar sistem ini selalu ada maintenance atau pemeliharaan sistem aplikasi sikibe.

Agar sistem informasi koperasi lebih kompleks maka disarankan harus ada pengembangan sistem berupa penambahan kode OTP *(One-Time Password*) untuk menambah keamanan teknologi autentikasi.

Dengan menerapkan saran-saran tersebut, diharapkan aplikasi SIKIBE (Sistem Informasi Koperasi Be7) dapat terus berkembang dan memberikan manfaat yang optimal bagi penggunanya. Pencarian data anggota koperasi akan menjadi lebih efisien, cepat, dan mudah diakses, sehingga mendukung kelancaran bagi sarana dan prasarana terutama pihak pengelola secara keseluruhan.

# DAFTAR PUSTAKA

’Afiifah, K., Azzahra, Z. F., & Anggoro, A. D. (2022). Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review. *Intech*, *3*(2), 18–22. https://doi.org/10.54895/intech.v3i2.1682

Anggara, M. B., Muslimin, I. A. ., & Marcela, H. M. (2023). *Optimasi Pengelolaan Persediaan Barang Menggunakan MetodePerpetual Pada Aplikasi Inventory Di Pt. Visi Karya Prakarsa*. *10*, 77–81.

Arhandi, P. P., Arief, S. N., & Firdausi, A. T. (2022). Pengembangan Website Pendukung Mastery Based Learning Untuk Pembelajaran Mahasiswa. *Jurnal Informatika Polinema*, *9*(1), 51–58. https://doi.org/10.33795/jip.v9i1.966

Departemen, D. I., Pt, K., Pacific, I., Muharam, Y., & Nuryadi, D. (2023). *Rancang Bangun Aplikasi Persediaan Kain Berbasis Webdengan Metode Reorder Point ( Rop )*. *10*, 59–64.

Handayani, H., Faizah, K. U., Mutiara Ayulya, A., Rozan, M. F., Wulan, D., & Hamzah, M. L. (2023). Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development Designing a Web-Based Inventory Information System Using the Agile Software Development Method. *Jurnal Testing Dan Implementasi Sistem Informasi*, *1*(1), 29–40.

Herdiana, Y. (2023). Perancangan Aplikasi E-Arsip Menggunakan Algoritma Sequential Search Untuk Pengarsipan Surat Pada PT. Visi Karya Prakarsa. *COMPUTING| Jurnal Informatika*, *10*, 72–76. https://ejournal.unibba.ac.id/index.php/computing/article/view/1297%0Ahttps://ejournal.unibba.ac.id/index.php/computing/article/download/1297/1016

Isa, I. G. T., & Hartawan, G. P. (2017). Perancangan Aplikasi Koperasi Simpan Pinjam Berbasis Web (Studi. *Jurnal Ilmiah Ilmu Ekonomi*, *5*(10), 139–151.

Kurniawan, t bayu, & Syarifuddin. (2020). Perancangan Sistem Aplikasi Pemesanan Makanan dan Minuman Pada Cafetaria NO Caffe di TAnjung Balai Karimun Menggunakan Bahasa Pemrograman PHP dan MySQL. *Jurnal Tikar*, *1*(2), 192–206.

Kurniawan, R., & Budi, M. (2022). Perancangan Ui/Ux Aplikasi Manajemen Penelitian Dan PengabdianKepada Masyarakat Menggunakan Aplikasi Figma. *Jurnal Sistem Informasi*, *5*(1), 2–7.

M Teguh Prihandoyo. (2018). Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web. *Jurnal Informatika: Jurnal Pengembangan IT*, *3*(1), 126–129.

Mubarak, A. (2019). Rancang Bangun Aplikasi Web Sekolah Menggunakan Uml (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek. *JIKO (Jurnal Informatika Dan Komputer)*, *2*(1), 19–25. https://doi.org/10.33387/jiko.v2i1.1052

Mulyanto, A., & Setiawan, W. (2020). Penerapan Metode Web Engineering Menggunakan Laravel 5 Dalam Pengembangan Penjualan Toko Online Hijapedia Berbasis Website Di Cikarang Bekasi. *Jurnal Informatika SIMANTIK*, *5*(2), 18–23.

Nanda Syarif, M., Pambudiyatno, N., Utomo, W., Jemur Andayani No, J. I., & Siwalankerto Kec Wonocolo, K. (2023). Rancangan Sistem Presensi Dan Rekapitulasi Jurnal Kegiatan Ojt Menggunakan Visual Studio Code Berbasis Web Di Airnav Cabang Matsc. *PROSIDING Seminar Nasional Inovasi Teknologi Penerbangan (SNITP) Tahun*, 2023.

Oktaviani, N., Widiarta, I. M., & Nurlaily. (2019). Sistem Informasi Inventaris Barang Berbasis Web Pada Smp Negeri 1 Buer. *Jurnal Informatika, Teknologi Dan Sains*, *1*(2), 160–168. https://doi.org/10.51401/jinteks.v1i2.422

Rahmat, D., A’zizah, S., & Mulyani, S. (2022). Perkalian Menggunakan Bahasa Pemrograman Perulangan (Looping) Berbantu Sublime Text Dan Xampp. *Djtechno: Jurnal Teknologi Informasi*, *3*(2), 149–155. https://doi.org/10.46576/djtechno.v3i2.2417

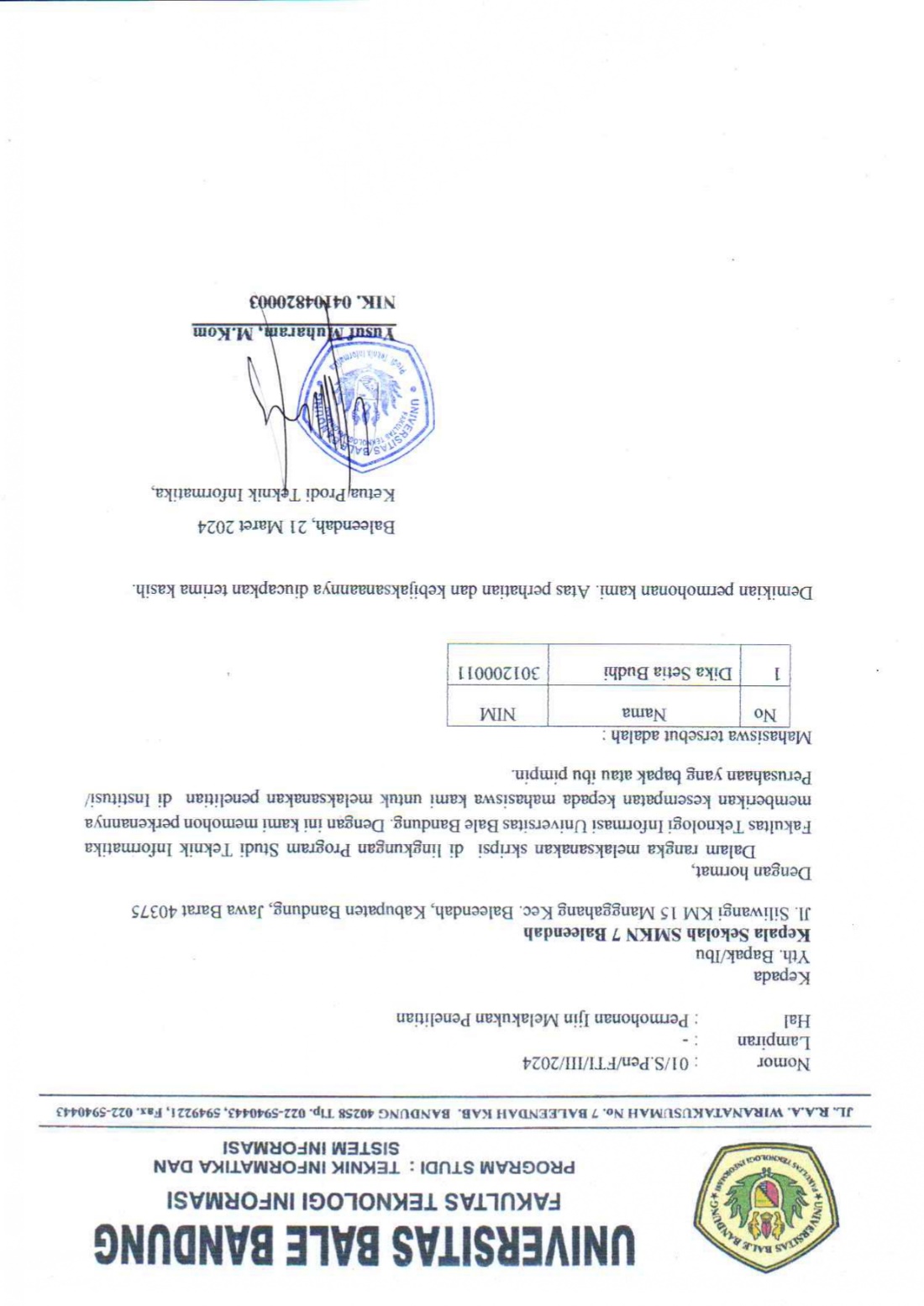
Stinjak, M. L., & Masya, F. (2021). Perancangan Dan Implementasi Sistem Informasi Inventory Berbasis Website Menggunakan Iterative Waterfall. *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, *6*(2), 83–91. https://doi.org/10.36341/rabit.v6i2.1687

Titi Komalasari, R. (2021). Menggunakan Metode White Box Testing dan Black Box Testing. *Jurnal Teknologi Informasi*, *7*(1), 50–57. http://ejournal.urindo.ac.id/index.php/TI

Wahyuni, W. S., Andryana, S., & Rahman, B. (2022). Penggunaan Algoritma Sequential Searching Pada Aplikasi Perpustakaan Berbasis Web. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *7*(2), 294–302. https://doi.org/10.29100/jipi.v7i2.2646

Yaya Suharya, Nurul Imamah, dan N. D. (2023). Penjadwalan Dan Pemberian Pakan Ikan Otomatis Berbasis Internet of Thing Menggunakan Nodemcu Esp8266 Dan Aplikasi Blynk Studi Kasus : Toko Fish Friendly. *Desember 2023 Journal Article*, *10*(02), 65–71. https://ejournal.unibba.ac.id/index.php/computing/article/view/1296

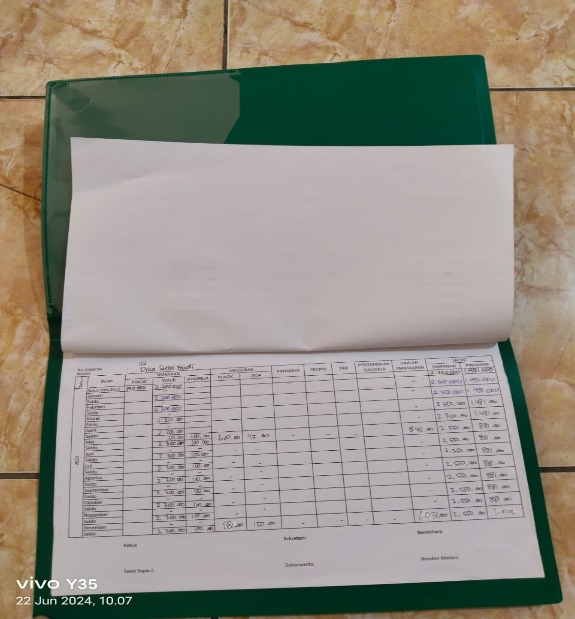
# LAMPIRAN

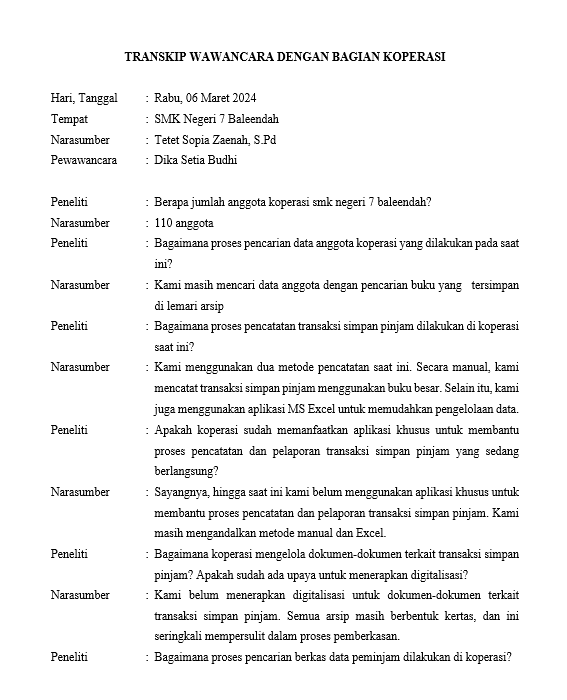


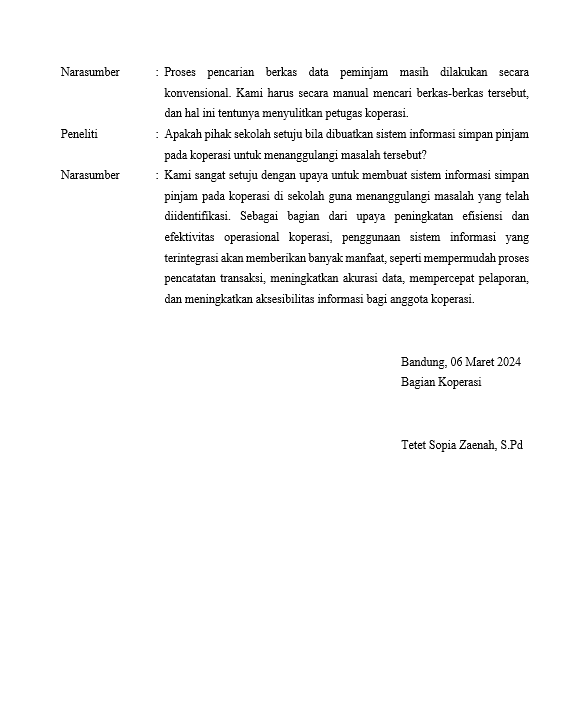
# LAMPIRAN 1: Surat Pengantar Penelitian

**LAMPIRAN 2:** Surat Balasan Penelitian

******LAMPIRAN 3:** Dokumentasi

****

**LAMPIRAN 4:** Transkip Wawancara

****

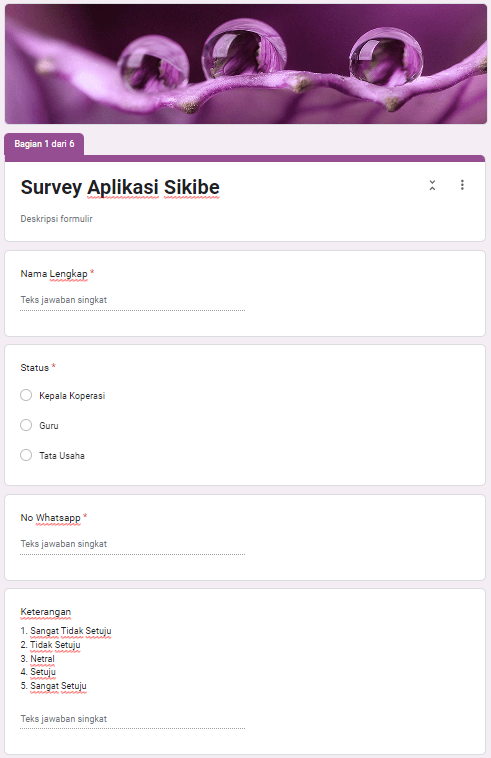
**LAMPIRAN 5:** Data Pengujian

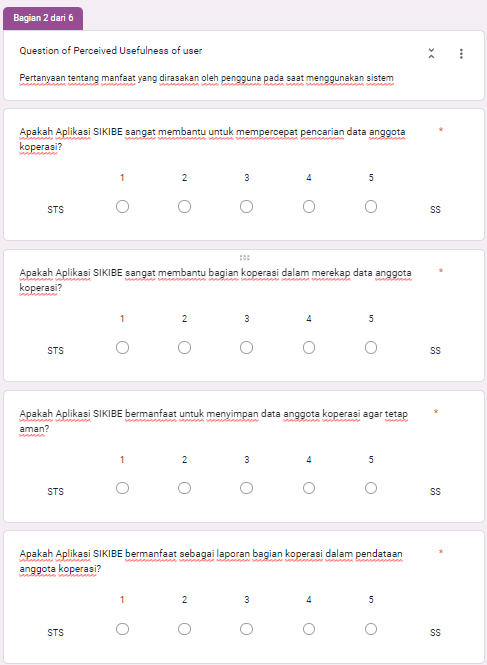
1. Apakah Aplikasi SIKIBE sangat membantu untuk mempercepat pencarian data anggota koperasi?
2. Apakah Aplikasi SIKIBE sangat membantu bagian koperasi dalam merekap data anggota koperasi?
3. Apakah Aplikasi SIKIBE bermanfaat untuk menyimpan data anggota koperasi agar tetap aman?
4. Apakah Aplikasi SIKIBE bermanfaat sebagai laporan bagian koperasi dalam pendataan anggota koperasi?
5. Apakah Aplikasi SIKIBE ini mudah digunakan?
6. Apakah Aplikasi SIKIBE ini mudah dimengerti dan dipahami?
7. Apakah Aplikasi SIKIBE ini mempunyai tampilan yang mudah dipahami?
8. Apakah Aplikasi SIKIBE ini mudah diakses?
9. Apakah anda merasa senang untuk menggunakan Aplikasi SIKIBE ini?
10. Apakah anda menikmati alur dari Aplikasi SIKIBE ini?
11. Apakah anda berencana untuk menggunakan Aplikasi SIKIBE ini?
12. Apakah Aplikasi SIKIBE ini mempunyai tampilan yang menarik?
13. Apakah anda berniat menggunakan Aplikasi SIKIBE ini?
14. Apakah anda akan selalu menggunakan Aplikasi SIKIBE ini untuk melakukan pendataan anggota?
15. Apakah anda akan menggunakan Aplikasi SIKIBE ini kapan saja saat butuh untuk pendataan anggota?
16. Apakah anda akan tetap menggunakan Aplikasi SIKIBE ini untuk mendapatkan informasi pendataan anggota?
17. Secara keseluruhan apakah anda merasa puas dengan kinerja Aplikasi SIKIBE ini?
18. Apakah anda menghabiskan waktu cukup lama dalam menggunakan Aplikasi SIKIBE ini?

Tabel

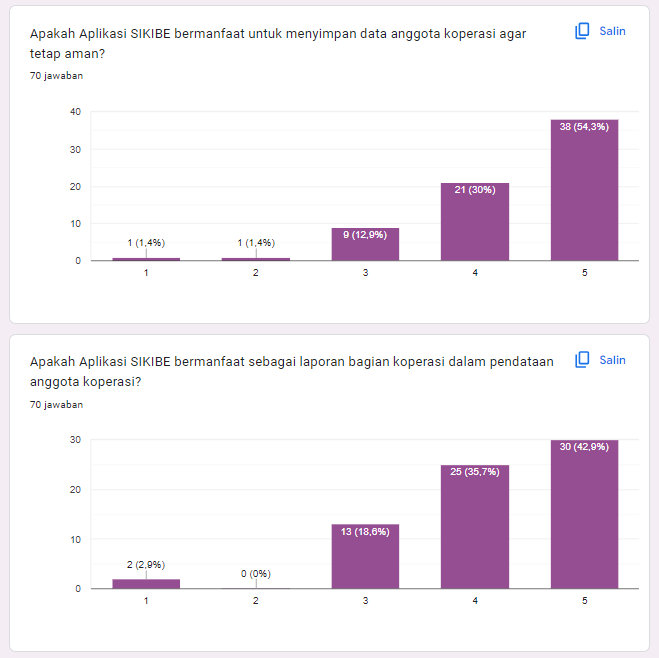
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Umur | Jenis Kelamin | Jabatan | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 | Q13 | Q14 | Q15 | Q16 | Q17 | Q18 | Q19 | Q20 | Q21 | Q22 | Q23 |
| 57 tahun | Laki-laki | Kepala Sekolah | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 44 - 58 tahun | Laki Laki Perempuan | Wakil Kepala Sekolah | 5 | 4 | 4 | 4 | 5 | 5 | 4 | 5 | 4 | 5 | 5 | 5 | 5 | 4 | 4 | 5 | 5 | 4 | 4 | 5 | 5 | 5 | 4 |
| 25 – 58 tahun | Laki Laki Perempuan | Guru | 4 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 4 | 4 | 4 | 5 | 4 | 4 | 5 | 5 | 5 | 5 |
| 22 - 53 tahun | Laki Laki Perempuan | Staf  Administrasi | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 5 | 4 | 5 | 5 | 4 | 4 | 5 | 4 | 4 | 4 | 5 | 4 | 5 | 4 |

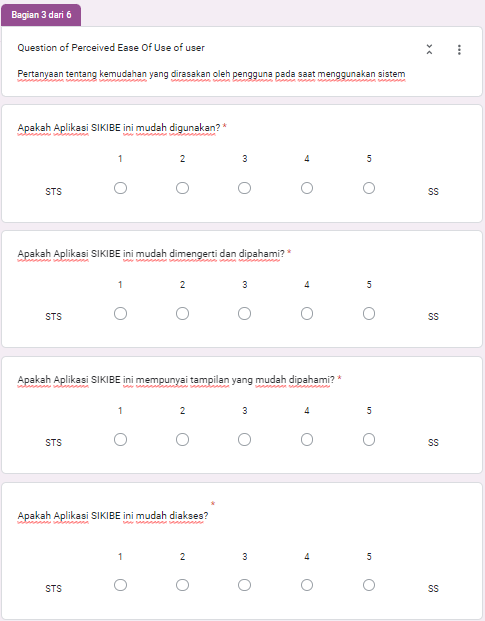
Diagram

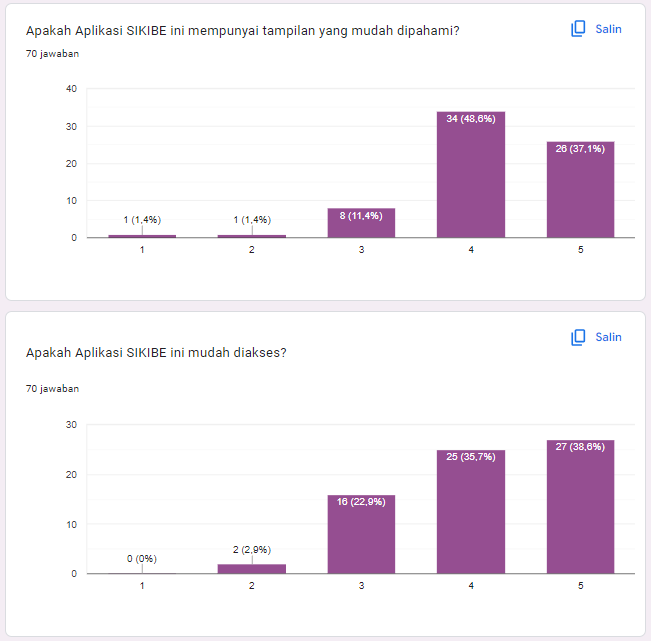
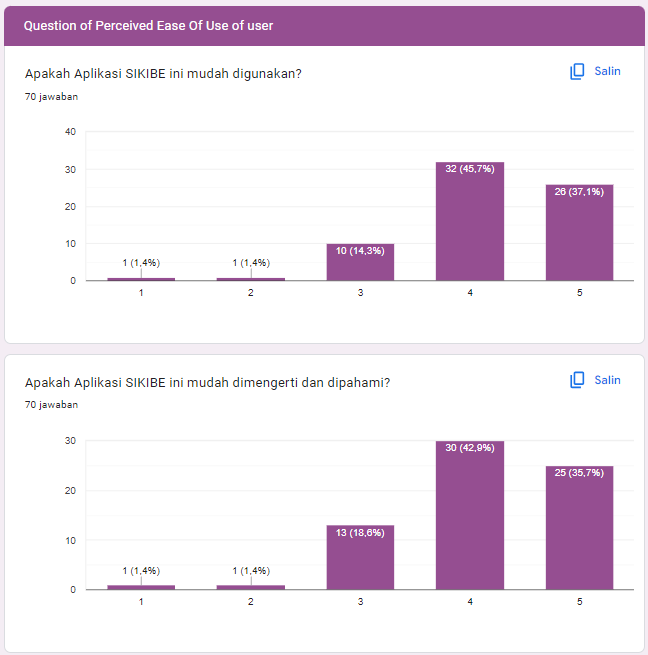
****

****

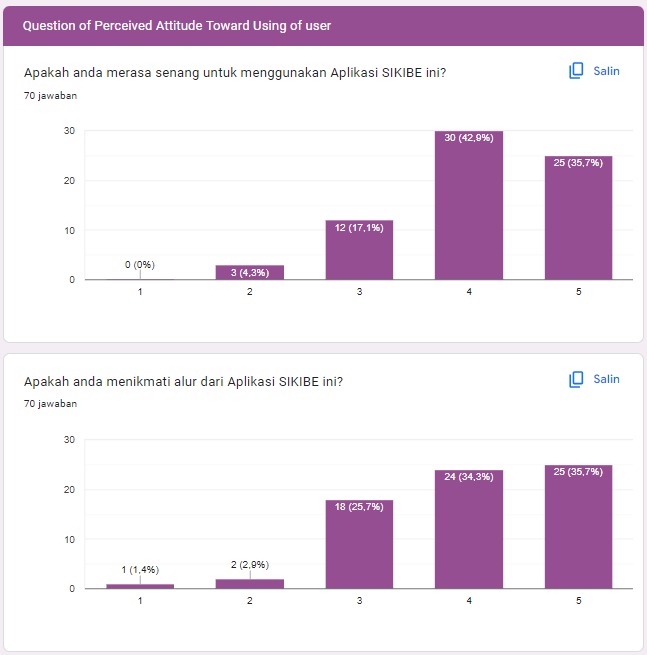
****

****

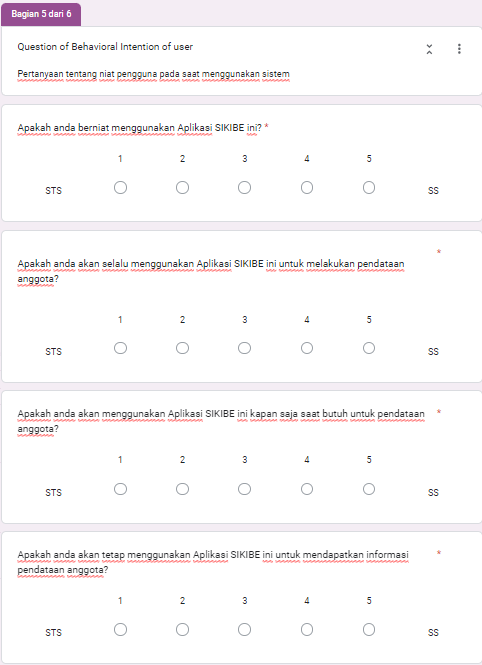
****

****

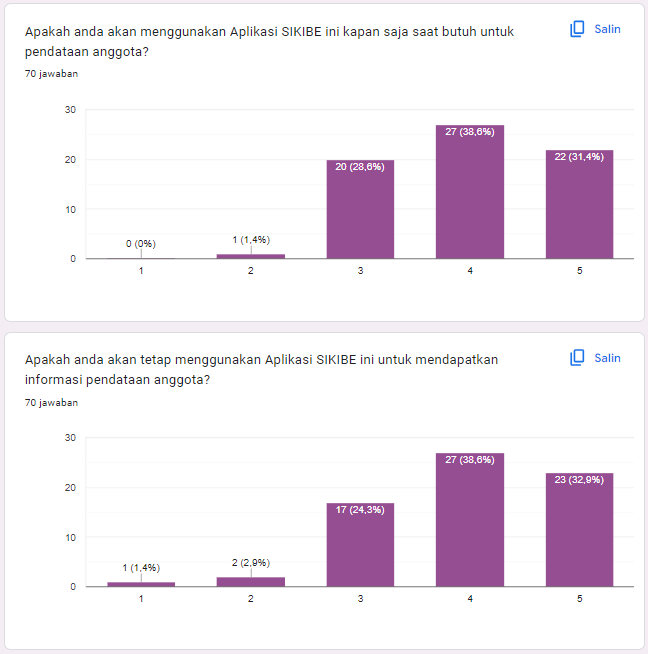
****

****

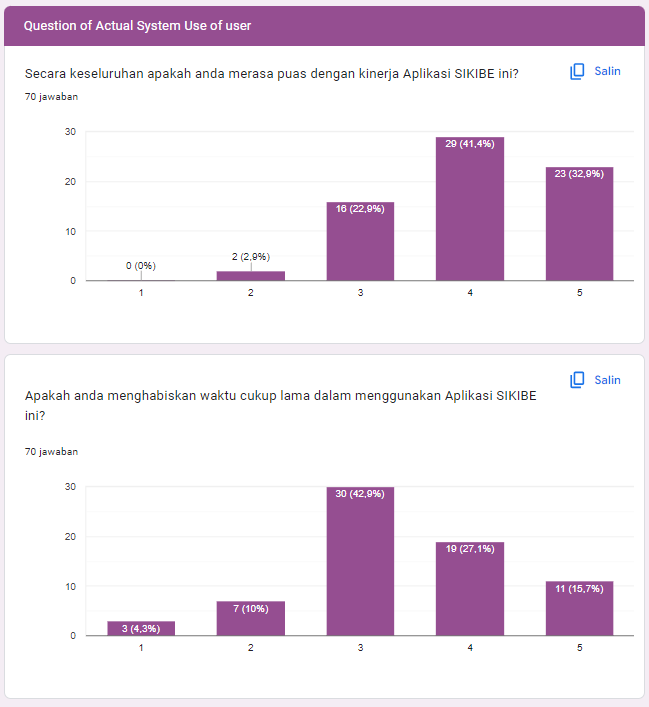
****

****

****

****

****

****

**LAMPIRAN 6:** ToR ( Term Of Reference)

Sebelum Melaksanakan penelitian skripsi penulis melakukan pengambilan data dengan yaitu observasi, wawancara, dan studi pustaka. Setelah mendapatkan data perihal lokasi penelitian dan permasalahan yang ada, serta disetujui oleh instansi tempat penelitian di SMK Negeri 7 Baleendah. Penelitian berfokus pada kendala pencarian data anggota koperasi, pencatatan transaksi simpan pinjam dan pelaporan yang masih dilakukan konvensional menggunakan buku besar.

Penelitian ini bertujuan untuk membangun Aplikasi SIKIBE (Sistem Informasi Koperasi be7) dengan fitur pencarian anggota, pencatatan transaksi dan pelaporan koperasi bisa dilakukan secara online oleh petugas koperasi. Bahasa pemrograman yang digunakan yaitu PHP, *Visual Studio Code* (VS Code) dan database menggunakan MySQL dengan penerapan *framework* *Laravel*. Untuk memastikan fokus penelitian adapun batasan masalah yang ditetapkan adalah sebagai berikut :

1. Fokus pada pembangunan Aplikasi SIKIBE (Sistem Informasi Koperasi Be7)
2. Aplikasi mencakup fitur admin dapat mengelola Pencarian data anggota, pencatatan transaksi dan pelaporan yang dilakukan oleh petugas koperasi
3. Aplikasi mencakup fitur *user* yang dapat melihat di Aplikasi SIKIBE yang dilakukan oleh admin.
4. Pengujian aplikasi dilakukan secara mandiri menggunakan beberapa *case* yang telah ditentukan.
5. Biaya tambahan seperti hosting atau pembayaran lainnya untuk aplikasi berada di luar tanggung jawab, peneliti hanya bertanggung jawab untuk menyediakan pendanaan awal pada tahap pembangunan aplikasi.

Baleendah, 6 Maret 2024

Mahasiswa Kepala Koperasi

**Dika Setia Budhi** **Tetet Sopiah, S.Pd.**

**LAMPIRAN 7:** Listing Program

|  |
| --- |
| Login:  class Login extends SimplePage  {      use InteractsWithFormActions;      use WithRateLimiting;      /\*\*       \* @var view-string       \*/      protected static string $view = 'filament.pages.auth.login';      /\*\*       \* @var array<string, mixed> | null       \*/      public ?array $data = [];      public function mount(): void      {          if (Filament::auth()->check()) {              redirect()->intended(Filament::getUrl());          }          $this->form->fill();      }      public function authenticate(): ?LoginResponse      {          try {              $this->rateLimit(5);          } catch (TooManyRequestsException $exception) {              Notification::make()                  ->title(\_\_('filament-panels::pages/auth/login.notifications.throttled.title', [                      'seconds' => $exception->secondsUntilAvailable,                      'minutes' => ceil($exception->secondsUntilAvailable / 60),                  ]))                  ->body(array\_key\_exists('body', \_\_('filament-panels::pages/auth/login.notifications.throttled') ?: []) ? \_\_('filament-panels::pages/auth/login.notifications.throttled.body', [                      'seconds' => $exception->secondsUntilAvailable,                      'minutes' => ceil($exception->secondsUntilAvailable / 60),                  ]) : null)                  ->danger()                  ->send();              return null;          }          $data = $this->form->getState();          if (! Filament::auth()->attempt($this->getCredentialsFromFormData($data), $data['remember'] ?? false)) {              $this->throwFailureValidationException();          }          $user = Filament::auth()->user();          if (              ($user instanceof FilamentUser) &&              (! $user->canAccessPanel(Filament::getCurrentPanel()))          ) {              Filament::auth()->logout();              $this->throwFailureValidationException();          }          session()->regenerate();          return app(LoginResponse::class);      }      protected function throwFailureValidationException(): never      {          throw ValidationException::withMessages([              'data.email' => \_\_('filament-panels::pages/auth/login.messages.failed'),          ]);      }      public function form(Form $form): Form      {          return $form;      }      /\*\*       \* @return array<int | string, string | Form>       \*/      protected function getForms(): array      {          return [              'form' => $this->form(                  $this->makeForm()                      ->schema([                          $this->getEmailFormComponent(),                          $this->getPasswordFormComponent(),                          $this->getRememberFormComponent(),                      ])                      ->statePath('data'),              ),          ];      }      protected function getEmailFormComponent(): Component      {          return TextInput::make('email')              ->label(\_\_('filament-panels::pages/auth/login.form.email.label'))              ->email()              ->required()              ->autocomplete()              ->autofocus()              ->extraInputAttributes(['tabindex' => 1]);      }      protected function getPasswordFormComponent(): Component      {          return TextInput::make('password')              ->label(\_\_('filament-panels::pages/auth/login.form.password.label'))              ->hint(filament()->hasPasswordReset() ? new HtmlString(Blade::render('<x-filament::link :href="filament()->getRequestPasswordResetUrl()"> {{ \_\_(\'filament-panels::pages/auth/login.actions.request\_password\_reset.label\') }}</x-filament::link>')) : null)              ->password()              ->revealable(filament()->arePasswordsRevealable())              ->autocomplete('current-password')              ->required()              ->extraInputAttributes(['tabindex' => 2]);      }      protected function getRememberFormComponent(): Component      {          return Checkbox::make('remember')              ->label(\_\_('filament-panels::pages/auth/login.form.remember.label'));      }      public function registerAction(): Action      {          return Action::make('register')              ->link()              ->label(\_\_('filament-panels::pages/auth/login.actions.register.label'))              ->url(filament()->getRegistrationUrl());      }      public function getTitle(): string | Htmlable      {          return \_\_('filament-panels::pages/auth/login.title');      }      public function getHeading(): string | Htmlable      {          return \_\_('filament-panels::pages/auth/login.heading');      }      /\*\*       \* @return array<Action | ActionGroup>       \*/      protected function getFormActions(): array      {          return [              $this->getAuthenticateFormAction(),          ];      }      protected function getAuthenticateFormAction(): Action      {          return Action::make('authenticate')              ->label(\_\_('filament-panels::pages/auth/login.form.actions.authenticate.label'))              ->submit('authenticate');      }      protected function hasFullWidthFormActions(): bool      {          return true;      }      /\*\*       \* @param  array<string, mixed>  $data       \* @return array<string, mixed>       \*/      protected function getCredentialsFromFormData(array $data): array      {          return [              'email' => $data['email'],              'password' => $data['password'],          ];      }        public function getExtraBodyAttributes(): array      {          return [              'style' => 'background-image: url(' . asset('bg-1.jpeg') . '); background-repeat: no-repeat; background-size: cover;'          ];      }  }  Register:  class Register extends SimplePage  {      use CanUseDatabaseTransactions;      use InteractsWithFormActions;      use WithRateLimiting;      /\*\*       \* @var view-string       \*/      protected static string $view = 'filament.pages.auth.register';      /\*\*       \* @var array<string, mixed> | null       \*/      public ?array $data = [];      protected string $userModel;      public function mount(): void      {          if (Filament::auth()->check()) {              redirect()->intended(Filament::getUrl());          }          $this->callHook('beforeFill');          $this->form->fill();          $this->callHook('afterFill');      }      public function register(): ?RegistrationResponse      {          try {              $this->rateLimit(2);          } catch (TooManyRequestsException $exception) {              Notification::make()                  ->title(\_\_('filament-panels::pages/auth/register.notifications.throttled.title', [                      'seconds' => $exception->secondsUntilAvailable,                      'minutes' => ceil($exception->secondsUntilAvailable / 60),                  ]))                  ->body(array\_key\_exists('body', \_\_('filament-panels::pages/auth/register.notifications.throttled') ?: []) ? \_\_('filament-panels::pages/auth/register.notifications.throttled.body', [                      'seconds' => $exception->secondsUntilAvailable,                      'minutes' => ceil($exception->secondsUntilAvailable / 60),                  ]) : null)                  ->danger()                  ->send();              return null;          }          $user = $this->wrapInDatabaseTransaction(function () {              $this->callHook('beforeValidate');              $data = $this->form->getState();              $this->callHook('afterValidate');              $data = $this->mutateFormDataBeforeRegister($data);              $this->callHook('beforeRegister');              $user = $this->handleRegistration($data);              $this->form->model($user)->saveRelationships();              $this->callHook('afterRegister');              return $user;          });          event(new Registered($user));          $this->sendEmailVerificationNotification($user);          Filament::auth()->login($user);          session()->regenerate();          return app(RegistrationResponse::class);      }      /\*\*       \* @param  array<string, mixed>  $data       \*/      protected function handleRegistration(array $data): Model      {          return $this->getUserModel()::create($data);      }      protected function sendEmailVerificationNotification(Model $user): void      {          if (! $user instanceof MustVerifyEmail) {              return;          }          if ($user->hasVerifiedEmail()) {              return;          }          if (! method\_exists($user, 'notify')) {              $userClass = $user::class;              throw new Exception("Model [{$userClass}] does not have a [notify()] method.");          }          $notification = app(VerifyEmail::class);          $notification->url = Filament::getVerifyEmailUrl($user);          $user->notify($notification);      }      public function form(Form $form): Form      {          return $form;      }      /\*\*       \* @return array<int | string, string | Form>       \*/      protected function getForms(): array      {          return [              'form' => $this->form(                  $this->makeForm()                      ->schema([                          $this->getNameFormComponent(),                          $this->getEmailFormComponent(),                          $this->getPasswordFormComponent(),                          $this->getPasswordConfirmationFormComponent(),                      ])                      ->statePath('data'),              ),          ];      }      protected function getNameFormComponent(): Component      {          return TextInput::make('name')              ->label(\_\_('filament-panels::pages/auth/register.form.name.label'))              ->required()              ->maxLength(255)              ->autofocus();      }      protected function getEmailFormComponent(): Component      {          return TextInput::make('email')              ->label(\_\_('filament-panels::pages/auth/register.form.email.label'))              ->email()              ->required()              ->maxLength(255)              ->unique($this->getUserModel());      }      protected function getPasswordFormComponent(): Component      {          return TextInput::make('password')              ->label(\_\_('filament-panels::pages/auth/register.form.password.label'))              ->password()              ->revealable(filament()->arePasswordsRevealable())              ->required()              ->rule(Password::default())              ->dehydrateStateUsing(fn ($state) => Hash::make($state))              ->same('passwordConfirmation')              ->validationAttribute(\_\_('filament-panels::pages/auth/register.form.password.validation\_attribute'));      }      protected function getPasswordConfirmationFormComponent(): Component      {          return TextInput::make('passwordConfirmation')              ->label(\_\_('filament-panels::pages/auth/register.form.password\_confirmation.label'))              ->password()              ->revealable(filament()->arePasswordsRevealable())              ->required()              ->dehydrated(false);      }      public function loginAction(): Action      {          return Action::make('login')              ->link()              ->label(\_\_('filament-panels::pages/auth/register.actions.login.label'))              ->url(filament()->getLoginUrl());      }      protected function getUserModel(): string      {          if (isset($this->userModel)) {              return $this->userModel;          }          /\*\* @var SessionGuard $authGuard \*/          $authGuard = Filament::auth();          /\*\* @var EloquentUserProvider $provider \*/          $provider = $authGuard->getProvider();          return $this->userModel = $provider->getModel();      }      public function getTitle(): string | Htmlable      {          return \_\_('filament-panels::pages/auth/register.title');      }      public function getHeading(): string | Htmlable      {          return \_\_('filament-panels::pages/auth/register.heading');      }      /\*\*       \* @return array<Action | ActionGroup>       \*/      protected function getFormActions(): array      {          return [              $this->getRegisterFormAction(),          ];      }      public function getRegisterFormAction(): Action      {          return Action::make('register')              ->label(\_\_('filament-panels::pages/auth/register.form.actions.register.label'))              ->submit('register');      }      protected function hasFullWidthFormActions(): bool      {          return true;      }      /\*\*       \* @param  array<string, mixed>  $data       \* @return array<string, mixed>       \*/      protected function mutateFormDataBeforeRegister(array $data): array      {          $data['role'] = UserRole::Anggota;          return $data;      }        public function getExtraBodyAttributes(): array      {          return [              'style' => 'background-image: url(' . asset('bg-1.jpeg') . '); background-repeat: no-repeat; background-size: cover;'          ];      }  }  Dashboard:  class Dashboard extends Page  {      protected static string $routePath = '/';      protected static ?int $navigationSort = -2;      /\*\*       \* @var view-string       \*/      protected static string $view = 'filament-panels::pages.dashboard';      public static function getNavigationLabel(): string      {          return static::$navigationLabel ??              static::$title ??              \_\_('filament-panels::pages/dashboard.title');      }      public static function getNavigationIcon(): string | Htmlable | null      {          return static::$navigationIcon              ?? FilamentIcon::resolve('panels::pages.dashboard.navigation-item')              ?? (Filament::hasTopNavigation() ? 'heroicon-m-home' : 'heroicon-o-home');      }      public static function getRoutePath(): string      {          return static::$routePath;      }      /\*\*       \* @return array<class-string<Widget> | WidgetConfiguration>       \*/      public function getWidgets(): array      {          return [              BelumBayarPinjaman::class,              BelumBayarIuran::class          ];      }      /\*\*       \* @return array<class-string<Widget> | WidgetConfiguration>       \*/      public function getVisibleWidgets(): array      {          return $this->filterVisibleWidgets($this->getWidgets());      }      /\*\*       \* @return int | string | array<string, int | string | null>       \*/      public function getColumns(): int | string | array      {          return 2;      }      public function getTitle(): string | Htmlable      {          return static::$title ?? \_\_('filament-panels::pages/dashboard.title');      }  }  Simpanan:  class DepositResource extends Resource  {      protected static ?string $model = Deposit::class;      protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';      public static function getNavigationGroup(): ?string      {          return trans('admin.menu.main');      }      public static function getModelLabel(): string      {          return trans('admin.resources.deposit.label');      }      public static function form(Form $form): Form      {          return $form              ->schema([                  Forms\Components\Grid::make()->schema([                      Forms\Components\Section::make([                          Forms\Components\Select::make('customer\_id')                              ->relationship('customer', 'name')                              ->label(trans('admin.resources.deposit.fields.customer'))                              ->translateLabel()                              ->searchable()                              ->preload()                              ->helperText(fn ($record) => (empty($record)) ? '' : new HtmlString('Current Balance: <strong><i>Rp ' . (number\_format($record?->customer?->balance) ?? 0) . '</i></strong>'))                              ->required(),                          Forms\Components\Select::make('type')                              ->options(fn () => DepositType::class)                              ->label(trans('admin.resources.deposit.fields.type'))                              ->translateLabel()                              ->searchable()                              ->preload()                              ->live(true)                              ->afterStateUpdated(function ($state, $set) {                                  $total = 0;                                  if ($state == DepositType::Pokok->value) {                                      $total = config('coop.simpanan\_pokok\_awal');                                  }                                  if ($state == DepositType::Wajib->value) {                                      $total = config('coop.simpanan\_wajib\_bulanan');                                  }                                  $set('subtotal', number\_format($total));                                  $set('total', number\_format($total));                              })                              ->required(),                          Forms\Components\Select::make('status')                              ->options(fn () => DepositStatus::class)                              ->searchable()                              ->preload()                              ->visibleOn(['view'])                              ->required(),                      ]),                  ])                      ->columnSpan(2),                  Forms\Components\Grid::make()->schema([                      Forms\Components\Section::make([                          Forms\Components\TextInput::make('subtotal')                              ->mask(RawJs::make('$money($input)'))                              ->prefix('Rp ')                              ->stripCharacters(',')                              ->afterStateUpdated(function ($state, $set, $get) {                                  $currentState = (float)str($state)->replace(',', '')->value;                                  $currentFee = (float)str($get('fee'))->replace(',', '')->value;                                  $total = $currentFee + $currentState;                                  $set('total', number\_format($total));                              })                              ->live(true)                              ->readonly(fn ($get) => in\_array($get('type'), [DepositType::Pokok->value, DepositType::Wajib->value]))                              ->visible(fn ($get) => !in\_array($get('type'), [DepositType::Pokok->value, DepositType::Wajib->value]))                              ->numeric(),                          Forms\Components\TextInput::make('fee')                              ->mask(RawJs::make('$money($input)'))                              ->label(trans('admin.resources.deposit.fields.fee'))                              ->translateLabel()                              ->prefix('Rp ')                              ->stripCharacters(',')                              ->live(true)                              ->afterStateUpdated(function ($state, $set, $get) {                                  $currentState = (float)str($state)->replace(',', '')->value;                                  $currentSubtotal = (float)str($get('subtotal'))->replace(',', '')->value;                                  $total = $currentSubtotal + $currentState;                                  $set('total', number\_format($total));                              })                              ->visible(fn ($get) => !in\_array($get('type'), [DepositType::Pokok->value, DepositType::Wajib->value]))                              ->numeric(),                          Forms\Components\TextInput::make('total')                              ->mask(RawJs::make('$money($input)'))                              ->prefix('Rp ')                              ->stripCharacters(',')                              ->readonly(fn ($get) => in\_array($get('type'), [DepositType::Pokok->value, DepositType::Wajib->value]))                              ->numeric(),                      ]),                  ])                      ->columnSpan(1),                  Forms\Components\Textarea::make('notes')                      ->label(trans('admin.resources.deposit.fields.notes'))                      ->translateLabel()                      ->columnSpanFull(),              ])              ->columns(3);      }      public static function table(Table $table): Table      {          $whitelist = [UserRole::Superadmin, UserRole::Admin, UserRole::Manajer];          $customerIds = auth()->user()->customer?->id;          return $table              ->modifyQueryUsing(                  fn ($query) => $query                      ->when(                          !in\_array(auth()->user()->role, $whitelist),                          fn ($query) => $query->whereIn('customer\_id', [$customerIds])                      )              )              ->defaultSort('fiscal\_date', 'desc')              ->columns([                  Tables\Columns\TextColumn::make('no')                      ->rowIndex(),                  Tables\Columns\TextColumn::make('customer.name')                      ->label(trans('admin.resources.deposit.fields.customer'))                      ->translateLabel()                      ->numeric()                      ->sortable(),                  Tables\Columns\TextColumn::make('type')                      ->label(trans('admin.resources.deposit.fields.type'))                      ->translateLabel()                      ->badge()                      ->tooltip(fn ($record): string => match ($record->type) {                          DepositType::Pokok => 'Iuran wajib anggota 1x',                          DepositType::Wajib => 'Setor wajib anggota',                          DepositType::Sukarela => 'Setor simpanan',                      })                      ->searchable(),                  Tables\Columns\TextColumn::make('status')                      ->badge()                      ->searchable(),                  Tables\Columns\TextColumn::make('subtotal')                      ->numeric()                      ->badge()                      ->tooltip('Total nilai masuk')                      ->color('success')                      ->prefix('Rp ')                      ->sortable(),                  Tables\Columns\TextColumn::make('fee')                      ->label(trans('admin.resources.deposit.fields.fee'))                      ->translateLabel()                      ->numeric()                      ->color('danger')                      ->prefix('Rp ')                      ->sortable(),                  Tables\Columns\TextColumn::make('total')                      ->prefix('Rp ')                      ->numeric()                      ->badge()                      ->color('warning')                      ->tooltip('Total tagihan harus dibayar')                      ->sortable(),                  Tables\Columns\TextColumn::make('fiscal\_date')                      ->dateTime('d F Y H:i')                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: false),                  Tables\Columns\TextColumn::make('created\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),                  Tables\Columns\TextColumn::make('updated\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),                  Tables\Columns\TextColumn::make('deleted\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),              ])              ->filters([                  Filter::make('created\_at')                      ->form([                          DatePicker::make('dari tanggal'),                          DatePicker::make('sampai tanggal'),                      ])                      ->query(function (Builder $query, array $data): Builder {                          return $query                              ->when(                                  $data['dari tanggal'],                                  fn (Builder $query, $date): Builder => $query->whereDate('created\_at', '>=', $date),                              )                              ->when(                                  $data['sampai tanggal'],                                  fn (Builder $query, $date): Builder => $query->whereDate('created\_at', '<=', $date),                              );                      }),                  Tables\Filters\TrashedFilter::make(),                  Tables\Filters\SelectFilter::make('type')                      ->searchable()                      ->preload()                      ->options(fn () => DepositType::class),                  Tables\Filters\SelectFilter::make('plan')                      ->searchable()                      ->preload()                      ->options(fn () => DepositPlan::class),                  Tables\Filters\SelectFilter::make('status')                      ->searchable()                      ->preload()                      ->options(fn () => DepositStatus::class),              ])              ->actions([                  Tables\Actions\ViewAction::make(),              ])              ->headerActions([                  Actions\Tables\ExportAction::make()->exports([                      Exports\ExcelExport::make('deposits')                          ->fromTable()                          ->except([                              'no'                          ])                  ])                      ->label('Unduh')              ])              ->bulkActions([                  //              ]);      }      public static function getRelations(): array      {          return [              //          ];      }      public static function getPages(): array      {          return [              'index' => Pages\ListDeposits::route('/'),              'create' => Pages\CreateDeposit::route('/create'),              'view' => Pages\ViewDeposit::route('/{record}'),              'edit' => Pages\EditDeposit::route('/{record}/edit'),          ];      }      public static function getEloquentQuery(): Builder      {          return parent::getEloquentQuery()              ->withoutGlobalScopes([                  SoftDeletingScope::class,              ]);      }  }  Pinjaman:  class LoanResource extends Resource  {      protected static ?string $model = Loan::class;      protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';      public static function getNavigationGroup(): ?string      {          return trans('admin.menu.main');      }      public static function getModelLabel(): string      {          return trans('admin.resources.loan.label');      }      public static function form(Form $form): Form      {          return $form              ->schema([                  Forms\Components\Grid::make()->schema([                      Forms\Components\Section::make([                          Forms\Components\Select::make('customer\_id')                              ->relationship('customer', 'name')                              ->label(trans('admin.resources.loan.fields.customer'))                              ->translateLabel()                              ->searchable()                              ->preload()                              ->helperText(fn ($record) => (empty($record)) ? '' : new HtmlString('Current Balance: <strong><i>Rp '. (number\_format($record?->customer?->balance) ?? 0) .'</i></strong>'))                              ->required(),                          Forms\Components\TextInput::make('instalment')                              ->required()                              ->label(trans('admin.resources.loan.fields.instalment'))                              ->translateLabel()                              ->step(1)                              ->numeric(),                          Forms\Components\Select::make('status')                              ->options(fn () => LoanStatus::class)                              ->searchable()                              ->preload()                              ->visibleOn(['view'])                              ->required(),                      ]),                  ])                  ->columnSpan(2),                    Forms\Components\Grid::make()->schema([                      Forms\Components\Section::make([                          Forms\Components\TextInput::make('subtotal')                              ->mask(RawJs::make('$money($input)'))                              ->prefix('Rp ')                              ->stripCharacters(',')                              ->afterStateUpdated(function ($state, $set, $get) {                                  $currentState = (double)str($state)->replace(',', '')->value;                                  // $currentFee = (double)str($get('fee'))->replace(',', '')->value;                                    $currentFee = $currentState\*config('coop.potongan\_pinjaman', 0.01);                                  $set('fee', number\_format($currentFee));                                  $total = $currentState-$currentFee;                                  $set('total', number\_format($total));                              })                              ->live(true)                              ->numeric(),                          Forms\Components\TextInput::make('fee')                              ->mask(RawJs::make('$money($input)'))                              ->prefix('Rp ')                              ->label(trans('admin.resources.loan.fields.fee'))                              ->translateLabel()                              ->stripCharacters(',')                              ->live()                              ->afterStateUpdated(function ($state, $set, $get) {                                  $currentState = (double)str($state)->replace(',', '')->value;                                  $currentSubtotal = (double)str($get('subtotal'))->replace(',', '')->value;                                  $total = $currentSubtotal - $currentState;                                  $set('total', number\_format($total));                              })                              ->lazy(true)                              ->numeric(),                          Forms\Components\TextInput::make('total')                              ->mask(RawJs::make('$money($input)'))                              ->prefix('Rp ')                              ->stripCharacters(',')                              ->readonly()                              ->numeric(),                      ]),                  ])                  ->columnSpan(1),                  Forms\Components\Textarea::make('notes')                      ->label(trans('admin.resources.loan.fields.notes'))                      ->translateLabel()                      ->columnSpanFull(),              ])              ->columns(3);      }      public static function table(Table $table): Table      {          $whitelist = [UserRole::Superadmin, UserRole::Admin, UserRole::Manajer];          $customerIds = auth()->user()->customer?->id;          return $table              ->modifyQueryUsing(fn ($query) => $query                  ->when(                      !in\_array(auth()->user()->role, $whitelist),                      fn ($query) => $query->whereIn('customer\_id', [$customerIds])                  )              )              ->defaultSort('fiscal\_date', 'desc')              ->columns([                  Tables\Columns\TextColumn::make('no')                      ->rowIndex(),                  Tables\Columns\TextColumn::make('customer.name')                      ->numeric()                      ->label(trans('admin.resources.loan.fields.customer'))                      ->translateLabel()                      ->sortable(),                  Tables\Columns\TextColumn::make('status')                      ->badge()                      ->searchable(),                  Tables\Columns\TextColumn::make('instalment')                      ->numeric()                      ->label(trans('admin.resources.loan.fields.instalment'))                      ->translateLabel()                      ->sortable(),                  Tables\Columns\TextColumn::make('subtotal')                      ->prefix('Rp ')                      ->numeric()                      ->badge()                      ->color('success')                      ->tooltip('Total pinjaman')                      ->sortable(),                  Tables\Columns\TextColumn::make('fee')                      ->prefix('Rp ')                      ->label(trans('admin.resources.loan.fields.fee'))                      ->translateLabel()                      ->numeric()                      ->color('danger')                      ->sortable(),                  Tables\Columns\TextColumn::make('total')                      ->prefix('Rp ')                      ->numeric()                      ->badge()                      ->color('warning')                      ->tooltip('Total tagihan harus dibayar')                      ->sortable(),                  Tables\Columns\TextColumn::make('fiscal\_date')                      ->dateTime('d F Y H:i')                      ->label(trans('admin.resources.loan.fields.fiscal\_date'))                      ->translateLabel()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: false),                  Tables\Columns\TextColumn::make('created\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),                  Tables\Columns\TextColumn::make('updated\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),                  Tables\Columns\TextColumn::make('deleted\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),              ])              ->filters([                  Filter::make('created\_at')                      ->form([                          DatePicker::make('dari tanggal'),                          DatePicker::make('sampai tanggal'),                      ])                      ->query(function (Builder $query, array $data): Builder {                          return $query                              ->when(                                  $data['dari tanggal'],                                  fn (Builder $query, $date): Builder => $query->whereDate('created\_at', '>=', $date),                              )                              ->when(                                  $data['sampai tanggal'],                                  fn (Builder $query, $date): Builder => $query->whereDate('created\_at', '<=', $date),                              );                      }),                  Tables\Filters\TrashedFilter::make(),                  Tables\Filters\SelectFilter::make('status')                      ->searchable()                      ->preload()                      ->options(fn () => LoanStatus::class),              ])              ->actions([                  Tables\Actions\ViewAction::make(),              ])              ->headerActions([                  Actions\Tables\ExportAction::make()->exports([                      Exports\ExcelExport::make('loans')                          ->fromTable()                          ->except([                              'no'                          ])                  ])                  ->label('Unduh')              ])              ->bulkActions([                  //              ]);      }      public static function getRelations(): array      {          return [              InstalmentsRelationManager::class          ];      }      public static function getPages(): array      {          return [              'index' => Pages\ListLoans::route('/'),              'create' => Pages\CreateLoan::route('/create'),              'view' => Pages\ViewLoan::route('/{record}'),              'edit' => Pages\EditLoan::route('/{record}/edit'),          ];      }      public static function getEloquentQuery(): Builder      {          return parent::getEloquentQuery()              ->withoutGlobalScopes([                  SoftDeletingScope::class,              ]);      }  }  Anggota:  class CustomerResource extends Resource  {      protected static ?string $model = Customer::class;      protected static ?string $navigationIcon = 'heroicon-o-user-plus';      public static function getNavigationGroup(): ?string      {          return trans('admin.menu.masterData');      }      public static function getModelLabel(): string      {          return trans('admin.resources.customer.label');      }      public static function shouldRegisterNavigation(): bool      {          return in\_array(auth()->user()->role, [UserRole::Superadmin, UserRole::Admin]);      }      public static function form(Form $form): Form      {          return $form              ->schema([                  Grid::make()->columns(5)                      ->schema([                          Grid::make()->schema([                              Forms\Components\Section::make([                                  Forms\Components\TextInput::make('name')                                      ->label(trans('admin.resources.customer.fields.name'))                                      ->translateLabel()                                      ->required()                                      ->maxLength(255),                                  Forms\Components\TextInput::make('phone')                                      ->label(trans('admin.resources.customer.fields.phone'))                                      ->translateLabel()                                      ->tel()                                      ->numeric()                                      ->default(null),                                  Forms\Components\TextInput::make('email')                                      ->label(trans('admin.resources.customer.fields.email'))                                      ->translateLabel()                                      ->email()                                      ->maxLength(255)                                      ->default(null),                                  Forms\Components\TextInput::make('password')                                      ->label(trans('admin.resources.customer.fields.password'))                                      ->translateLabel()                                      ->password()                                      ->maxLength(255)                                      ->default(null)                                      ->visibleOn(['create', 'edit']),                              ])->columnSpan(3),                          ])->columnSpan(3),                          Grid::make()->schema([                              Forms\Components\Section::make([                                  Forms\Components\TextInput::make('code')                                      ->label(trans('admin.resources.customer.fields.code'))                                      ->translateLabel()                                      ->live()                                      ->afterStateUpdated(function ($state, $set) {                                          $code = str($state)->slug('\_')->upper();                                          $set('code', $code);                                      })                                      ->required()                                      ->maxLength(255),                                  Forms\Components\DatePicker::make('birthdate')                                      ->label(trans('admin.resources.customer.fields.birthdate'))                                      ->translateLabel()                                      ->required(),                                  Forms\Components\TextInput::make('birthplace')                                      ->label(trans('admin.resources.customer.fields.birthplace'))                                      ->translateLabel()                                      ->required()                                      ->maxLength(255),                              ])->columnSpan(2),                          ])->columnSpan(2),                          Grid::make()->schema([                              Forms\Components\Section::make([                                  Forms\Components\RichEditor::make('address')                                      ->label(trans('admin.resources.customer.fields.address'))                                      ->translateLabel()                              ])->columnSpanFull(),                          ])->columnSpanFull(),                      ])              ]);      }      public static function table(Table $table): Table      {          $whitelist = [UserRole::Superadmin, UserRole::Admin, UserRole::Manajer];          $customerIds = auth()->user()->customer?->id;          return $table              ->modifyQueryUsing(fn ($query) => $query                  ->when(                      !in\_array(auth()->user()->role, $whitelist),                      fn ($query) => $query->whereIn('id', [$customerIds])                  )              )              ->columns([                  Tables\Columns\TextColumn::make('no')                      ->rowIndex(),                  Tables\Columns\TextColumn::make('name')                      ->label(trans('admin.resources.customer.fields.name'))                      ->translateLabel()                      ->description(fn ($record) => str($record?->code)->limit(20))                      ->wrap()                      ->searchable(),                  Tables\Columns\TextColumn::make('email')                      ->label(trans('admin.resources.customer.fields.email'))                      ->translateLabel()                      ->searchable(),                  Tables\Columns\TextColumn::make('phone')                      ->label(trans('admin.resources.customer.fields.phone'))                      ->translateLabel()                      ->searchable(),                  Tables\Columns\TextColumn::make('total\_deposits')                      ->label(trans('admin.resources.customer.fields.total\_deposits'))                      ->translateLabel()                      ->prefix('Rp ')                      ->badge()                      ->color('success')                      ->numeric(),                  Tables\Columns\TextColumn::make('total\_loans')                      ->label(trans('admin.resources.customer.fields.total\_loans'))                      ->translateLabel()                      ->prefix('Rp ')                      ->badge()                      ->color('danger')                      ->numeric(),                  Tables\Columns\TextColumn::make('birthdate')                      ->label(trans('admin.resources.customer.fields.birthdate'))                      ->translateLabel()                      ->date()                      ->toggleable(isToggledHiddenByDefault: true)                      ->sortable(),                  Tables\Columns\TextColumn::make('birthplace')                      ->label(trans('admin.resources.customer.fields.birthplace'))                      ->translateLabel()                      ->toggleable(isToggledHiddenByDefault: true)                      ->searchable(),                  Tables\Columns\TextColumn::make('deleted\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),                  Tables\Columns\TextColumn::make('created\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),                  Tables\Columns\TextColumn::make('updated\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),              ])              ->filters([                  Filter::make('created\_at')                      ->form([                          DatePicker::make('dari tanggal'),                          DatePicker::make('sampai tanggal'),                      ])                      ->query(function (Builder $query, array $data): Builder {                          return $query                              ->when(                                  $data['dari tanggal'],                                  fn (Builder $query, $date): Builder => $query->whereDate('created\_at', '>=', $date),                              )                              ->when(                                  $data['sampai tanggal'],                                  fn (Builder $query, $date): Builder => $query->whereDate('created\_at', '<=', $date),                              );                      }),                  Tables\Filters\TrashedFilter::make(),              ])              ->actions([                  Tables\Actions\ViewAction::make(),                  // Tables\Actions\EditAction::make(),              ])              ->bulkActions([                  // Tables\Actions\BulkActionGroup::make([                  //     Tables\Actions\DeleteBulkAction::make(),                  //     Tables\Actions\ForceDeleteBulkAction::make(),                  //     Tables\Actions\RestoreBulkAction::make(),                  // ]),              ]);      }      public static function getRelations(): array      {          return [              //          ];      }      public static function getPages(): array      {          return [              'index' => Pages\ListCustomers::route('/'),              'create' => Pages\CreateCustomer::route('/create'),              'view' => Pages\ViewCustomer::route('/{record}'),              'edit' => Pages\EditCustomer::route('/{record}/edit'),          ];      }      public static function getEloquentQuery(): Builder      {          return parent::getEloquentQuery()              ->withoutGlobalScopes([                  SoftDeletingScope::class,              ]);      }  }  Pengguna:  class UserResource extends Resource  {      protected static ?string $model = User::class;      protected static ?string $navigationIcon = 'heroicon-o-user-group';      // protected static ?string $navigationGroup = 'Settings';      public static function getNavigationGroup(): ?string      {          return trans('admin.menu.settings');      }      public static function getModelLabel(): string      {          return trans('admin.resources.user.label');      }      public static function shouldRegisterNavigation(): bool      {          return in\_array(auth()->user()->role, [UserRole::Superadmin, UserRole::Admin]);      }      public static function form(Form $form): Form      {          return $form              ->schema([                  Forms\Components\Section::make([                      Forms\Components\Select::make('role')                          ->options( fn () => UserRole::class)                          ->searchable()                          ->preload()                          ->required(),                      Forms\Components\TextInput::make('name')                          ->required()                          ->maxLength(255),                      Forms\Components\TextInput::make('email')                          ->email()                          ->required()                          ->maxLength(255),                      Forms\Components\TextInput::make('password')                          ->password()                          ->dehydrateStateUsing(fn ($state) => Hash::make($state))                          ->dehydrated(fn ($state) => filled($state))                          ->required(fn (string $context): bool => $context === 'create'),                  ])              ]);      }      public static function table(Table $table): Table      {          return $table              ->columns([                  Tables\Columns\TextColumn::make('index')->state(                      static function ($livewire, $rowLoop): string {                          return (string) (                              $rowLoop->iteration +                              ($livewire->getTableRecordsPerPage() \* (                                  $livewire->getTablePage() - 1                              ))                          );                      }                  ),                  Tables\Columns\TextColumn::make('role')                      ->badge()                      ->searchable(),                  Tables\Columns\TextColumn::make('name')                      ->searchable(),                  Tables\Columns\TextColumn::make('email')                      ->icon('heroicon-m-envelope')                      ->searchable(),                  Tables\Columns\TextColumn::make('created\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),                  Tables\Columns\TextColumn::make('updated\_at')                      ->dateTime()                      ->sortable()                      ->toggleable(isToggledHiddenByDefault: true),              ])              ->filters([                  Filter::make('created\_at')                      ->form([                          DatePicker::make('dari tanggal'),                          DatePicker::make('sampai tanggal'),                      ])                      ->query(function (Builder $query, array $data): Builder {                          return $query                              ->when(                                  $data['dari tanggal'],                                  fn (Builder $query, $date): Builder => $query->whereDate('created\_at', '>=', $date),                              )                              ->when(                                  $data['sampai tanggal'],                                  fn (Builder $query, $date): Builder => $query->whereDate('created\_at', '<=', $date),                              );                      }),              ])              ->actions([                  Tables\Actions\ViewAction::make(),                  // Tables\Actions\EditAction::make(),              ])              ->bulkActions([                  // Tables\Actions\BulkActionGroup::make([                  //     Tables\Actions\DeleteBulkAction::make()                  //         ->successNotification(                  //             Notification::make()                  //                 ->success()                  //                 ->title('User deleted')                  //                 ->body('The user has been deleted successfully.'),                  //         ),                  // ]),              ])              ->emptyStateActions([                  Tables\Actions\CreateAction::make(),              ]);      }      public static function getRelations(): array      {          return [              //          ];      }      public static function getPages(): array      {          return [              'index' => Pages\ListUsers::route('/'),              'create' => Pages\CreateUser::route('/create'),              'view' => Pages\ViewUser::route('/{record}'),              'edit' => Pages\EditUser::route('/{record}/edit'),          ];      }      public static function getEloquentQuery(): Builder      {          return parent::getEloquentQuery()->whereNotIn('email', ['superadmin@majalio.com']);      }  }  DB Simpanan:  class Deposit extends Model  {      use SoftDeletes;      protected $fillable = [          'customer\_id',          'customer\_balance\_before',          'customer\_balance\_after',          'status',          'type',          'plan',          'subtotal',          'fee',          'total',          'notes',          'fiscal\_date',      ];      protected $casts = [          'type' => DepositType::class,          'plan' => DepositPlan::class,          'status' => DepositStatus::class,      ];      public function customer(): BelongsTo      {          return $this->belongsTo(              Customer::class,              'customer\_id',              'id'          );      }      public function cancel()      {          $customer = $this->customer;          $customer->charge($this->subtotal);      }  }  DB Pinjaman:  class Loan extends Model  {      use SoftDeletes;      protected $fillable = [          'customer\_id',          'status',          'instalment',          'fee',          'subtotal',          'total',          'notes',          'fiscal\_date',      ];      protected $casts = [          'status' => LoanStatus::class,      ];      public function customer(): BelongsTo      {          return $this->belongsTo(              Customer::class,              'customer\_id',              'id'          );      }      public function instalments(): HasMany      {          return $this->hasMany(              LoanInstalment::class,              'loan\_id',              'id'          );      }      public function getCurrentTotalPaidAttribute()      {          return $this->instalments()?->where('status', LoanInstalmentStatus::Verified)?->sum('subtotal') ?? 0;      }  }  DB Anggota:  class Customer extends Model  {      use HasFactory, SoftDeletes;      protected $fillable = [          'user\_id',          'balance',          'code',          'name',          'email',          'phone',          'address',          'gender',          'birthdate',          'birthplace',      ];      public function user(): BelongsTo      {          return $this->belongsTo(              User::class,              'user\_id',              'id'          );      }      public function deposits(): HasMany      {          return $this->hasMany(              Deposit::class,              'customer\_id',              'id'          );      }      public function loans(): HasMany      {          return $this->hasMany(              Loan::class,              'customer\_id',              'id'          );      }      public function topup($amount): void      {          Log::notice('Adding balance Rp'. number\_format($amount) .' to ' . $this->name);          $this->balance += $amount;          $this->save();      }      public function charge($amount): void      {          Log::notice('Chargin balance Rp'. number\_format($amount) .' to ' . $this->name);          $this->balance -= $amount;          $this->save();      }      public function getTotalDepositsAttribute()      {          return $this->deposits()?->where('status', DepositStatus::Verified)?->sum('subtotal') ?? 0;      }      public function getTotalLoansAttribute()      {          return $this->loans()?->where('status', LoanStatus::Loaned)?->sum('subtotal') ?? 0;      }  }  DB Pengguna:  class User extends Authenticatable implements FilamentUser  {      use HasApiTokens, HasFactory, Notifiable;      protected $fillable = [          'role',          'name',          'email',          'password',      ];      protected $hidden = [          'password',          'remember\_token',      ];      protected $casts = [          'email\_verified\_at' => 'datetime',          'password' => 'hashed',          'role' => UserRole::class,      ];        public function canAccessPanel(Panel $panel): bool      {          return true;      }      public function customer(): HasOne      {          return $this->hasOne(              Customer::class,              'user\_id',              'id'          );      }  } |