

**PENGEMBANGAN APLIKASI PENDAFTARAN ANGGOTA
PERPUSTAKAN BERBASIS ANDROID MENGGUNAKAN MODEL
WATERFALL PADA DINAS PERPUSTAKAAN KOTA SAMARINDA**

SKRIPSI

Diajukan oleh:

Muhammad Ramadhani Saputra

2011102441087



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
JULI 2024**

**PENGEMBANGAN APLIKASI PENDAFTARAN ANGGOTA
PERPUSTAKAN BERBASIS ANDROID MENGGUNAKAN MODEL
WATERFALL PADA DINAS PERPUSTAKAAN KOTA SAMARINDA**

SKRIPSI

Diajukan Sebagai Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana Teknik
Informatika Fakultas Sains Dan Teknologi Universitas Muhammadiyah Kalimantan Timur

Diajukan oleh:

Muhammad Ramadhani Saputra

2011102441087



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
JULI 2024**

LEMBAR PERSETUJUAN

PENGEMBANGAN APLIKASI PENDAFTARAN ANGGOTA PERPUSTAKAN BERBASIS ANDROID MENGGUNAKAN MODEL WATERFALL PADA DINAS PERPUSTAKAAN KOTA SAMARINDA

SKRIPSI

Diajukan oleh:

Muhammad Ramadhani Saputra

2011102441087

Disetujui untuk diujikan

Pada tanggal 26 Juni 2024

Pembimbing



Abdul Rahim, S.Kom., M.Cs

NIDN: 0009047901

Mengetahui,

Koordinator Skripsi



Abdul Rahim, S.Kom., M.Cs

NIDN: 0009047901

LEMBAR PENGESAHAN

PENGEMBANGAN APLIKASI PENDAFTARAN ANGGOTA PERPUSTAKAN BERBASIS ANDROID MENGGUNAKAN MODEL WATERFALL PADA DINAS PERPUSTAKAAN KOTA SAMARINDA

SKRIPSI



Diajukan oleh:

Muhammad Ramadhani Saputra

2011102441087

Diseminarkan dan Diujikan

Pada Tanggal 05 Juli 2024

Penguji I	Penguji II
 <u>Sayekti Harits Suryawan, S.Kom, M.Kom</u> NIDN 1119048901	 <u>Abdul Rahim, S.Kom., M.Cs</u> NIDN 0009047901

Mengetahui,

Ketua

Program Studi Teknik Informatika

Arbansyah, S.Kom., M.TI

NIDN 1118019203

PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Ramadhani Saputra
NIM : 2011102441087
Program Studi : S1 Teknik Informatika
Judul Penelitian : Pengembangan Aplikasi Pendaftaran Anggota Perpustakaan Berbasis Android Menggunakan Model Waterfall Pada Dinas Perpustakaan Kota Samarinda

Menyatakan bahwa Skripsi yang saya tulis ini benar-benar hasil karya saya sendiri, dan bukan merupakan hasil plagiasi/falsifikasi/fabrikasi baik sebagian atau seluruhnya.

Atas pernyataan ini, saya siap menanggung resiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam skripsi saya ini, atau klaim dari pihak lain terhadap keaslian karya saya ini.

Samarinda 17 Juli 2024
Yang membuat pernyataan



Muhammad Ramadhani Saputra
NIM: 2011102441087

ABSTRAK

Penggunaan teknologi telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari dengan smartphone sebagai salah satu teknologi utama yang memungkinkan akses cepat dan mudah terhadap berbagai layanan dan informasi. Pendaftaran anggota di Dinas Perpustakaan dan Kearsipan Kota Samarinda saat ini masih dilakukan secara manual, yang menyebabkan proses yang kurang efisien. Penelitian ini bertujuan mengembangkan aplikasi pendaftaran anggota perpustakaan berbasis Android menggunakan framework Flutter untuk frontend, Laravel untuk backend, dan MySQL sebagai basis data. Metode pengembangan Waterfall digunakan untuk memastikan tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta operasional dan pemeliharaan berjalan dengan terstruktur. Dalam pengembangan ini, pemodelan data menggunakan UML (Unified Modelling Language) dengan Use Case Diagram, Class Diagram, dan Activity Diagram. Berdasarkan hasil penelitian dan pengembangan dapat disimpulkan bahwa, aplikasi ini berhasil mengatasi permasalahan dalam proses pendaftaran anggota perpustakaan yang sebelumnya dilakukan secara manual. Calon anggota dapat melakukan pendaftaran secara online tanpa harus mengunjungi perpustakaan berulang kali, menghemat waktu proses pendaftaran. Fitur pencarian buku yang diimplementasikan dalam aplikasi memudahkan anggota perpustakaan untuk mencari ketersediaan buku tanpa harus datang langsung ke perpustakaan, mengurangi kemungkinan kunjungan yang sia-sia karena buku yang dicari tidak tersedia.

Kata kunci : Aplikasi Android, Pendaftaran Anggota Perpustakaan, flutter,laravel MySQL,metode Waterfall

ABSTRACT

The use of technology has become an integral part of everyday life with smartphones as one of the main technologies that allow quick and easy access to various services and information. Member registration at the Samarinda City Library and Archives Office is currently still done manually, which causes a less efficient process. This research aims to develop an Android-based library member registration application using the Flutter framework for the frontend, Laravel for the backend, and MySQL as a database. The Waterfall development method is used to ensure that the stages of needs analysis, system design, implementation, testing, and operations and maintenance are structured. In this development, data modelling uses UML (Unified Modelling Language) with Use Case Diagram, Class Diagram, and Activity Diagram. Based on the results of research and development, it can be concluded that this application has succeeded in overcoming problems in the process of registering library members which were previously carried out manually. Prospective members can register online without having to visit the library repeatedly, saving time in the registration process. The book search feature implemented in the application makes it easy for library members to search for book availability without having to come directly to the library, reducing the possibility of wasted visits because the book they are looking for is not available.

Keywords: Android Application, Library Member Registration, flutter, laravel MySQL, Waterfall method

PRAKATA

Assalamu'alaikum Wr.Wb puji dan syukur penulis sampaikan kehadiran Allah SWT, karena berkat rahmat dan hidayah-Nya-lah saya dapat menyelesaikan skripsi yang berjudul "Pengembangan Aplikasi Pendaftaran Anggota Perpustakaan Berbasis Android Menggunakan Model Waterfall Pada Dinas Perpustakaan Kota Samarinda". Skripsi ini dibuat untuk memenuhi tugas akhir perkuliahan dan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Strata 1 di Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Muhammadiyah Kalimantan Timur.

Penulis menyadari bahwa skripsi masih jauh dari sempurna. Oleh karena itu, penulis berharap dapat belajar lebih banyak lagi dalam mengimplementasikan ilmu yang didapatkan. Skripsi ini tentunya tidak lepas dari bimbingan, masukan, dan arahan dari berbagai pihak. Oleh karena itu, pada kesempatan ini saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

- Bapak Abdul Rahim, S.Kom., M.Cs sebagai dosen pembimbing sekaligus penguji 2 yang telah meluangkan waktu memberikan dukungan, bimbingan, dan motivasi dalam menyelesaikan skripsi ini
- Bapak Sayekti Harits Suryawan, S.Kom, M.Kom sebagai dosen penguji 1 Fakultas Sains dan Teknologi Universitas Muhammadiyah Kalimantan Timur
- Bapak Arbansyah, S.Kom., M.TI sebagai Ketua Program Studi Teknik Informatika yang telah memberikan dukungan selama masa perkuliahan di Teknik Informatika Universitas Muhammadiyah Kalimantan Timur
- Orangtua tercinta, Ramly Djafar dan Malarifjana serta kakak saya Muhammad Wahyu Aditya, dan adik saya Anisa Rahmaniah, memberikan dukungan dan memotivasi dalam menyelesaikan skripsi ini.
- Bapak Andri sebagai Staff Otomasi Perpustakaan yang telah membantu menyelesaikan skripsi ini.
- Ibu Putry Desintia Ramadhani sebagai Staff layanan perpustakaan yang telah membantu menyelesaikan skripsi ini.

Semua pihak yang telah membantu dan tidak dapat disebutkan satu persatu. Saya berharap semoga Tuhan Yang Maha Esa mengaruniakan rahmat dan hidayah-Nya kepada mereka semua. Semoga skripsi ini dapat bermanfaat bagi kita semua, Aamiin. Wassalamu'alaikum Wr. Wb.

Samarinda 30 Juni 2024

Penyusun



Muhammad Ramadhani Saputra

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN KEASLIAN PENELITIAN.....	iii
ABSTRAK	iv
ABSTRACT	v
PRAKATA.....	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Batasan Masalah	5
1.3 Rumusan Masalah.....	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian.....	5
BAB II METODE PENELITIAN	7
2.1 Objek Penelitian	7
2.2 Alat dan Bahan	7
2.3 Metode Pengembangan Sistem.....	7
2.3.1 Analisis Kebutuhan (<i>Requirements Analysis</i>)	8
2.3.2 Perancangan (<i>Design</i>)	8
2.3.3 Implementasi (<i>Implementation</i>)	9
2.3.4 Pengujian (<i>Testing</i>).....	9
2.3.5 Pemeliharaan (<i>Maintenance</i>).....	9
2.4 Pengumpulan Data.....	10
BAB III HASIL DAN PEMBAHASAN.....	11
3.1 Analisis Kebutuhan.....	11
3.1.1 Hasil Studi Literatur	11
3.1.2 Hasil Observasi	12

3.1.3	Hasil Wawancara.....	12
3.2	Perancangan (Desain).....	13
3.4.1	Perancangan Sistem.....	13
3.4.2	Perancangan antarmuka.....	28
3.3	Implementasi (<i>Implementation</i>).....	38
3.3.1	Halaman Admin.....	38
3.3.2	Halaman Calon Anggota	43
3.4	Pengujian (<i>Testing</i>).....	51
3.4.1	Pengujian Sistem Website	51
3.4.2	Pengujian Sistem Android.....	52
3.4.3	Pengujian TAM	53
3.5	Pemeliharaan (<i>Maintenance</i>).....	56
BAB IV PENUTUP.....		57
4.1	Kesimpulan.....	57
4.2	Saran	57
DAFTAR RUJUKAN.....		58
RIWAYAT HIDUP.....		60
LAMPIRAN		61

DAFTAR TABEL

Tabel	Halaman
3. 1 Use Case Tabel Login Admin.....	14
3. 2 <i>Use Case</i> Tabel Register Calon Anggota	14
3. 3 <i>Use Case</i> Tabel Melihat Dashboard	15
3. 4 Use Case Tabel Mengelola Data Anggota Perpustakaan.....	15
3. 5 <i>Use Case</i> Tabel Mengelola Data Calon Anggota Perpustakaan.....	15
3. 6 <i>Use Case</i> Tabel Mengelola Data Buku.....	16
3. 7 <i>Use Case</i> Tabel Mengisi Formulir.....	16
3. 8 <i>Use Case</i> Tabel Melihat Halaman Profil.....	17
3. 9 <i>Use Case</i> Tabel Melihat Halaman Home	17
3. 10 <i>Use Case</i> Tabel Menyimpan Data	18
3. 11 <i>Use Case</i> Tabel Lihat Buku	18
3. 12 Pengujian Sistem Website	51
3. 13 Pengujian Sistem Android.....	52
3. 14 Kuesioner	53
3. 15 Hasil Nilai Persentase Tiap Pertanyaan Kuesioner	55

DAFTAR GAMBAR

Gambar	Halaman
1. 1 Tahapan Model Waterfall	8
3. 1 Use Case Diagram Sistem Pendaftaran Anggota	13
3. 2 Class diagram Sistem Pendaftaran Anggota.....	19
3. 3 Activity Diagram Login Admin Sistem Website	20
3. 4 Activity Diagram Detail Data Sistem Website	20
3. 5 Activity Diagram Hapus Data Sistem Website	21
3. 6 Activity Diagram Cari Data Sistem Website	21
3. 7 Activity Diagram Tambah Data Buku	22
3. 8 Activity Diagram Logout Sistem Website.....	23
3. 9 Activity Diagram Login Sistem Android	23
3. 10 Activity Diagram Register Sistem Android.....	24
3. 11 Activity Diagram Input Data Sistem Android	25
3. 12 Activity Diagram Profil Sistem Android	26
3. 13 Activity Diagram Buku Sistem Android	26
3. 14 Activity Diagram Search Buku Sistem Android	27
3. 15 Activity Diagram Logout Sistem Android	27
3. 16 Rancangan struktur menu admin.....	28
3. 17 Rancangan struktur menu CalonAnggota	29
3. 18 Wireframe Login Admin	29
3. 19 Wireframe Dashboard Admin.....	30
3. 20 Wireframe Data Anggota Perpustakaan	30
3. 21 Wireframe Data Calon Anggota	31
3. 22 Wireframe Data Buku.....	31
3. 23 Wireframe Tambah Buku	32
3. 24 Wireframe Detail Data	33
3. 25 Wireframe Hapus Data	33
3. 26 Wireframe Logout	34
3. 27 Wireframe Halaman Splash screen	34
3. 28 Wireframe Halaman Login dan Register.....	35
3. 29 Wireframe Halaman Home.....	35
3. 30 Wireframe Halaman Form dan Konfirmasi Data	36
3. 31 Wireframe Halaman Buku dan kategori.....	37
3. 32 Wireframe Halaman Detail Buku.....	37
3. 33 Wireframe Halaman Profil dan Logout.....	38
3. 34 Tampilan Halaman Login website.....	39
3. 35 Tampilan Halaman Dashboard website.....	39
3. 36 Tampilan Halaman Data Anggota Perpustakaan website.....	40
3. 37 Tampilan Detail Data Anggota Perpustakaan website.....	40
3. 38 Tampilan Hapus Data website	41
3. 39 Tampilan Halaman Data Calon Anggota Website	41
3. 40 Tampilan Halaman Data Buku	42

3. 41	Tampilan Input buku	42
3. 42	Tampilan Splash Screen	43
3. 43	Tampilan Halaman Login	44
3. 44	Tampilan Kesalahan login	44
3. 45	Tampilan Halaman Register	45
3. 46	Tampilan Halaman Home.....	46
3. 47	Tampilan Halaman Formulir	46
3. 48	Tampilan Kesalahan Halaman Formulir.....	47
3. 49	Tampilan Halaman Confirm Data	48
3. 50	Tampilan Kesalahan Halaman Confirm Data.....	48
3. 51	Tampilan Halaman Buku dan Halaman Kategori.....	49
3. 52	Tampilan Search Buku dan Halaman Detail Buku.....	50
3. 53	Tampilan Halaman Profil	50
3. 54	Grafik Hasil Kuesioner.....	55

DAFTAR LAMPIRAN

Lampiran	Halaman
L1 Surat Izin Penelitian.....	61
L2 Surat Balasan Izin Penelitian	62
L3 Kartu Kendali Bimbingan.....	63
L4 Jadwal Penelitian	64
L5 Code main.dart.....	65
L6 splashscreens.dart	65
L7 Code login.dart.....	67
L8 Code Register.dart.....	69
L9 globals.dart.....	70
L10 code form.dart.....	70
L11 profil.dart	73
L12 screenbuku.dart	74
L13 code Login.dart	76
L14 code dashboard.dart	77
L15 code anggota perpustakaan	78
L16 code calon anggota.....	80
L17 code detail data	82
L18 code routes/api.php	83
L19 routes/web.php.....	83
L20 code app/http/controllers/logincontroller.php	84
L21 code app/http/controllers/daftaranggotacontroller.php	85
L22 code app/http/controllers/anggotaapicontroller.php.....	86
L23 wawancara	87

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Penggunaan teknologi telah mengalami peningkatan yang signifikan, menjadi bagian tak terpisahkan dari berbagai aspek kehidupan. Teknologi yang semakin canggih memberikan kontribusi dalam berbagai aktivitas sehari-hari. Salah satu contohnya adalah penggunaan *smartphone* (Bimantoro et al., 2021). Saat ini, penggunaan *smartphone* telah menjadi kebiasaan umum di masyarakat, dengan berbagai aplikasi yang menyediakan layanan dan informasi yang dapat diakses dengan cepat dan mudah di mana pun (Ridwansyah, 2023).

Menurut artikel dari databoks.katada.co.id oleh Yosepha Pusparisa, penggunaan ponsel pintar di Indonesia diperkirakan akan terus meningkat. Hanya 28,6% masyarakat Indonesia yang memiliki ponsel pintar pada tahun 2015. Namun, seiring berjalannya waktu, ponsel menjadi lebih terjangkau sehingga penggunaannya semakin luas. Pada tahun 2018, sebagian besar penduduk Indonesia atau sekitar 56,2% menggunakan telepon seluler. Pada tahun berikutnya, persentase ini meningkat menjadi 63,3%. Pada tahun 2025, setidaknya 89,2 persen penduduk Indonesia diperkirakan akan menggunakan ponsel pintar, dengan penetrasi ponsel cerdas meningkat sebesar 25,9 persen dalam enam tahun sejak tahun 2019 (Yosepha Pusparisa, 2020).

Dinas Perpustakaan dan Kearsipan Kota Samarinda telah menggunakan teknologi informasi berbasis website untuk pendaftaran anggota baru dan pengelolaan data buku. Namun, akses ke website ini terbatas hanya untuk petugas atau admin perpustakaan, sehingga semua proses pendaftaran anggota harus dilakukan di perpustakaan. Admin akan menginput data pendaftaran melalui website tersebut. Database yang digunakan untuk menyimpan data anggota perpustakaan adalah MySQL.

Meskipun telah mengadopsi sistem berbasis website untuk admin, pada praktiknya Dinas Perpustakaan dan Kearsipan Kota Samarinda masih menerapkan sistem pendaftaran anggota baru dan pencarian buku secara langsung di perpustakaan untuk pengunjung. Pendaftaran anggota baru saat ini masih dilakukan secara manual dengan mengisi formulir yang disediakan oleh perpustakaan. Untuk mendaftar sebagai anggota baru, calon anggota harus membawa persyaratan yang telah ditetapkan, seperti fotocopy kartu tanda penduduk. Persyaratan ini terkadang menyebabkan calon anggota harus mengunjungi perpustakaan beberapa kali karena kelengkapan dokumen yang belum terpenuhi. Situasi ini menimbulkan kebutuhan akan sebuah aplikasi yang memudahkan calon anggota dalam proses pendaftaran serta menghemat waktu. Selain itu, anggota perpustakaan harus datang langsung untuk mencari buku yang diinginkan, namun seringkali buku yang dicari tidak tersedia sehingga mereka pulang dengan tangan kosong. Hal ini menunjukkan perlunya sebuah sistem yang dapat mempermudah proses pencarian buku dan meningkatkan efisiensi layanan perpustakaan.

Untuk mengatasi permasalahan tersebut, penelitian ini akan mengembangkan aplikasi pendaftaran anggota perpustakaan berbasis Android menggunakan framework Flutter untuk *frontend* dan Laravel untuk *backend*, dengan database MySQL sebagai basis data. Flutter adalah framework open-source buatan Google untuk pengembangan aplikasi mobile menggunakan bahasa Dart. Flutter memungkinkan pembuatan aplikasi yang kompatibel dengan Android dan iOS (Chandra et al., 2019). Laravel adalah kerangka kerja PHP yang populer untuk pengembangan situs web. Dengan fitur-fitur terintegrasi, Laravel mengurangi kompleksitas penulisan kode PHP, meningkatkan efisiensi pengelolaan kode, dan mempercepat pembuatan situs web (Hendrawan et al., 2020). MySQL adalah RDBMS gratis berlisensi GPL yang populer karena kinerja query data yang unggul. Penggunaannya yang bebas biaya membuatnya pilihan terjangkau, meskipun harus mematuhi ketentuan lisensi GPL yang melarang konversi ke produk tertutup atau komersial (Zulfa & Wanda, 2023).

Dalam rangka pengembangan, aplikasi ini akan mengambil pendekatan waterfall, sebuah metodologi terstruktur yang sering digunakan dalam industri teknologi informasi. Pendekatan ini membagi pengembangan perangkat lunak menjadi serangkaian tahapan yang dilakukan secara berurutan. Keistimewaan dari Model Waterfall adalah kejelasan langkah-langkahnya dan ketidakmungkinan untuk mundur ke tahap sebelumnya setelah tahap tertentu telah selesai (Anis et al., 2023).

Metode waterfall melibatkan lima tahap utama. Tahap pertama, Analisis Kebutuhan (*Requirements Analysis*), melibatkan konsultasi dengan pengguna untuk menentukan kebutuhan sistem, batasan, dan tujuan. Informasi ini kemudian dijabarkan secara rinci dan berfungsi sebagai spesifikasi sistem (Siregar & Pristiwanto, 2022), Tahap kedua, Pengembangan sistem dan perangkat lunak melibatkan perencanaan aplikasi, termasuk pengaturan struktur data, penentuan arsitektur, perancangan antarmuka, dan prosedur pengkodean. Kebutuhan yang telah dianalisis disusun menjadi rencana yang dapat dijalankan. Selama proses ini, hasil desain didokumentasikan untuk referensi di masa depan (Kahfi et al., 2023), Ketiga Implementation and unit testing adalah tahapan yang secara nyata dilalui dalam pengembangan suatu sistem. Tahapan ini mengoptimalkan penggunaan komputer. Setelah proses pengkodean selesai, sistem akan diuji untuk menemukan kesalahan dan kemudian diperbaiki (Azrial & Fadillah, 2020), Tahap keempat, *Integration and system testing*, bertujuan memverifikasi kecocokan sistem dengan tujuan yang ditetapkan melalui black box testing. Metode ini menguji aplikasi secara fungsional tanpa memperhatikan detail internal. Pengujian ini mengidentifikasi kesalahan, fungsi yang tidak sesuai, atau fitur yang terlewat, serta mengukur tingkat kepuasan pengguna (Nugraha et al., 2020), Tahap kelima, Operation and maintenance, Mengikutsertakan pengguna dalam uji langsung aplikasi, dilanjutkan dengan evaluasi potensi kelemahan. Jika ditemukan kelemahan, langkah pemeliharaan akan dijalankan untuk memperbaikinya (Handayani & Salam, 2023).

Beberapa penelitian sebelumnya telah mengimplementasikan model Waterfall dan framework Flutter dalam pengembangan sistem, salah satunya adalah sebuah penelitian yang dilakukan oleh Nur Fadilah, Ali Ikhwan dan Muhamad Alda dengan judul “Pengembangan Sistem Informasi Perpustakaan Pada Dinas Perpustakaan dan Kearsipan Kota Medan Berbasis Android”, penelitian ini sukses mengaplikasikan Flutter sebagai alat utama untuk mempermudah proses pendaftaran, peminjaman, dan perpanjangan buku. Menggunakan metode Research and Development dengan pendekatan waterfall, aplikasi ini terintegrasi dengan Firebase untuk efisiensi pengelolaan data (Fadilah et al., 2023). Konsep-konsep ini dapat menjadi inspirasi dalam pengembangan aplikasi pendaftaran anggota perpustakaan berbasis Android untuk meningkatkan pengalaman pengguna.

Penelitian selanjutnya dilakukan oleh Nana Supiana dalam artikelnya yang berjudul "pengembangan aplikasi geolocation untuk monitoring lokasi mahasiswa selama pandemi berbasis android menggunakan metode waterfall" memiliki relevansi yang signifikan dalam pengembangan aplikasi pendaftaran anggota perpustakaan berbasis Android. Meskipun fokusnya berbeda, namun konsep penggunaan metode waterfall dalam pengembangan perangkat lunak, penggunaan platform Android dapat memberikan inspirasi dan panduan yang berharga dalam pengembangan aplikasi pendaftaran anggota perpustakaan (Supiana, 2022).

Dengan adanya permasalahan dalam proses pendaftaran anggota perpustakaan yang masih menggunakan formulir manual, khususnya di Dinas Perpustakaan dan Kearsipan Kota Samarinda, serta melihat tren penggunaan teknologi yang semakin meningkat di Indonesia, terutama dalam penggunaan smartphone, maka pengembangan sebuah aplikasi pendaftaran anggota perpustakaan berbasis Android menjadi langkah yang strategis. Peneliti ini akan mengusulkan solusi berupa pengembangan aplikasi tersebut menggunakan framework Flutter untuk frontend, Laravel untuk backend, dan MySQL sebagai basis data. Dengan demikian, diharapkan dapat meningkatkan efektivitas dan efisiensi layanan perpustakaan serta mengikuti

arus digitalisasi yang semakin berkembang. Melalui penggabungan kontribusi dari penelitian-penelitian terdahulu yang telah sukses mengimplementasikan metode waterfall dalam pengembangan aplikasi mobile, peneliti akan menjadikan pendekatan ini sebagai landasan utama dalam pengembangan aplikasi pendaftaran anggota perpustakaan berbasis Android.

1.2 Batasan Masalah

Batasan masalah dalam penelitian ini, (i) Pengembangan aplikasi ini terbatas pada pendaftaran anggota perpustakaan dan mencari buku. (ii) Aplikasi akan dikembangkan khusus untuk platform android menggunakan framework flutter dan website menggunakan laravel untuk backend. Aplikasi ini tidak mencakup pengembangan untuk platform iOS atau lainnya, (iii) aplikasi akan memiliki fitur-fitur dasar seperti pendaftaran anggota baru, verifikasi data anggota, mencari buku dan penyimpanan data anggota dalam basis data MySQL.

1.3 Rumusan Masalah

Permasalahan yang dapat dirumuskan adalah (i) Bagaimana mengembangkan aplikasi pendaftaran anggota perpustakaan berbasis Android menggunakan model waterfall yang dapat mempermudah proses pendaftaran anggota perpustakaan di Kota Samarinda?. (ii) Bagaimana merancang dan mengimplementasikan fitur pencarian buku dalam aplikasi tersebut untuk memudahkan anggota perpustakaan dalam mencari buku yang diinginkan?

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk (i) mengembangkan aplikasi pendaftaran anggota perpustakaan berbasis Android menggunakan model waterfall yang dapat membantu mempermudah pendaftaran anggota perpustakaan di Kota Samarinda. (ii) Mempermudah proses pencarian buku bagi anggota perpustakaan melalui fitur pencarian dalam aplikasi.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut (i) Bagi Peneliti, Penelitian ini memberikan pengalaman praktis serta pengetahuan mendalam dalam

pengembangan sistem Pendaftaran Anggota Perpustakaan berbasis Android. Selain itu, penelitian ini juga memperkuat keterampilan teknis dalam menggunakan framework Flutter, Dart, Laravel, PHP, dan MySQL. (ii) Bagi peneliti lain atau mahasiswa, Hasil penelitian ini akan menjadi sumber inspirasi dan referensi yang berharga bagi peneliti lain atau mahasiswa yang tertarik mengembangkan aplikasi berbasis Android. Dengan memanfaatkan teknologi terkini seperti framework Flutter dan model pengembangan perangkat lunak seperti model Waterfall, peneliti lain atau mahasiswa dapat memperoleh panduan yang berguna dalam mengatasi tantangan pengembangan perangkat lunak modern. (iii) Bagi Universitas Muhammadiyah Kalimantan Timur, Penelitian ini diharapkan dapat memberikan kontribusi penambahan ilmu pengetahuan, khususnya di bidang Informatika serta menjadi bahan bacaan di perpustakaan Universitas Muhammadiyah Kalimantan Timur dan dapat memberikan referensi bagi mahasiswa lain. (iv) Bagi Perpustakaan Kota Samarinda, Hasil penelitian ini diharapkan dapat membantu meningkatkan efisiensi proses pendaftaran anggota perpustakaan di Kota Samarinda.

BAB II

METODE PENELITIAN

2.1 Objek Penelitian

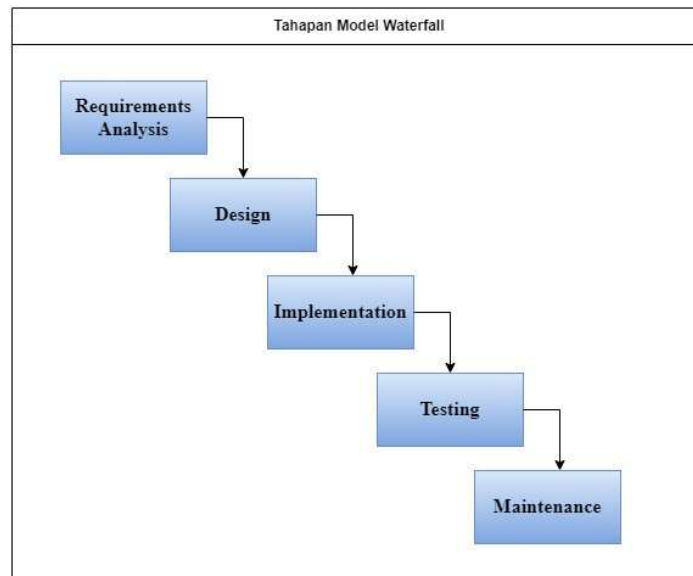
Dalam penelitian ini yang menjadi objek penelitian adalah Dinas Perpustakaan dan Kearsipan Kota Samarinda, yang berfungsi sebagai lembaga penyedia layanan informasi dan literasi bagi masyarakat. permasalahan yang dihadapi ialah proses pendaftaran anggota perpustakaan yang masih manual, yang dapat menyebabkan kesalahan data. Penelitian ini berfokus pada pengembangan aplikasi pendaftaran anggota berbasis Android untuk meningkatkan akurasi dan efisiensi layanan perpustakaan. Aplikasi ini diharapkan dapat memberikan manfaat bagi pengguna perpustakaan.

2.2 Alat dan Bahan

Dalam penelitian ini, digunakan alat penelitian dalam bentuk perangkat keras dan perangkat lunak sebagai sarana untuk mengembangkan aplikasi tersebut. (i) Perangkat keras, INTEL CORE i5 12400F, RAM 16GB, Vga RTX 2060 Super, Nvme 500GB, Mouse, Keyboard dan Pocophone f1. (ii) Perangkat Lunak, Windows 10 pro, Microsoft Office 2019, Visual Studio Code, Xampp, Balsamiq Wireframes.

2.3 Metode Pengembangan Sistem

Pada penelitian ini, metode pengembangan sistem menggunakan metode Waterfall, yang termasuk dalam kerangka kerja *System Development Life Cycles* (SDLC). Dikenal karena pendekatannya yang terstruktur dan berkesinambungan dalam pengembangan perangkat lunak, model Waterfall menuntut penyelesaian setiap tahap sebelum memasuki tahap berikutnya, tanpa opsi untuk mundur ke tahap sebelumnya. Gambar 1.1 dibawah ini merupakan tahapan-tahapan yang terdapat dalam model waterfall.



Gambar 1. 1 Tahapan Model Waterfall

Tahapan-tahapan pendekatan model waterfall adalah sebagai berikut.

2.3.1 Analisis Kebutuhan (*Requirements Analysis*)

Pada tahap ini, dilakukan analisis kebutuhan pengembangan aplikasi pendaftaran anggota perpustakaan berbasis Android. Kegiatan meliputi observasi proses pendaftaran di Dinas Perpustakaan Kota Samarinda untuk mengidentifikasi masalah, wawancara dengan petugas perpustakaan untuk memahami kebutuhan dan fitur yang diharapkan, serta studi literatur terkait pengembangan aplikasi Android menggunakan Flutter, Laravel, dan MySQL. Hasil dari kegiatan ini digunakan untuk mendefinisikan fitur dan fungsi aplikasi.

2.3.2 Perancangan (*Design*)

Tahap perancangan sistem melibatkan beberapa aspek. Pertama, struktur data dirancang menggunakan MySQL untuk menyimpan informasi anggota perpustakaan dan data terkait. Kedua, pemodelan sistem dilakukan dengan UML (*Unified Modelling Language*), mencakup Use Case, Class Diagram, dan Activity Diagram. Arsitektur perangkat lunak mengadopsi Flutter untuk frontend dan Laravel untuk backend. Antarmuka pengguna didesain dengan

software Balsamiq Wireframes, menekankan kemudahan penggunaan dan kejelasan informasi. Terakhir, algoritma dan alur kerja aplikasi dirancang untuk mengoptimalkan efisiensi proses pendaftaran anggota perpustakaan.

2.3.3 Implementasi (*Implementation*)

Tahap implementasi merupakan fase penting di mana rancangan sistem diterjemahkan menjadi kode program fungsional. Dalam proses ini, peneliti memanfaatkan bahasa pemrograman Dart dengan Flutter sebagai framework frontend, sementara Laravel digunakan sebagai backend dan untuk pengembangan API. Kombinasi framework ini memfasilitasi pengembangan dan integrasi fitur-fitur yang telah direncanakan. Untuk meningkatkan efisiensi dan organisasi pengkodean, Visual Studio Code dipilih sebagai editor teks utama.

2.3.4 Pengujian (*Testing*)

Pada tahap ini, dilakukan pengujian terhadap aplikasi yang telah diimplementasikan. Kegiatan yang dilakukan meliputi black box testing untuk memverifikasi semua fitur aplikasi berfungsi dengan baik. Black box testing ini dilakukan tanpa memperhatikan detail kode sumber, dengan tujuan memastikan aplikasi dapat menerima input dengan benar, memproses data secara tepat, dan menghasilkan output sesuai kebutuhan pengguna. Pengujian ini krusial untuk mendeteksi dan memperbaiki kesalahan atau bug sebelum aplikasi diimplementasikan secara penuh.

2.3.5 Pemeliharaan (*Maintenance*)

Pada tahap ini, dilakukan pemeliharaan terhadap aplikasi setelah implementasi. Ini mencakup perbaikan bug atau kesalahan yang mungkin muncul selama penggunaan aplikasi. Bug tersebut dapat berupa kesalahan perhitungan, tampilan yang tidak sesuai, atau masalah lainnya. Selain itu, dilakukan peningkatan kinerja aplikasi seperti optimalisasi basis data, perbaikan algoritma, dan peningkatan performa secara keseluruhan.

2.4 Pengumpulan Data

Proses pengumpulan data dilakukan untuk mendapatkan informasi relevan guna mencapai tujuan penelitian. Pendekatan yang digunakan meliputi: (i) Observasi, mengamati langsung proses pendaftaran anggota perpustakaan di Dinas Perpustakaan Kota Samarinda untuk mengidentifikasi masalah. (ii) Wawancara, berbicara dengan petugas perpustakaan untuk memahami kebutuhan pengembangan aplikasi pendaftaran anggota berbasis Android. (iii) Studi pustaka, meneliti penelitian terdahulu seperti jurnal, skripsi, dan literatur terkait masalah penelitian ini.

BAB III

HASIL DAN PEMBAHASAN

3.1 Analisis Kebutuhan

Analisis kebutuhan aplikasi pendaftaran anggota perpustakaan berbasis Android dilakukan melalui studi literatur, observasi, dan wawancara dengan petugas pendaftaran perpustakaan. Observasi mempelajari alur pendaftaran saat ini, sedangkan studi literatur mengacu pada berbagai jurnal terkait. Berikut adalah hasil analisis berdasarkan metode tersebut:

3.1.1 Hasil Studi Literatur

Studi Literatur yang dilakukan berfokus pada pengembangan sistem pendaftaran anggota perpustakaan berbasis android dengan metode waterfall, framework flutter dengan bahasa pemrograman dart dan framework laravel dengan bahasa pemrograman PHP. Berikut adalah temuan utama dari studi literatur ini:

- a. **MySQL:** MySQL adalah DBMS terbuka, gratis, dan mendukung banyak pengguna. SQL adalah bahasa untuk berinteraksi dengan database, memungkinkan pembuatan dan manipulasi data, sesuai prinsip RDBMS.(Rina Noviana, 2022).
- b. **Metode Waterfall:** Metode ini membuat pengembangan perangkat lunak lebih terorganisir dan sesuai untuk proyek dengan persyaratan jelas. Setiap fase harus selesai sebelum pindah ke fase berikutnya, memastikan dokumentasi lengkap dan manajemen proyek rapi (Badrul, 2021).
- c. **Bahasa Dart:** Bahasa pemrograman dari Google ini dirancang untuk pengembangan aplikasi lintas platform, termasuk mobile, web, dan desktop, sehingga sangat cocok untuk membuat aplikasi pendaftaran anggota perpustakaan berbasis Android (Hanif & Sinambela, 2020).
- d. **Framework Flutter :** framework yang dikembangkan oleh Google untuk pengembangan aplikasi lintas platform, termasuk mobile, web, dan desktop. Dengan Flutter, pengembang

dapat membuat antarmuka pengguna yang indah dan responsif menggunakan satu basis kode (Santoso et al., 2020).

- e. **Bahasa PHP (*Hypertext Preprocessor*):** Bahasa pemrograman ini sering digunakan untuk pengembangan situs web dinamis, memungkinkan pengembang membuat situs interaktif yang berkomunikasi dengan basis data dan menghasilkan konten secara dinamis (Mahdalena et al., 2023).
- f. **Framework Laravel:** Laravel adalah framework PHP yang terkenal karena kecepatan pengembangan, keamanan yang solid, dan dukungan komunitas yang besar, menjadikannya pilihan utama untuk pengembangan aplikasi web modern (Alfarisi et al., 2023).

3.1.2 Hasil Observasi

Proses pendaftaran anggota baru di Dinas Perpustakaan dan Kearsipan Kota Samarinda terdiri dari lima tahap: (i) Calon anggota datang ke perpustakaan, (ii) Mengisi formulir pendaftaran manual dengan data pribadi, (iii) Petugas memverifikasi data yang diisi, (iv) Pembuatan kartu anggota, meliputi pencetakan dan input data ke sistem, (v) Penyerahan kartu kepada calon anggota yang kini resmi menjadi anggota perpustakaan.

3.1.3 Hasil Wawancara

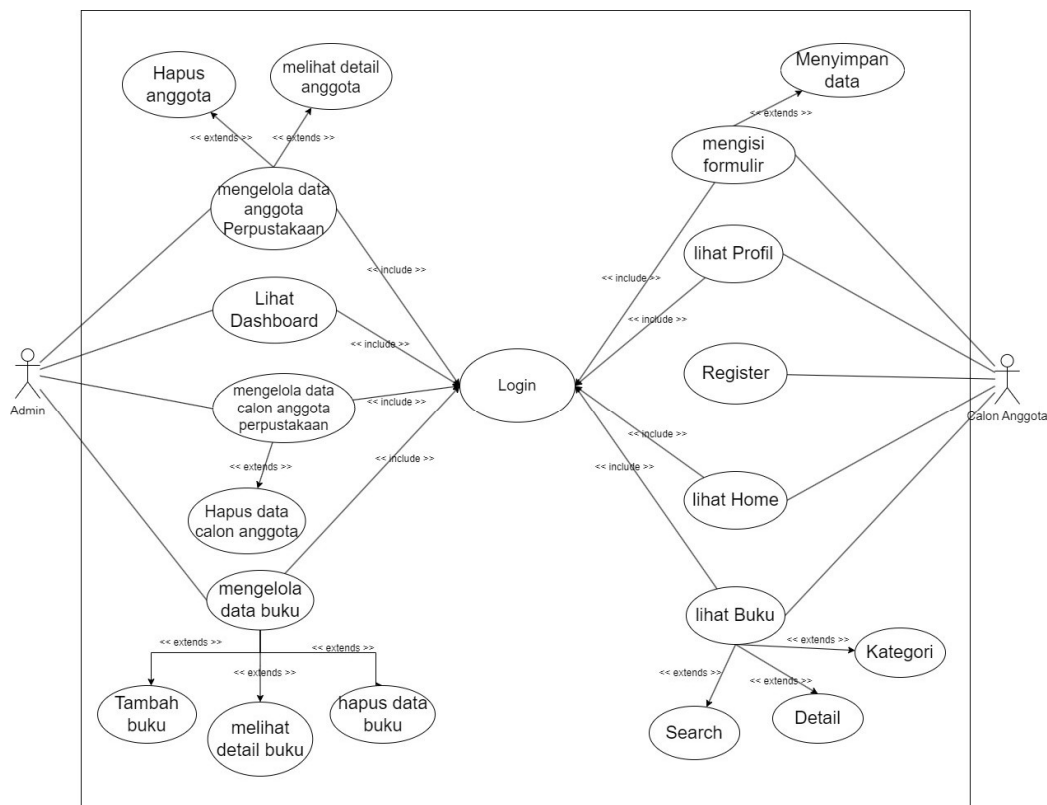
Penulis mewawancarai Putri Desintia Ramadhani, Staff Layanan Perpustakaan berusia 24 tahun dari Samarinda, untuk mendapatkan informasi mendalam tentang proses pendaftaran anggota di Dinas Perpustakaan Kota Samarinda. Fitur yang diharapkan dalam sistem baru meliputi: login akun, registrasi anggota baru, input dan penyimpanan data, penghapusan data, pencarian berdasarkan kriteria, serta kemampuan melihat detail data anggota secara lengkap.

3.2 Perancangan (Desain)

3.4.1 Perancangan Sistem

Pada tahap ini dilakukan perancangan sistem berdasarkan hasil analisis yang diperoleh dari tahap sebelumnya. Pada tahap ini, rancangan sistem dikembangkan dalam bentuk aplikasi android dan website. Rancangan sistem mencakup struktur menu, desain basis data, dan berbagai fitur. Untuk pemodelan data, digunakan UML (*Unified Modelling Language*) berupa *Use Case Diagram*, *Class Diagram* dan *activity Diagram*.

a. Use Case Diagram



Gambar 3. 1 Use Case Diagram Sistem Pendaftaran Anggota

Untuk menjelaskan use case diagram pada gambar 3.1 maka dibuatlah use case tabel. Pada tabel 3.1 merupakan *use case* tabel yang menjelaskan proses login Admin.

Tabel 3. 1 Use Case Tabel Login Admin

Nama Use Case	Login Admin
Aktor	Administrator
Deskripsi	Administrator masuk ke dalam sistem
Prasyarat	Administrator harus memiliki akun yang telah terdaftar di dalam sistem
Alur Utama	1. Administrator membuka halaman login. 2. Administrator memasukan username dan password. 3. Sistem memverifikasi. 4. Administrator berhasil login dan diarahkan ke dashboard
Alur Alternatif	1. Administrator membuka halaman login. 2. Administrator memasukan username dan password. 3. Sistem memverifikasi dan gagal. 4. Sistem menampilkan pesan kesalahan
Hasil Akhir	Admin berhasil login dan sistem menampilkan halaman dashboard

Pada tabel 3.2 merupakan *use case* tabel untuk *use case* yang menjelaskan proses register calon anggota.

Tabel 3. 2 Use Case Tabel Register Calon Anggota

Nama Use Case	Register Calon Anggota
Aktor	Calon Anggota
Deskripsi	Calon Anggota mendaftar ke dalam sistem
Prasyarat	-
Alur Utama	1. Calon anggota membuka halaman register. 2. Calon anggota mengisi data pendaftaran. 3. Sistem memverifikasi. 4. Calon anggota berhasil register dan diarahkan ke halaman login
Alur Alternatif	1. Calon anggota membuka halaman register. 2. Calon anggota mengisi data pendaftaran. 3. Sistem memverifikasi dan gagal. 4. Sistem menampilkan pesan kesalahan
Hasil Akhir	1. Calon anggota membuka halaman register. 2. Calon anggota mengisi data pendaftaran. 3. Sistem memverifikasi dan gagal. 4. Sistem menampilkan pesan kesalahan

Pada tabel 3.3 merupakan *use case* tabel untuk *use case* yang menjelaskan proses lihat dashboard.

Tabel 3.3 Use Case Tabel Melihat Dashboard

Nama Use Case	Lihat Dashboard
Aktor	Administrator
Deskripsi	Administrator melihat halaman dashboard
Prasyarat	Administrator telah login ke dalam sistem
Alur Utama	1. Administrator berhasil login ke dalam sistem. 2. Administrator memilih opsi dashboard. 3. Sistem akan membuka halaman dashboard
Alur Alternatif	-
Hasil Akhir	Calon anggota berhasil register dan login, kemudian sistem menampilkan halaman home

Pada tabel 3.4 merupakan *use case* tabel untuk *use case* yang menjelaskan proses mengelola data anggota perpustakaan.

Tabel 3.4 Use Case Tabel Mengelola Data Anggota Perpustakaan

Nama Use Case	Mengelola Data Anggota perpustakaan
Aktor	Admin
Deskripsi	Admin dapat melihat detail data anggota dan menghapus data anggota perpustakaan
Prasyarat	Admin telah login ke dalam sistem
Alur Utama	1. Admin membuka halaman kelola data anggota perpustakaan. 2. Admin melihat detail data atau menghapus data anggota perpustakaan. 3. Sistem menyimpan perubahan
Alur Alternatif	1. Admin membuka halaman kelola data anggota perpustakaan. 2. Admin melihat detail data atau menghapus data anggota perpustakaan. 3. Sistem menemukan kesalahan, misalnya data tidak valid. 4. Sistem menampilkan pesan kesalahan.
Hasil Akhir	Admin mengelola data anggota perpustakaan berhasil.

Pada tabel 3.5 merupakan *use case* tabel untuk *use case* yang menjelaskan proses mengelola data anggota perpustakaan.

Tabel 3.5 Use Case Tabel Mengelola Data Calon Anggota Perpustakaan

Nama Use Case	Mengelola Data Calon Anggota
Aktor	Admin
Deskripsi	Admin dapat menghapus data calon anggota perpustakaan
Prasyarat	Admin telah login ke dalam sistem
Alur Utama	1. Admin membuka halaman kelola data calon anggota perpustakaan. 2. Admin menghapus data anggota perpustakaan. 3. Sistem menyimpan perubahan
Alur Alternatif	1. Admin membuka halaman kelola data calon anggota perpustakaan. 2. Admin menghapus data anggota perpustakaan. 3. Sistem menemukan kesalahan, misalnya data tidak dapat di hapus. 4. Sistem menampilkan pesan kesalahan.
Hasil Akhir	Admin mengelola data calon anggota perpustakaan berhasil.

Pada tabel 3.6 merupakan *use case* tabel untuk *use case* yang menjelaskan proses mengelola data buku.

Tabel 3. 6 Use Case Tabel Mengelola Data Buku

Nama Use Case	Mengelola Data Buku
Aktor	Admin
Deskripsi	Admin dapat menambahkan buku, melihat detail buku, menghapus data buku
Prasyarat	Admin telah login ke dalam sistem
Alur Utama	1. Admin membuka halaman kelola data buku. 2. Admin menambahkan data buku, melihat detail buku dan menghapus data buku. 3. Sistem menyimpan perubahan
Alur Alternatif	1. Admin membuka halaman kelola data buku. 2. Admin menambahkan data buku, melihat detail buku dan menghapus data buku. 3. Sistem menemukan kesalahan, misalnya data buku tidak dapat di hapus. 4. Sistem menampilkan pesan kesalahan.
Hasil Akhir	Admin mengelola data buku berhasil.

Pada tabel 3.7 merupakan *use case* tabel untuk *use case* yang menjelaskan proses Mengisi Formulir.

Tabel 3. 7 Use Case Tabel Mengisi Formulir

Nama Use Case	Mengisi Formulir
Aktor	Calon Anggota
Deskripsi	Calon Anggota mengisi formulir untuk mendaftar sebagai anggota perpustakaan
Prasyarat	Calon Anggota telah login ke dalam sistem
Alur Utama	1. Calon anggota berhasil login kedalam sistem 2. Calon anggota membuka halaman formulir. 3. Halaman formulir ini berisi data pribadi yang harus diisi oleh calon anggota. 4. sistem memverifikasi. 5. Calon anggota berhasil mengisi formulir
Alur Alternatif	1. Calon anggota berhasil login ke dalam sistem. 2. Calon anggota membuka halaman formulir. 3. Halaman formulir ini berisi data pribadi yang harus diisi oleh calon anggota. 4. sistem memverifikasi dan gagal. 5. sistem akan menampilkan kesalahan, misalnya ada form yang belum ke isi.
Hasil Akhir	Calon anggota berhasil mengisi formulir pendaftaran anggota perpustakaan

Pada tabel 3.8 merupakan *use case* tabel untuk *use case* yang menjelaskan proses Melihat Halaman Profil.

Tabel 3. 8 *Use Case* Tabel Melihat Halaman Profil

Nama Use Case	Lihat Profil
Aktor	Calon Anggota
Deskripsi	Calon anggota melihat halaman profil yang berisi data yang telah di input pada halaman formulir
Prasyarat	Calon Anggota telah login ke dalam sistem dan calon anggota sudah menginput formulir
Alur Utama	1. Calon anggota berhasil login ke dalam sistem 2. Calon anggota memilih halaman profil. 2. sistem akan menampilkan halaman profil
Alur Alternatif	-
Hasil Akhir	Calon anggota berhasil melihat halaman profil

Pada tabel 3.9 merupakan *use case* tabel untuk *use case* yang menjelaskan proses Melihat halaman home.

Tabel 3. 9 *Use Case* Tabel Melihat Halaman Home

Nama Use Case	Lihat Home
Aktor	Calon Anggota
Deskripsi	Calon anggota melihat halaman home
Prasyarat	Calon Anggota telah login ke dalam sistem
Alur Utama	1. Calon anggota berhasil login ke dalam sistem 2. Calon anggota memilih halaman Home. 2. sistem akan menampilkan halaman home
Alur Alternatif	-
Hasil Akhir	Calon anggota berhasil melihat halaman home

Pada tabel 3.10 merupakan *use case* tabel untuk *use case* yang menjelaskan proses menyimpan data.

Tabel 3. 10 Use Case Tabel Menyimpan Data

Nama Use Case	Menyimpan Data
Aktor	Calon Anggota
Deskripsi	Calon anggota menyimpan data pendaftaran ke sistem
Prasyarat	Calon Anggota telah login ke dalam sistem
Alur Utama	1. Calon anggota berhasil login ke dalam sistem 2. Calon anggota membuka halaman formulir. 3. Halaman formulir ini berisi data pribadi yang harus diisi oleh calon anggota. 4. Calon anggota mengklik submit. 5. sistem akan menampilkan halaman confirm data yang di mana halaman ini untuk memverifikasi lagi kepada calon anggota untuk memastikan bahwa data yang telah diisi sudah benar. 6. Calon anggota mengklik simpan data. 7. sistem memverifikasi. 8. Calon anggota berhasil mendaftar
Alur Alternatif	1. Calon anggota berhasil login ke dalam sistem 2. Calon anggota membuka halaman formulir. 3. Halaman formulir ini berisi data pribadi yang harus diisi oleh calon anggota. 4. Calon anggota mengklik submit. 5. sistem akan menampilkan halaman confirm data yang di mana halaman ini untuk memverifikasi lagi kepada calon anggota untuk memastikan bahwa data yang telah diisi sudah benar. 6. Calon anggota mengklik simpan data. 7. sistem memverifikasi dan gagal. 8. sistem akan menampilkan popup kesalahan, misalnya data yang anda input sudah ada, jadi untuk menyimpan data tidak boleh 2 kali
Hasil Akhir	Calon anggota berhasil mendaftar sebagai anggota perpustakaan

Pada tabel 3.11 merupakan use case tabel untuk use case yang menjelaskan proses buku.

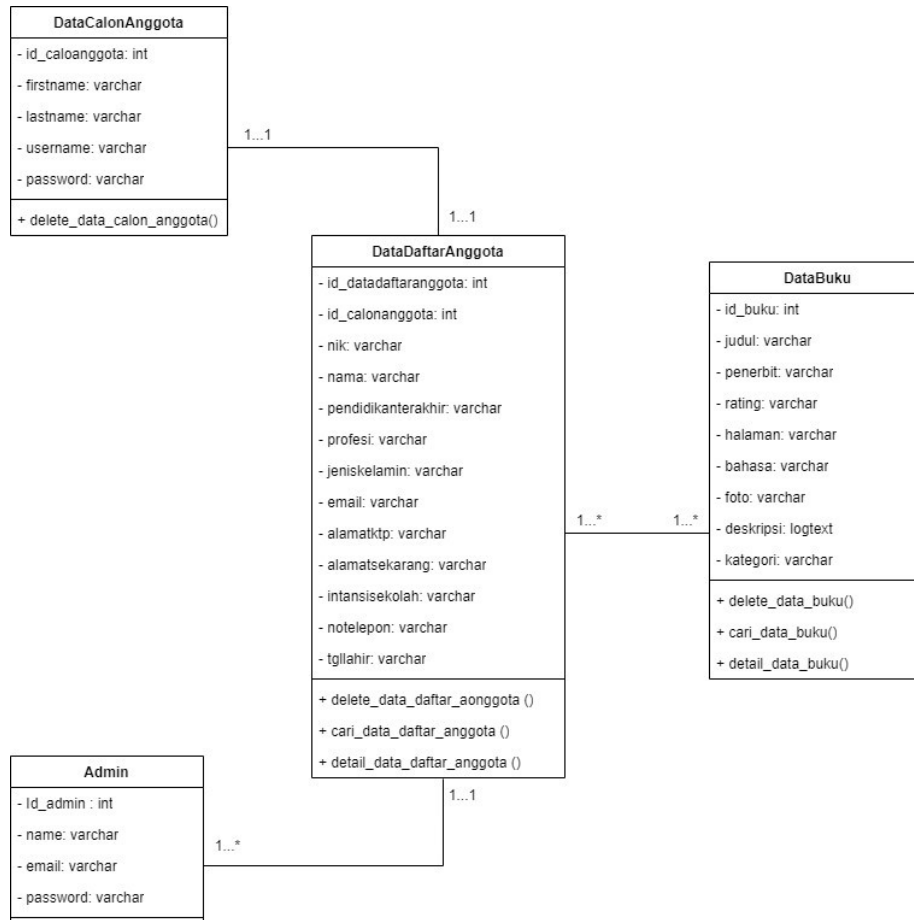
Tabel 3. 11 Use Case Tabel Lihat Buku

Nama Use Case	Lihat Buku
Aktor	Calon Anggota
Deskripsi	Calon anggota dapat mencari buku, melihat detail buku, memilih kategori buku
Prasyarat	Calon Anggota telah login ke dalam sistem
Alur Utama	1. Calon anggota berhasil login ke dalam sistem 2. Calon anggota membuka halaman buku. 3. Calon anggota dapat mencari buku, melihat detail buku, memilih kategori buku, 3. sistem akan menampilkan perubahan
Alur Alternatif	-
Hasil Akhir	Calon anggota berhasil melihat halaman buku

b. Class Diagram

Gambar 3.2 menampilkan class diagram sistem pendaftaran anggota perpustakaan, menunjukkan relasi antar entitas. DataCalonAnggota dan DataDaftarAnggota memiliki relasi one-to-one, artinya setiap calon anggota memiliki satu entri dalam data pendaftaran. DataDaftarAnggota dan DataBuku memiliki relasi one-to-many, di mana satu anggota dapat melihat banyak buku. Admin, meski tidak tergambar langsung, mengelola semua data dengan

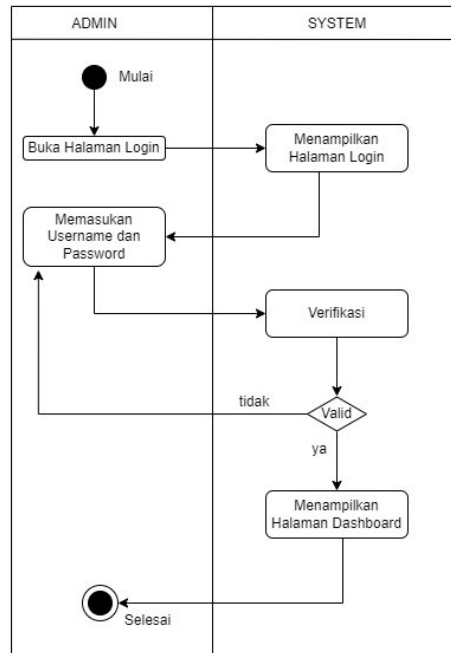
hak akses penuh. Struktur ini memastikan data tersusun baik dan terhubung logis, mendukung fungsionalitas aplikasi dalam mengelola data anggota dan buku perpustakaan.



Gambar 3. 2 *Class diagram* Sistem Pendaftaran Anggota

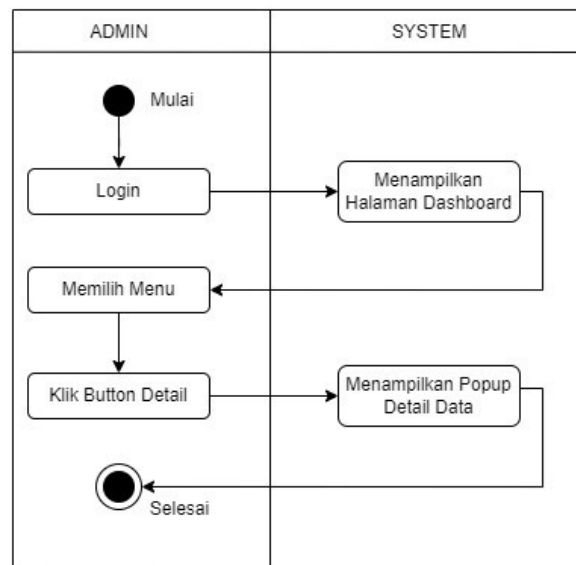
c. Activity Diagram

Activity Diagram menunjukkan aliran kerja atau aktivitas dalam suatu sistem, tetapi tidak mencakup aktivitas aktor. Diagram ini juga menggambarkan bagaimana proses dimulai dan berakhir dalam suatu sistem. Adapun *activity diagram* pada sistem pendaftaran anggota adalah sebagai berikut:



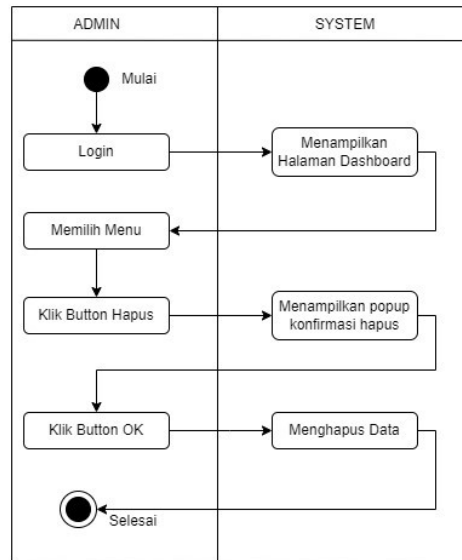
Gambar 3.3 Activity Diagram Login Admin Sistem Website

Pada gambar 3.3 Activity diagram dimulai dengan login, diikuti admin memasukkan dan memvalidasi username serta password. Jika valid, sistem tampilkan dashboard; jika tidak, admin diminta masukkan kembali *username* dan *password*.



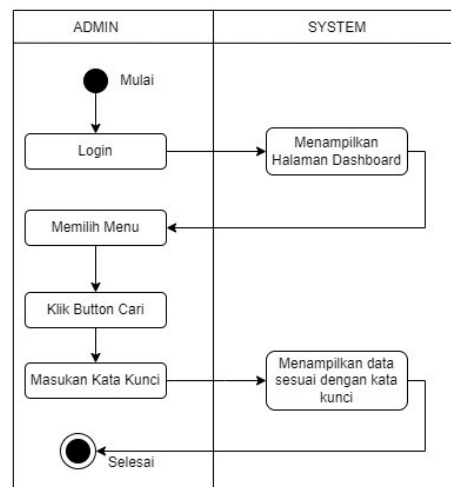
Gambar 3.4 Activity Diagram Detail Data Sistem Website

Pada gambar 3.4 *Activity diagram* tersebut dimulai dengan admin login. Setelah itu, sistem menampilkan halaman dashboard, selanjutnya admin memilih menu yang akan dilihat detail datanya dengan mengklik tombol detail sehingga sistem akan menampilkan popup detail data.



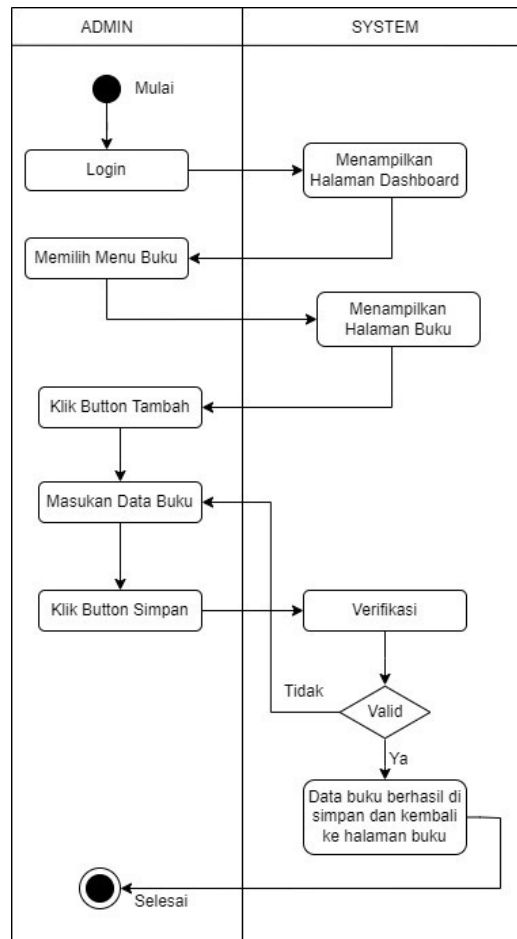
Gambar 3.5 *Activity Diagram* Hapus Data Sistem Website

Pada gambar 3.5 *Activity diagram* dimulai dengan login admin, lalu menuju *dashboard*. Admin memilih data untuk dihapus, mengklik tombol hapus untuk konfirmasi, lalu mengkonfirmasi penghapusan dengan tombol OK.



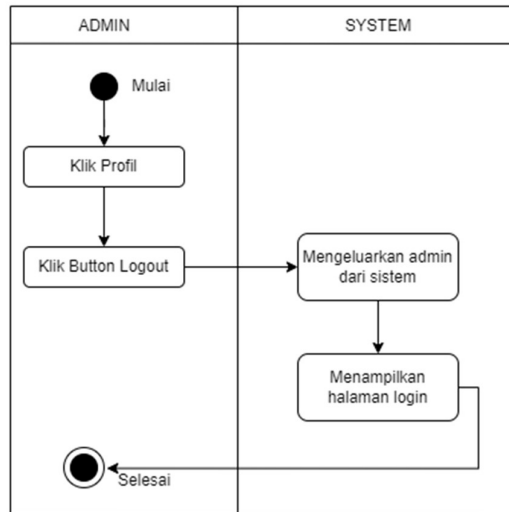
Gambar 3.6 *Activity Diagram* Cari Data Sistem Website

Pada gambar 3.6 *Activity diagram* dimulai dengan login admin, menuju dashboard. Admin mencari data dengan tombol cari dan memasukkan kata kunci untuk menampilkan hasil pencarian.



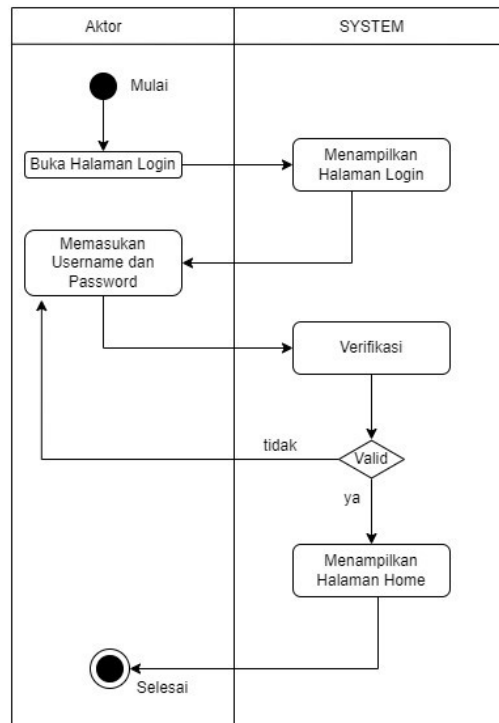
Gambar 3. 7 *Activity Diagram* Tambah Data Buku

Pada Gambar 3.7, *Activity diagram* dimulai dengan login admin, menuju dashboard. Admin pilih menu 'Buku', tambah data dengan memasukkan informasi buku, lalu simpan. Sistem verifikasi data; jika valid, informasi buku disimpan dan kembali ke halaman 'Buku'. Jika tidak, kembali ke halaman 'Tambah Data Buku'.



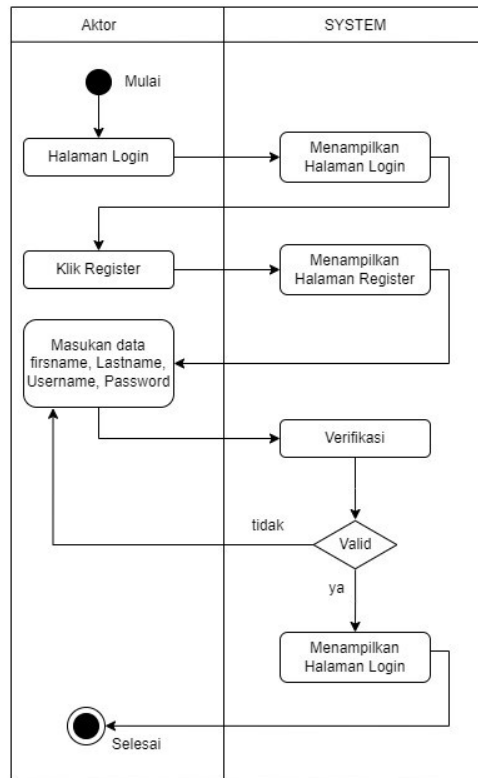
Gambar 3. 8 *Activity Diagram* Logout Sistem Website

Pada gambar 3.8 *Activity diagram* dimulai dengan admin mengklik profil, kemudian memilih tombol logout. Setelah itu, sistem akan mengeluarkan admin dari sistem dan mengarahkan kembali ke halaman login.



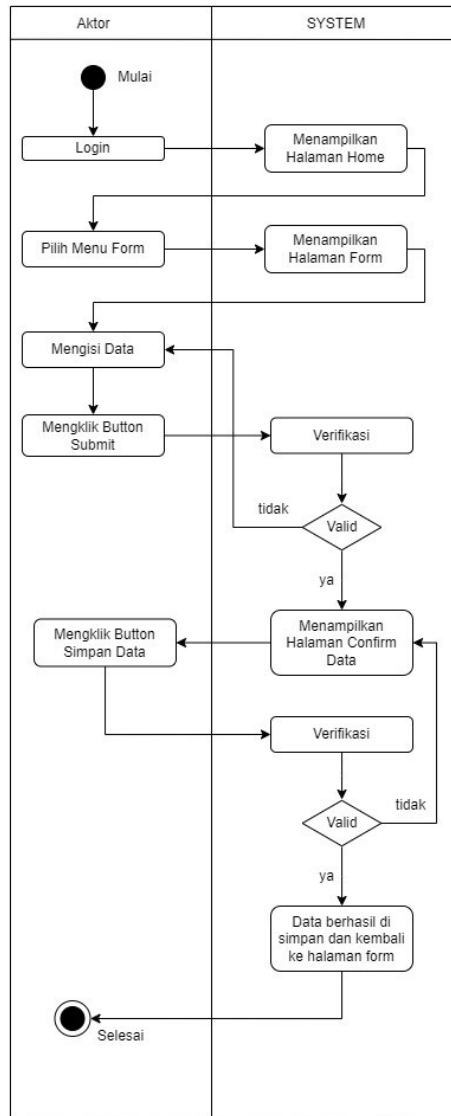
Gambar 3. 9 *Activity Diagram* Login Sistem Android

Pada gambar 3.9 *Activity diagram* dimulai dengan membuka halaman login. Setelah itu, aktor memasukkan username dan password. Pada tahap ini, aktor harus memastikan bahwa data yang dimasukkan benar. Jika data valid, sistem akan menampilkan halaman Home. Namun, jika data yang dimasukkan salah, CalonAnggota akan diarahkan kembali untuk memasukkan username dan password.



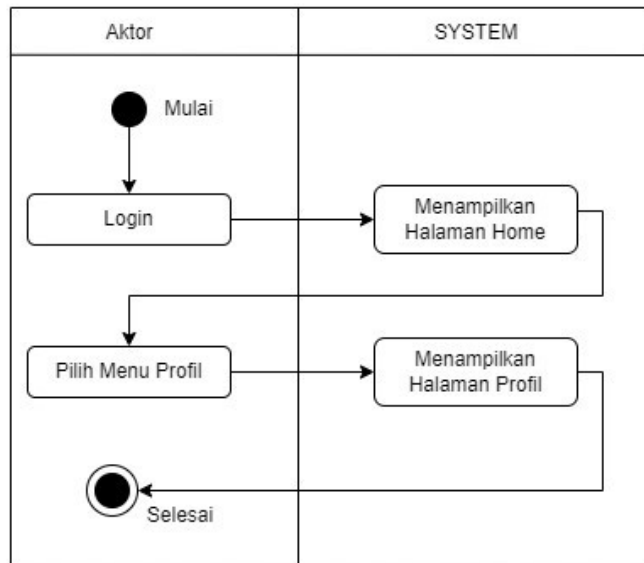
Gambar 3. 10 *Activity Diagram* Register Sistem Android

Pada gambar 3.10 *Activity diagram* dimulai dengan login. Aktor *register* dengan memasukkan *firstname*, *lastname*, *username*, dan *password*, memvalidasi data. Jika *valid*, sistem tampilkan halaman Login; jika tidak, aktor diminta masukkan data lagi.



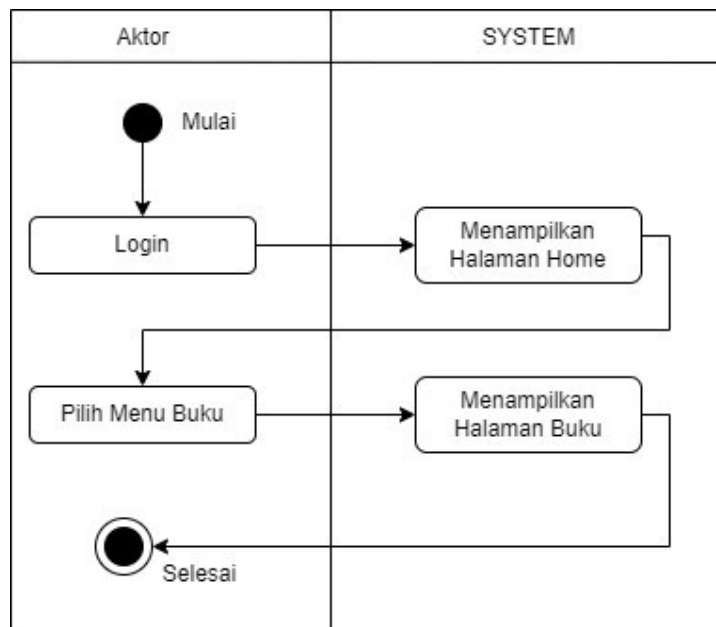
Gambar 3. 11 Activity Diagram Input Data Sistem Android

Pada gambar 3.11 *Activity diagram* dimulai dengan login aktor, menuju halaman home. Aktor pilih menu form, sistem tampilkan halaman form. Aktor isi dan submit data, sistem verifikasi. Jika tidak valid, aktor isi ulang. Jika valid, sistem menampilkan halaman konfirmasi data, aktor simpan. Sistem verifikasi lagi; tidak valid, ulang pengisian. Jika valid, data disimpan, aktor kembali ke halaman form.



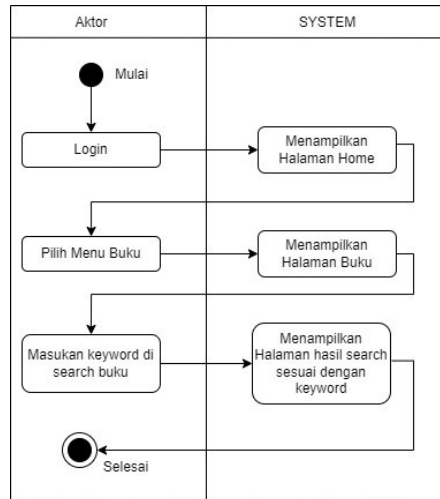
Gambar 3. 12 Activity Diagram Profil Sistem Android

Pada gambar 3.12 Activity diagram dimulai dengan proses login. Setelah berhasil login aktor akan memilih menu profil dan sistem akan menampilkan halaman profil.



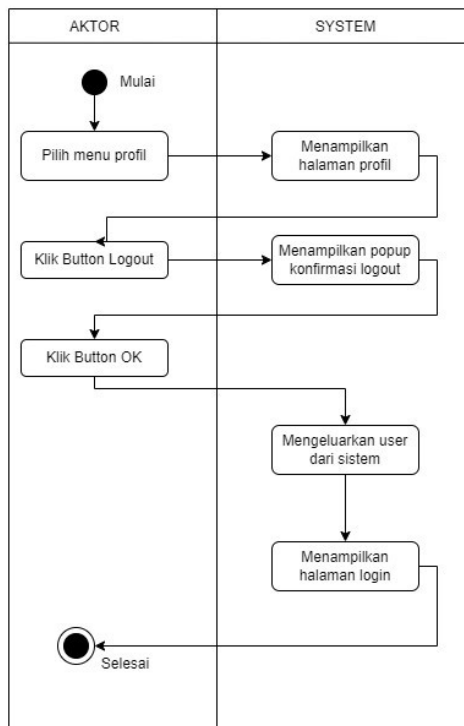
Gambar 3. 13 Activity Diagram Buku Sistem Android

Pada gambar 3.12 activity diagram dimulai dengan proses login. setelah berhasil login aktor akan memilih menu buku dan sistem akan menampilkan halaman buku.



Gambar 3. 14 Activity Diagram Search Buku Sistem Android

Pada Gambar 3.14, *activity diagram* dimulai dengan proses login. Setelah berhasil login, aktor akan memilih menu buku dan sistem akan menampilkan halaman buku. Aktor memasukkan *keyword* di pencarian buku, dan sistem akan menampilkan halaman hasil sesuai dengan *keyword* tersebut.



Gambar 3. 15 Activity Diagram Logout Sistem Android

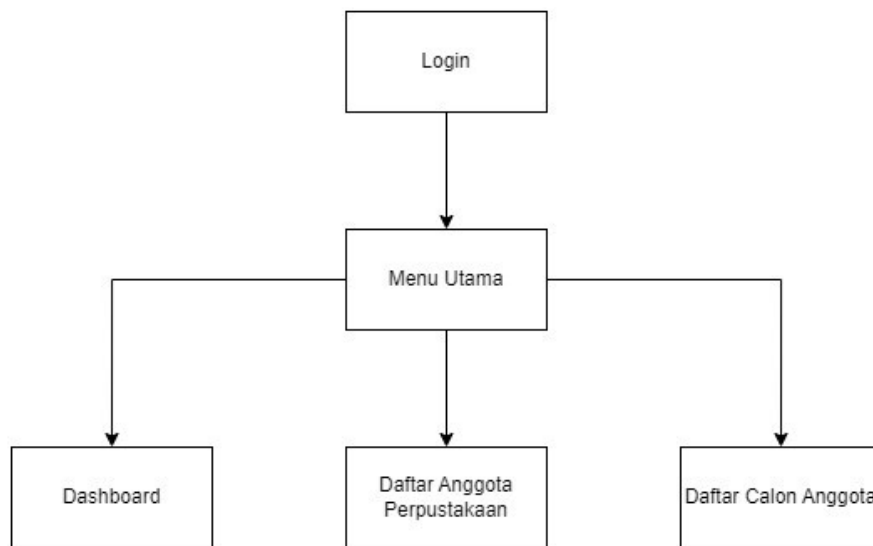
Pada gambar 3.15 *Activity diagram* dimulai dengan mengklik tombol *logout*. Selanjutnya, sistem akan menampilkan popup konfirmasi. Aktor kemudian mengklik tombol OK, dan sistem akan mengeluarkan aktor serta menampilkan halaman login.

3.4.2 Perancangan antarmuka

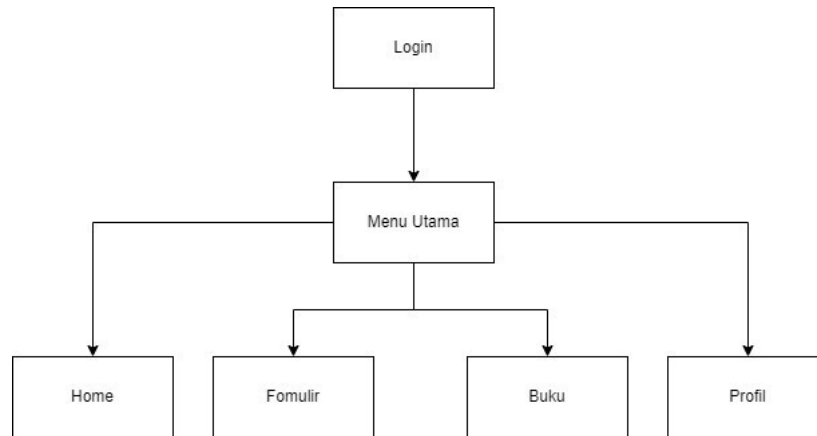
Dalam tahap desain antarmuka, *wireframe* dibuat menggunakan *software* Balsamiq Wireframes untuk merancang tampilan awal dan struktur navigasi aplikasi. *Wireframe* ini berfungsi untuk memvisualisasikan tata letak elemen-elemen antarmuka pengguna, termasuk tombol, menu, formulir, dan konten lainnya.

a. Struktur Menu

Susunan menu pada sistem pendaftaran anggota perpustakaan berbasis Android, baik untuk admin maupun calon anggota, dapat dilihat pada gambar 3.16 dan gambar 3.17.



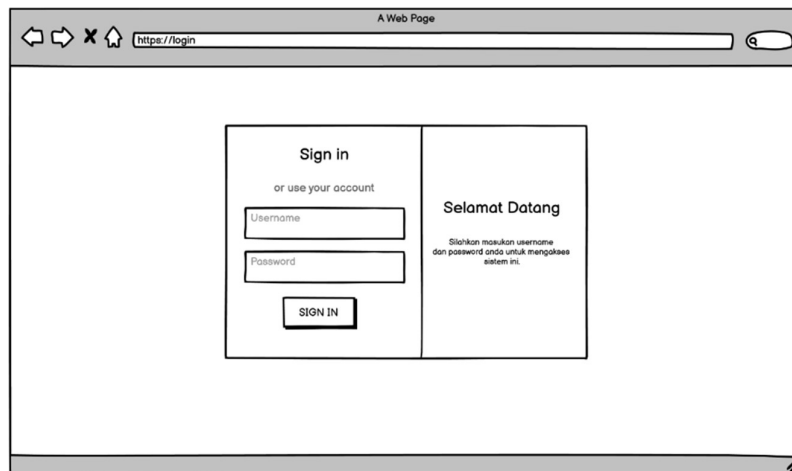
Gambar 3. 16 Rancangan struktur menu admin



Gambar 3. 17 Rancangan struktur menu CalonAnggota

b. Desain Wireframe Admin

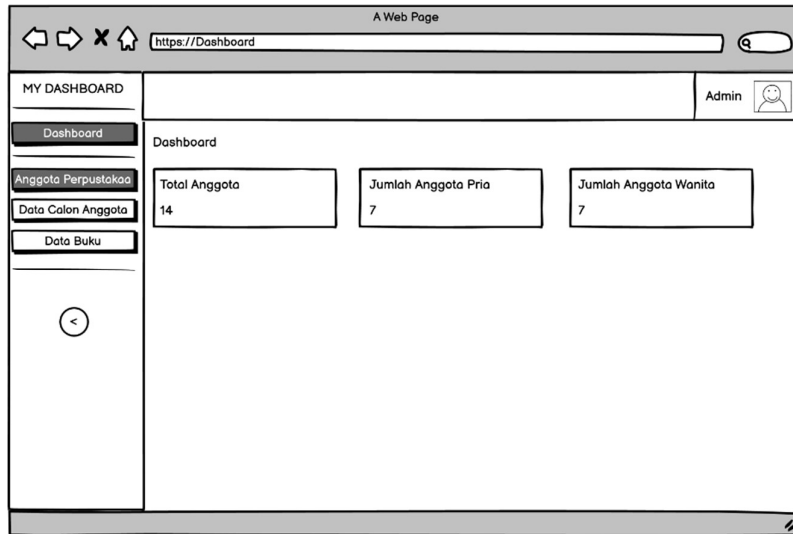
Pada gambar 3.18 *wireframe* login admin menampilkan elemen penting: kolom untuk *username*, *password*, dan tombol "*SIGN IN*" untuk *autentikasi*, memastikan akses hanya bagi pengguna dengan kredensial benar.



Gambar 3. 18 Wireframe *Login* Admin

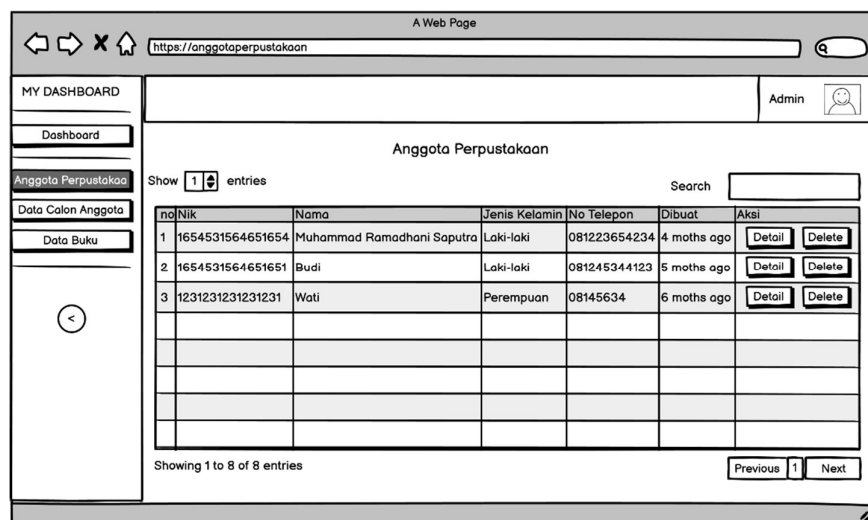
Pada gambar 3.19 menampilkan *wireframe* dashboard admin menampilkan statistik keanggotaan perpustakaan. Di bagian tengah, terdapat tiga kotak informasi ringkas: total anggota terdaftar (14 anggota), serta detail masing-masing 7 orang di kotak kedua dan ketiga.

Panel navigasi kiri memberikan akses ke bagian lain dashboard: daftar anggota, calon anggota, dan data buku. Pojok kanan atas memiliki ikon profil.



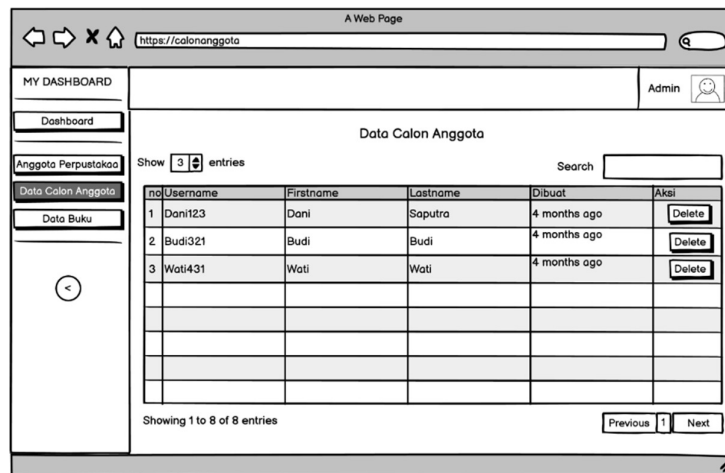
Gambar 3. 19 Wireframe *Dashboard* Admin

Pada gambar 3.20 menampilkan *wireframe* data anggota perpustakaan admin menampilkan tabel terstruktur. Di bagian atas halaman, terdapat baris pencarian dan opsi untuk jumlah entri per halaman. Tabel ini mencakup kolom nomor, NIK, nama anggota, jenis kelamin, nomor telepon, tanggal dibuat, dan aksi (hapus dan detail data).



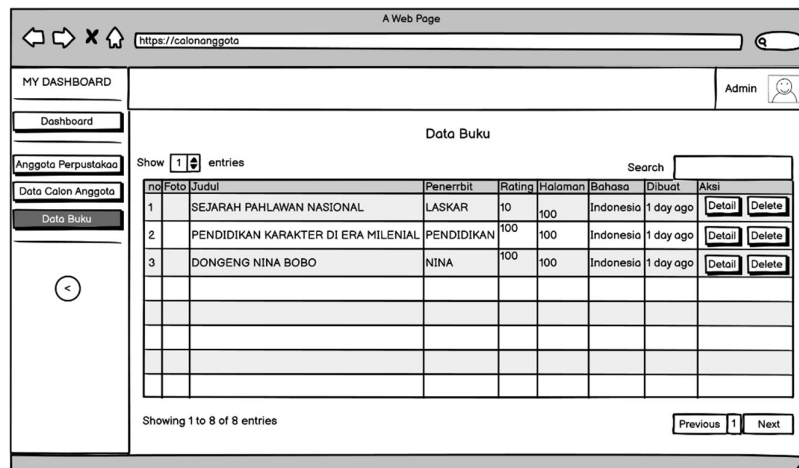
Gambar 3. 20 Wireframe Data Anggota Perpustakaan

Pada gambar 3.21 menampilkan *wireframe* data calon anggota perpustakaan admin menampilkan tabel dengan rincian informasi terstruktur. Di bagian atas halaman, terdapat baris pencarian dan opsi untuk jumlah entri per halaman. Tabel ini mencakup kolom nomor, *username*, *firstname*, *lastname*, tanggal dibuat, dan aksi (hapus data).



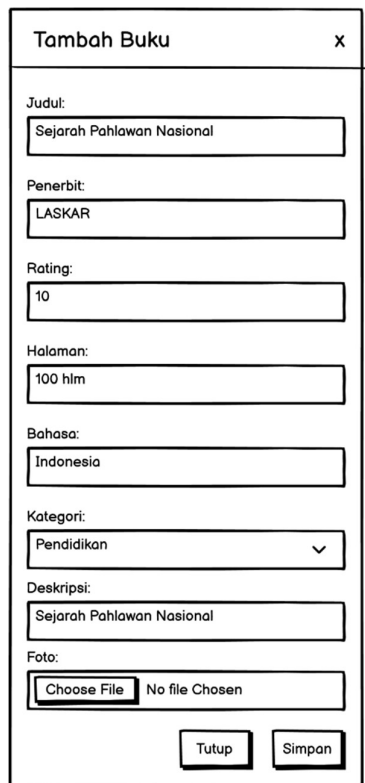
Gambar 3. 21 *Wireframe* Data Calon Anggota

Pada gambar 3.22 menampilkan *Wireframe* data buku perpustakaan admin menampilkan tabel dengan rincian informasi terstruktur. Di bagian atas halaman, terdapat baris pencarian dan opsi untuk jumlah entri per halaman. Tabel ini mencakup kolom nomor, foto, judul, penerbit, rating, halaman, bahasa, tanggal dibuat, dan aksi (hapus dan detail data).



Gambar 3. 22 *Wireframe* Data Buku

Gambar 3.23 menampilkan *wireframe* untuk menambahkan buku. Dalam *wireframe* ini, pengguna diminta mengisi informasi seperti judul, penerbit, rating, jumlah halaman, bahasa, memilih kategori dari dropdown, deskripsi, dan mengunggah foto. Terdapat tombol simpan untuk mengirimkan data ke sistem dan tombol tutup untuk menutup popup.



Tambah Buku		x
Judul:	<input type="text" value="Sejarah Pahlawan Nasional"/>	
Penerbit:	<input type="text" value="LASKAR"/>	
Rating:	<input type="text" value="10"/>	
Halaman:	<input type="text" value="100 hlm"/>	
Bahasa:	<input type="text" value="Indonesia"/>	
Kategori:	<input type="text" value="Pendidikan"/>	
Deskripsi:	<input type="text" value="Sejarah Pahlawan Nasional"/>	
Foto:	<input type="button" value="Choose File"/> No file Chosen	
<input type="button" value="Tutup"/>		<input type="button" value="Simpan"/>

Gambar 3. 23 *Wireframe* Tambah Buku

Pada gambar 3.24 menampilkan *wireframe* halaman untuk admin melihat detail data dari salah satu anggota perpustakaan, yang dimana detail data terdapat nik, nama, tempat/tanggal lahir, alamat sesuai KTP, alamat sekarang, no telepon, jenis kelamin, pendidikan, pekerjaan, dan email. Terdapat ikon X di atas kanan untuk keluar dari tampilan detail data.

Detail Data		x	
NIK:	1654531564651654	Nama:	MUHAMMAD RAMADHANI SAPUTRA
Tempat/Tanggal Lahir:	Samarinda, 02/02/2002		
Alamat Sesuai KTP:	ini alamat ktp	Alamat Sekarang:	ini alamat ktp
No Telepon:	0812315456121		
Jenis Kelamin	Laki-laki		
Pendidikan:	SMK	Pekerjaan	Mahasiswa
Email:	dummy12@gmail.com		

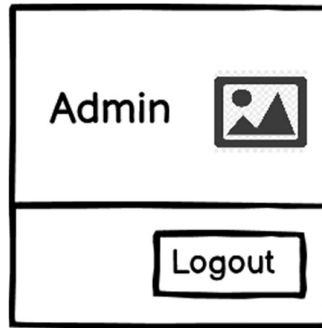
Gambar 3. 24 Wireframe Detail Data

Pada Gambar 3.25 menampilkan *wireframe* konfirmasi untuk menghapus data. Terdapat dua tombol: "Cancel" dan "OK". Jika tombol "Cancel" dipilih, data tidak akan dihapus. Sebaliknya, jika tombol "OK" dipilih, data akan dihapus dari sistem.


<p>Yakin ?</p> <p>Apakah kamu yakin menghapus data ini</p>
<input type="button" value="Cancel"/> <input type="button" value="OK"/>

Gambar 3. 25 Wireframe Hapus Data

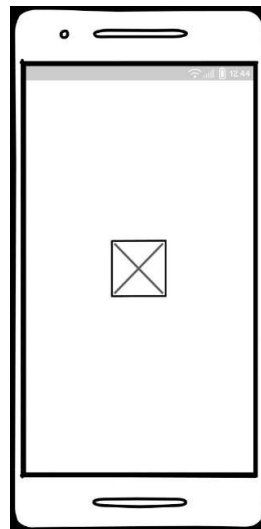
Pada Gambar 3.26 menampilkan *wireframe* profil admin. Terdapat dua bagian utama pada layar ini. Bagian atas menampilkan teks "Admin" di sebelah kiri dan sebuah ikon gambar di sebelah kanan. Bagian bawah memiliki sebuah tombol dengan label "Logout." Jika tombol "Logout" dipilih, pengguna akan keluar dari akun mereka dan kembali ke layar login.



Gambar 3. 26 *Wireframe Logout*

c. Tampilan Desain *Wireframe* Calon Anggota

Pada gambar 3.27 merupakan *wireframe splashscreen* untuk Android menampilkan logo aplikasi di tengah.



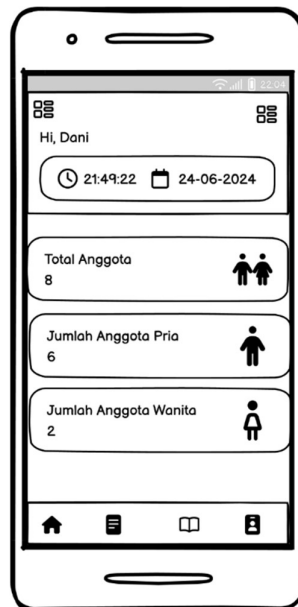
Gambar 3. 27 *Wireframe Halaman Splash screen*

Gambar 3.28 menampilkan *wireframe* halaman login dan register pada aplikasi Android. Calon anggota diharuskan memasukkan *username* dan *password* terlebih dahulu sebelum dapat mengakses sistem. Jika belum memiliki akun, mereka dapat mendaftar terlebih dahulu. Pada halaman registrasi, pengguna diminta mengisi beberapa data seperti nama depan (*firstname*), nama belakang (*lastname*), nama pengguna (*username*), dan kata sandi (*password*), serta menggunakan tombol registrasi untuk menyimpan data ke sistem.



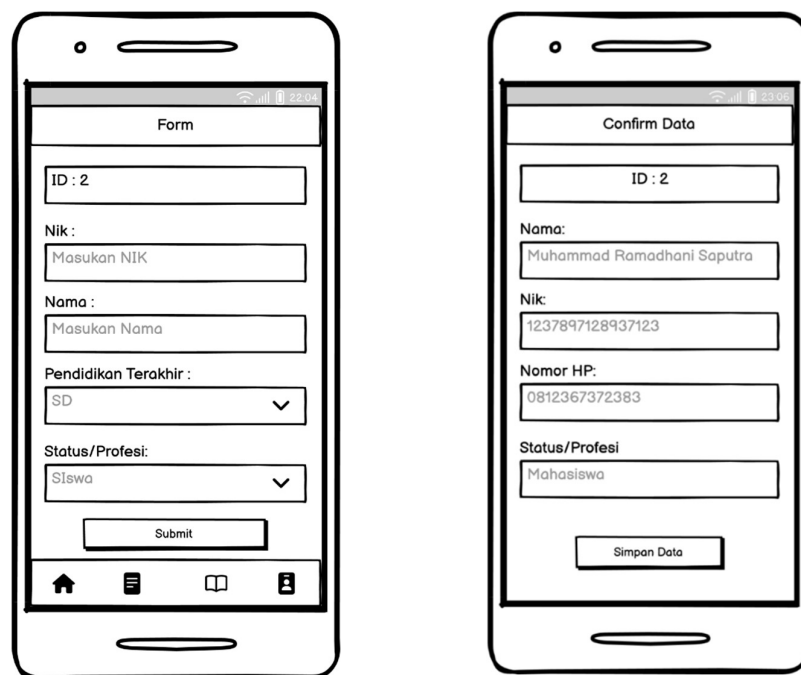
Gambar 3. 28 Wireframe Halaman Login dan Register

Pada Gambar 3.29 menampilkan wireframe halaman utama aplikasi Android menampilkan salam "Hi, Dani", waktu, dan tanggal saat ini. Terdapat tiga kartu informasi: total anggota (8), anggota pria (6), dan anggota wanita (2), dengan ikon yang sesuai. Di bagian bawah layar, terdapat bilah navigasi dengan empat ikon: home, form, buku, dan profil, mewakili berbagai fungsi aplikasi.



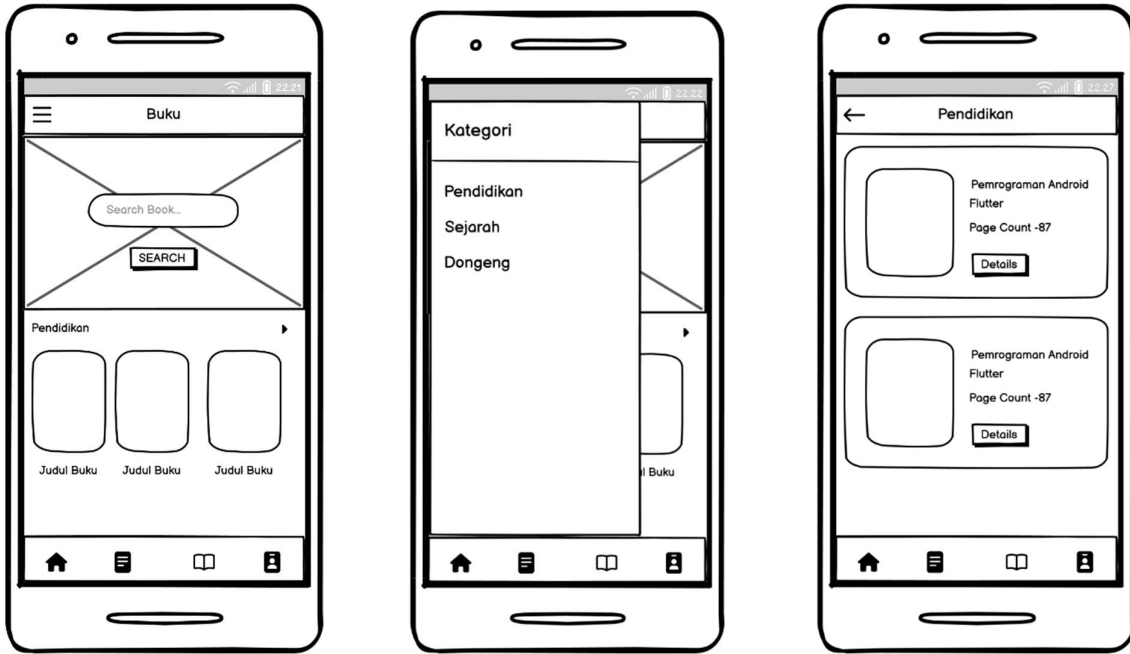
Gambar 3. 29 Wireframe Halaman Home

Pada gambar 3.30 menampilkan *wireframe* halaman formulir dan konfirmasi data pada aplikasi Android menampilkan dua bagian. Pada halaman formulir, pengguna diminta mengisi informasi seperti ID, NIK, Nama, dengan opsi *dropdown* untuk Pendidikan Terakhir dan Status/Profesi. Terdapat tombol Submit untuk mengirimkan data ke halaman konfirmasi. Di halaman konfirmasi, pengguna diminta memeriksa dan mengkonfirmasi data yang telah diinput sebelumnya, seperti ID, Nama, NIK, Nomor HP, dan Status/Profesi, sebelum menggunakan tombol Simpan Data untuk menyimpan data ke sistem.



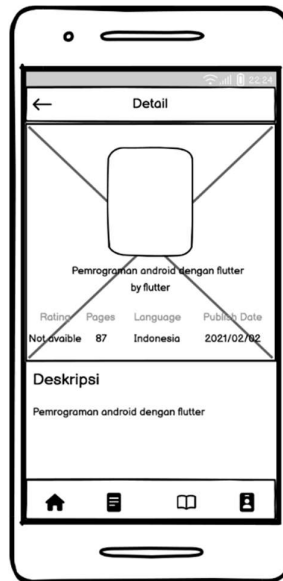
Gambar 3. 30 *Wireframe* Halaman *Form* dan Konfirmasi Data

Gambar 3.31 menampilkan *wireframe* halaman buku dan kategori dengan fitur input pencarian buku, serta sidebar yang terletak di bagian atas kiri. Sidebar tersebut menunjukkan kategori-kategori seperti pendidikan, sejarah, dan dongeng.



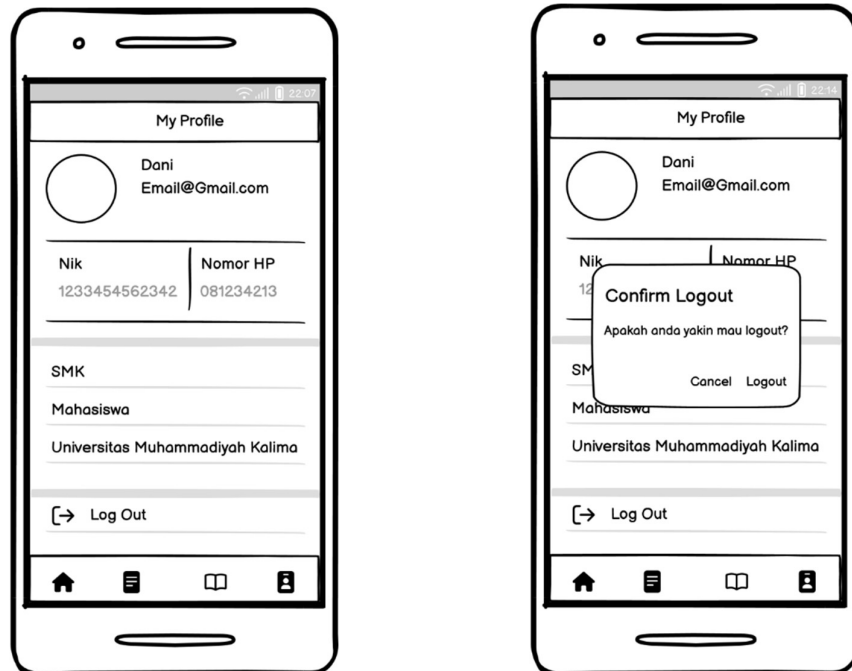
Gambar 3. 31 Wireframe Halaman Buku dan kategori

Pada gambar 3.32 menampilkan wireframe halaman detail buku yang berisi gambar buku, judul buku, rating, pages, language, publish date, dan deskripsi.



Gambar 3. 32 Wireframe Halaman Detail Buku

Pada Gambar 3.33 menampilkan *wireframe* halaman profil dan logout pengguna di aplikasi Android menampilkan informasi pribadi seperti foto profil, nama, dan email. Di bawahnya, terdapat detail tambahan seperti NIK, Nomor HP, pendidikan terakhir, status/profesi, institusi pendidikan, alamat, dan jenis kelamin. Bagian navigasi terletak di bagian bawah layar dengan ikon menuju home, form, buku, dan profil.



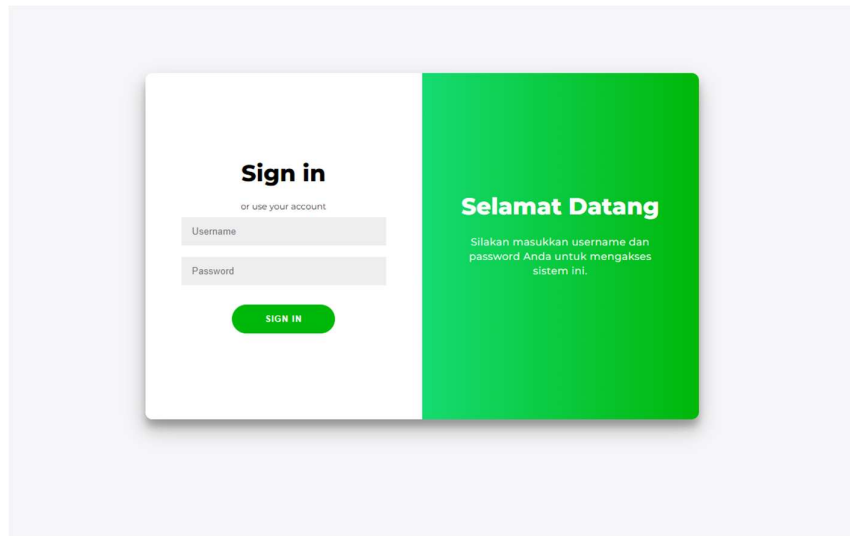
Gambar 3. 33 *Wireframe* Halaman Profil dan Logout

3.3 Implementasi (*Implementation*)

Implementasi sistem ini bertujuan untuk memberikan panduan kepada pengguna mengenai cara penggunaan sistem tersebut, sehingga pengguna dapat mengoperasikannya dengan mudah.

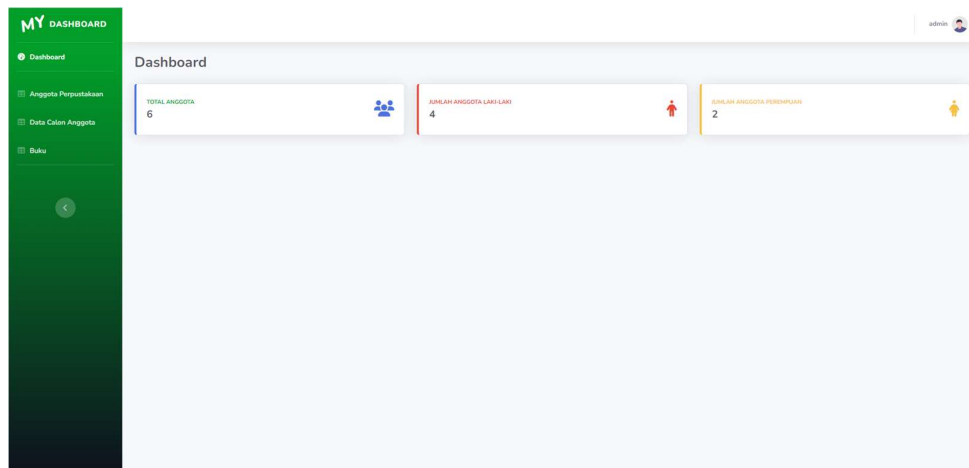
3.3.1 Halaman Admin

Gambar 3.34 menampilkan halaman *login* untuk *website*. Pengguna harus memasukkan *username* dan *password*. Jika *username* atau *password* salah, pengguna tidak dapat mengakses sistem.



Gambar 3. 34 Tampilan Halaman *Login* website

Gambar 3.35 menampilkan halaman *dashboard* website. Pengguna akan langsung melihat data, seperti total anggota, jumlah anggota laki-laki, dan jumlah anggota perempuan. Panel navigasi di sebelah kiri menyediakan akses mudah ke berbagai bagian lain dari dashboard, seperti daftar anggota perpustakaan, data calon anggota dan data buku. Di bagian kanan atas, terdapat nama dan profil admin.



Gambar 3. 35 Tampilan Halaman *Dashboard* website

Gambar 3.36 menampilkan halaman "Data Anggota" pada website. Pada halaman ini, pengguna dapat melihat tabel yang berisi informasi anggota perpustakaan, termasuk NIK, nama, jenis kelamin, nomor telepon, tanggal pembuatan akun, serta opsi untuk melihat detail

atau menghapus data anggota. Fitur pencarian di sudut kanan atas tabel memungkinkan pengguna untuk mencari data anggota dengan cepat.

no	Nik	Nama	Jenis Kelamin	No Telp	Dibuat	Aksi
1	1654531564651654	MUHAMMAD RAMADHANI SAPUTRA	Laki-laki	0812315456121	4 months ago	Detail Delete
2	0821488885656565	ADITYA PRANATA	Laki-laki	082148	4 days ago	Detail Delete
3	5454364646454464	DANI	Laki-laki	0812535950886	1 day ago	Detail Delete
4	6478054632158467	ABRAHAM	Laki-laki	0812563455789	6 minutes ago	Detail Delete
5	6472036455545488	ANSEL	Perempuan	0812534694987	4 minutes ago	Detail Delete
6	6470846345484649	OLIVER	Perempuan	0822451345464	31 seconds ago	Detail Delete

Gambar 3. 36 Tampilan Halaman Data Anggota Perpustakaan website

Gambar 3.37 menampilkan "Detail Data" yang muncul ketika pengguna mengklik tombol "Detail" pada halaman "Data Anggota". Popup ini memperlihatkan informasi lengkap mengenai seorang anggota, seperti NIK, nama, tempat/tanggal lahir, alamat sesuai KTP, alamat sekarang, nomor telepon, jenis kelamin, pendidikan, pekerjaan, dan email. Informasi ini ditampilkan dalam bentuk form yang memudahkan pengguna untuk melihat detail dari setiap anggota secara menyeluruh.

Detail Data

NIK: 6472084649499897

Nama: MUHAMMAD RAMADHANI SAPUTRA

Tempat/Tanggal lahir: Samarinda, 13/11/2002

Alamat Sesuai KTP: ini alamat ktp

Alamat Sekarang: ini alamat ktp

No Telepon: 0822457867646

Jenis Kelamin: Laki-laki

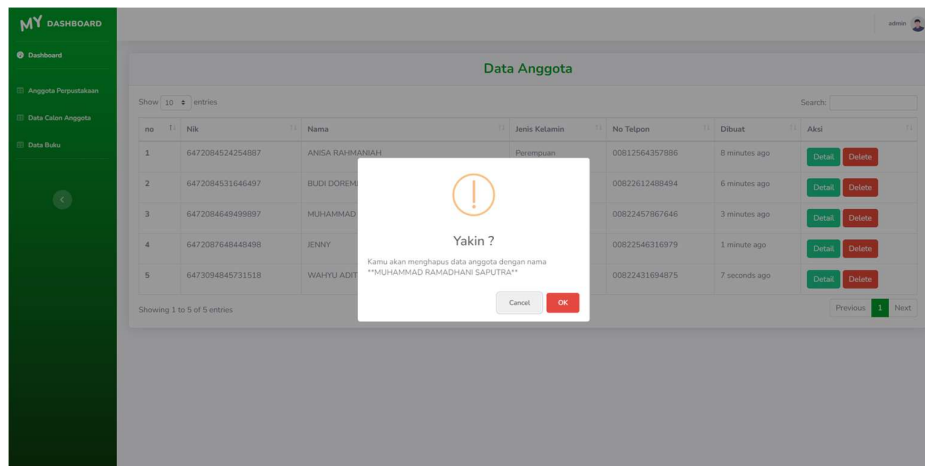
Pendidikan: SMK

Pekerjaan: Mahasiswa

Email: [input field]

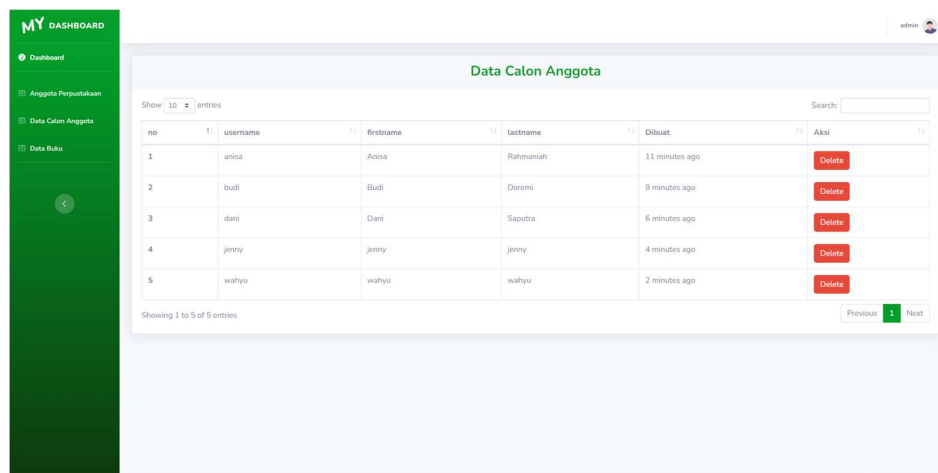
Gambar 3. 37 Tampilan Detail Data Anggota Perpustakaan website

Gambar 3.38 menampilkan kotak dialog konfirmasi penghapusan data anggota yang muncul saat pengguna mengklik tombol "Delete" pada halaman "Data Anggota". Kotak dialog ini memberikan peringatan kepada pengguna dengan pesan "Yakin?" diikuti dengan konfirmasi bahwa pengguna akan menghapus data anggota dengan nama yang ditentukan, dalam contoh ini adalah "Muhammad Ramadhani Saputra". Pengguna diberikan dua opsi: tombol "Cancel" untuk membatalkan tindakan penghapusan dan tombol "OK" untuk melanjutkan penghapusan.



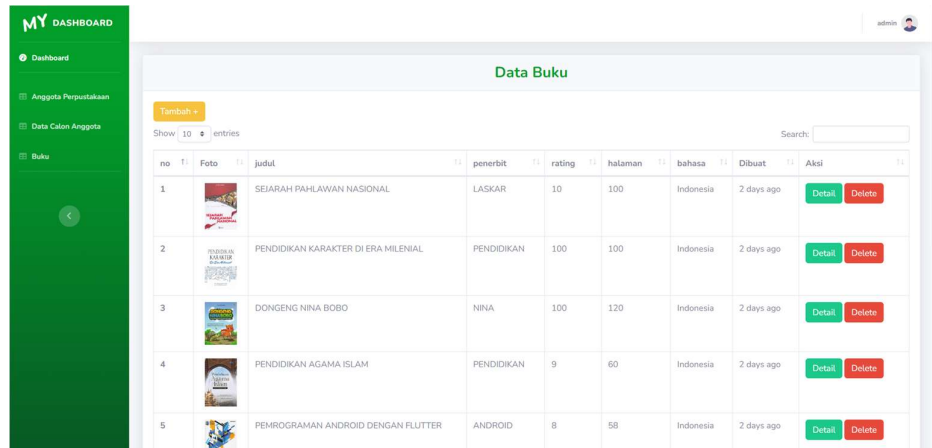
Gambar 3. 38 Tampilan Hapus Data website

Gambar 3.39 menampilkan halaman data calon anggota pada website. Pada halaman ini, pengguna dapat melihat tabel yang berisi informasi data calon anggota, termasuk username, firstname, lastname, serta opsi untuk menghapus data anggota.



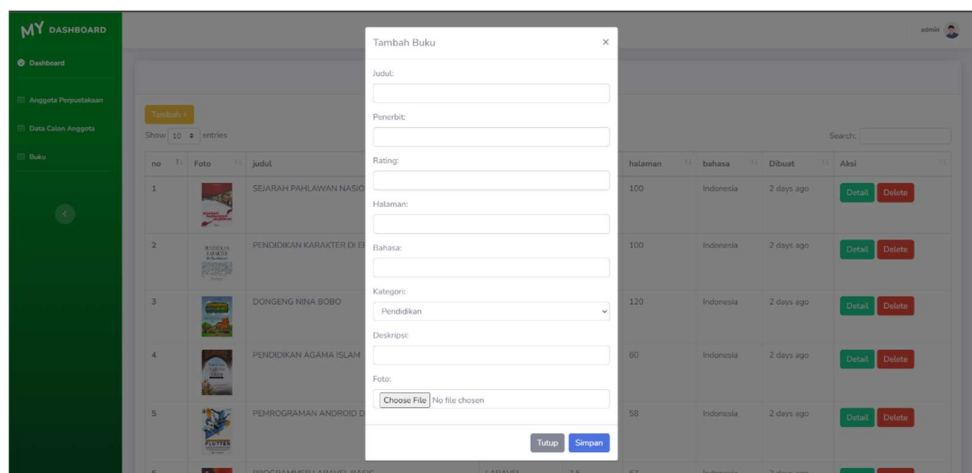
Gambar 3. 39 Tampilan Halaman Data Calon Anggota Website

Gambar 3.40 menampilkan halaman daftar buku pada website. Pada halaman ini, pengguna dapat melihat tabel yang berisi informasi buku perpustakaan, termasuk foto, judul, penerbit, rating, halaman, bahasa, dibuat serta opsi untuk melihat detail atau menghapus data buku. Fitur pencarian di sudut kanan atas tabel memungkinkan pengguna untuk mencari data buku dengan cepat.



Gambar 3. 40 Tampilan Halaman Data Buku

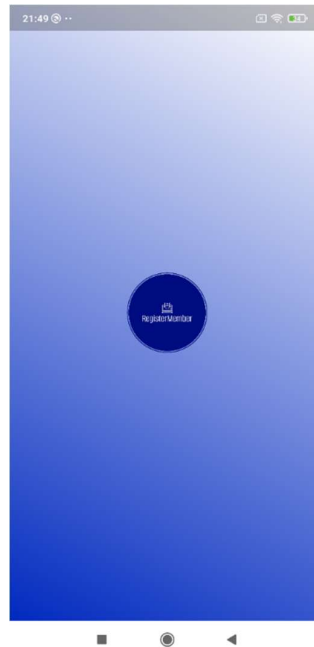
Gambar 3.41 menampilkan halaman input buku yang memerlukan pengguna untuk mengisi judul, penerbit, rating, jumlah halaman, bahasa, kategori, deskripsi, serta mengunggah foto buku. Terdapat pula tombol simpan yang berfungsi untuk menyimpan data ke sistem.



Gambar 3. 41 Tampilan Input buku

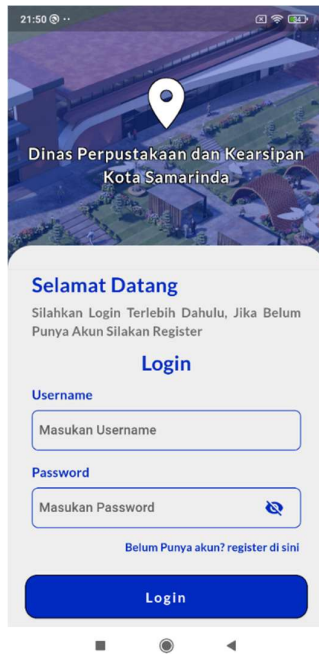
3.3.2 Halaman Calon Anggota

Gambar 3.42 menampilkan halaman splash screen aplikasi "*RegisterMember*". Pada halaman ini, terdapat sebuah logo dan teks "*RegisterMember*" yang terletak di tengah layar dengan latar belakang gradien biru. Halaman ini berfungsi sebagai tampilan pembuka saat aplikasi pertama kali dijalankan.



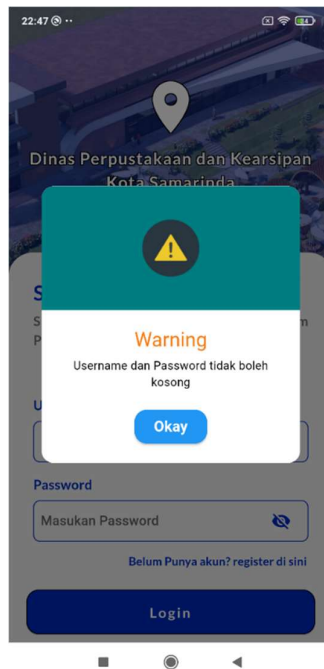
Gambar 3. 42 Tampilan *Splash Screen*

Gambar 3.43 menampilkan halaman *login* aplikasi. Pada halaman ini, terdapat text dengan nama instansi di bagian atas, diikuti dengan pesan selamat datang yang mengarahkan pengguna untuk login terlebih dahulu atau mendaftar jika belum memiliki akun. Terdapat dua kolom input untuk username dan password, serta sebuah tombol "*Login*" untuk melanjutkan proses masuk ke aplikasi. Selain itu, ada juga tautan untuk pengguna yang belum memiliki akun agar dapat melakukan pendaftaran.



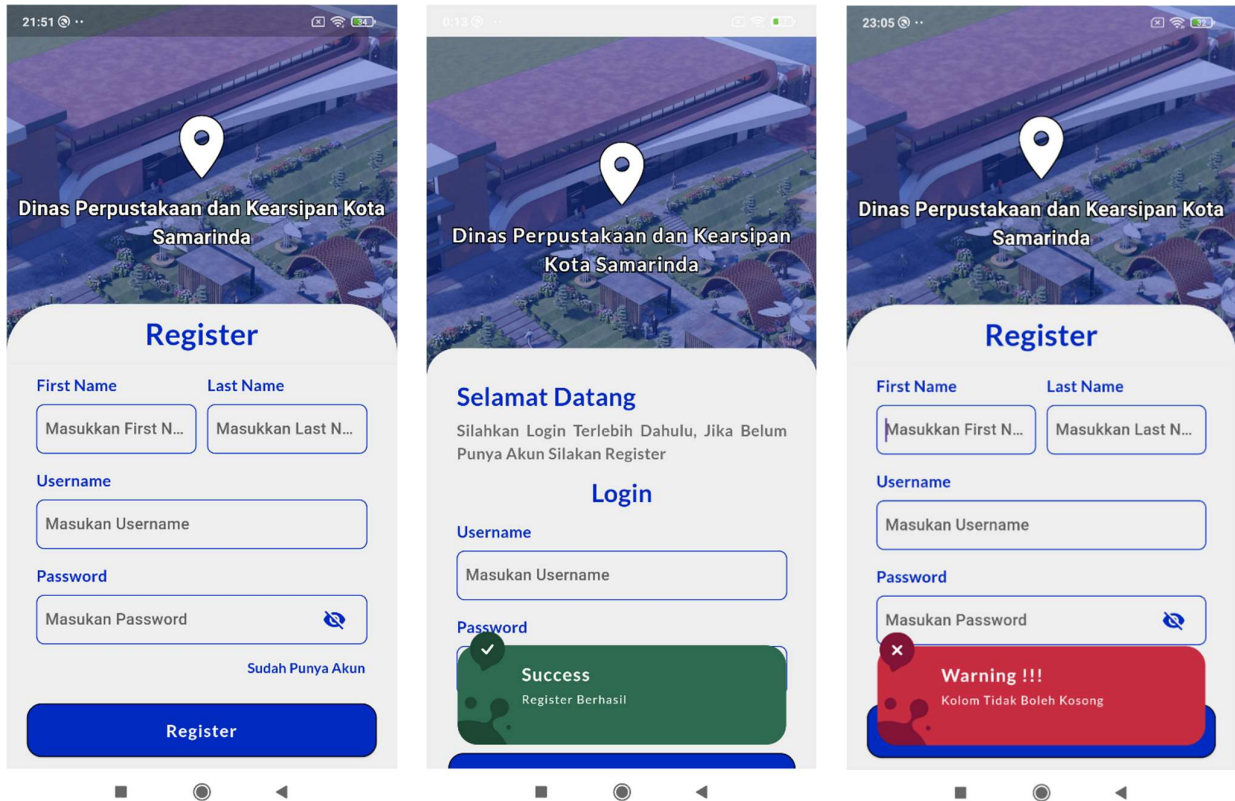
Gambar 3. 43 Tampilan Halaman *Login*

Gambar 3.44 menampilkan pesan kesalahan yang muncul saat pengguna tidak memasukkan username atau password, dilengkapi dengan tombol 'OK' untuk menutup pesan kesalahan tersebut.



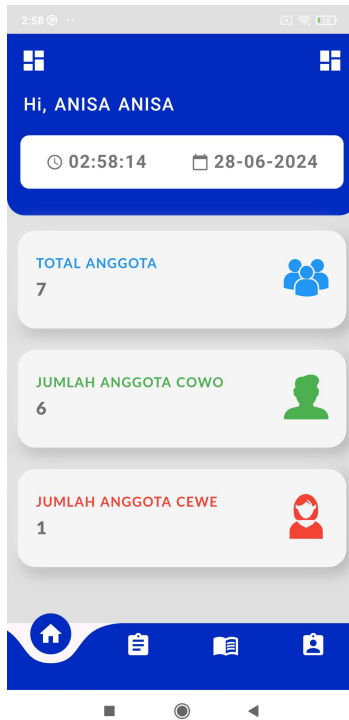
Gambar 3. 44 Tampilan Kesalahan login

Gambar 3.45 menunjukkan halaman pendaftaran dengan kolom untuk *firstname*, *lastname*, *username*, dan *password*, serta tombol 'Register'. Sistem memberikan pemberitahuan keberhasilan pendaftaran atau pesan kesalahan jika ada kolom yang tidak diisi.



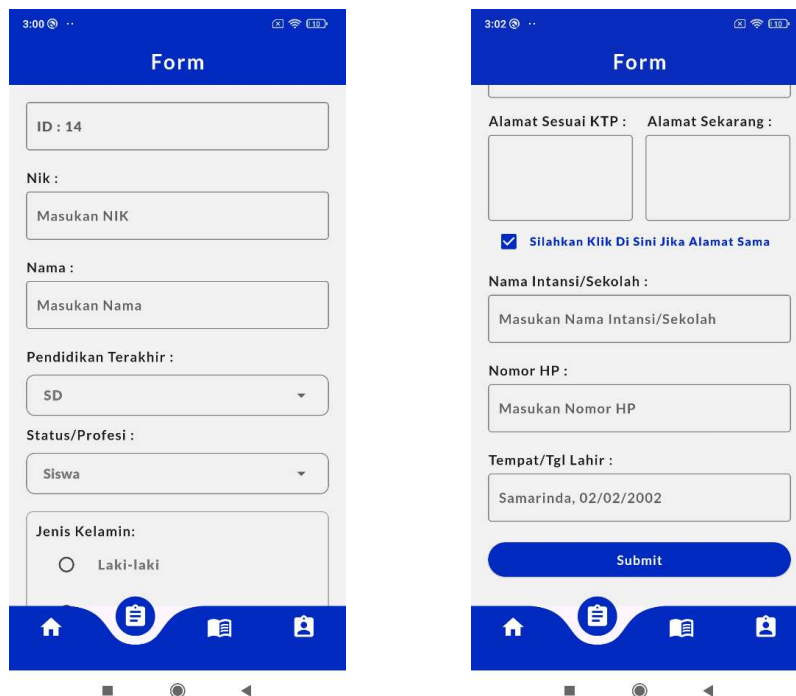
Gambar 3. 45 Tampilan Halaman *Register*

Gambar 3.46 menampilkan halaman *home*. Pada halaman ini terdapat informasi nama pengguna, disertai dengan jam dan tanggal saat ini. Terdapat juga tiga data statistik, mencakup total anggota, jumlah anggota pria, jumlah anggota wanita. Di bagian bawah terdapat menu navigasi *home*, *formulir*, *bukut* dan *profil*.



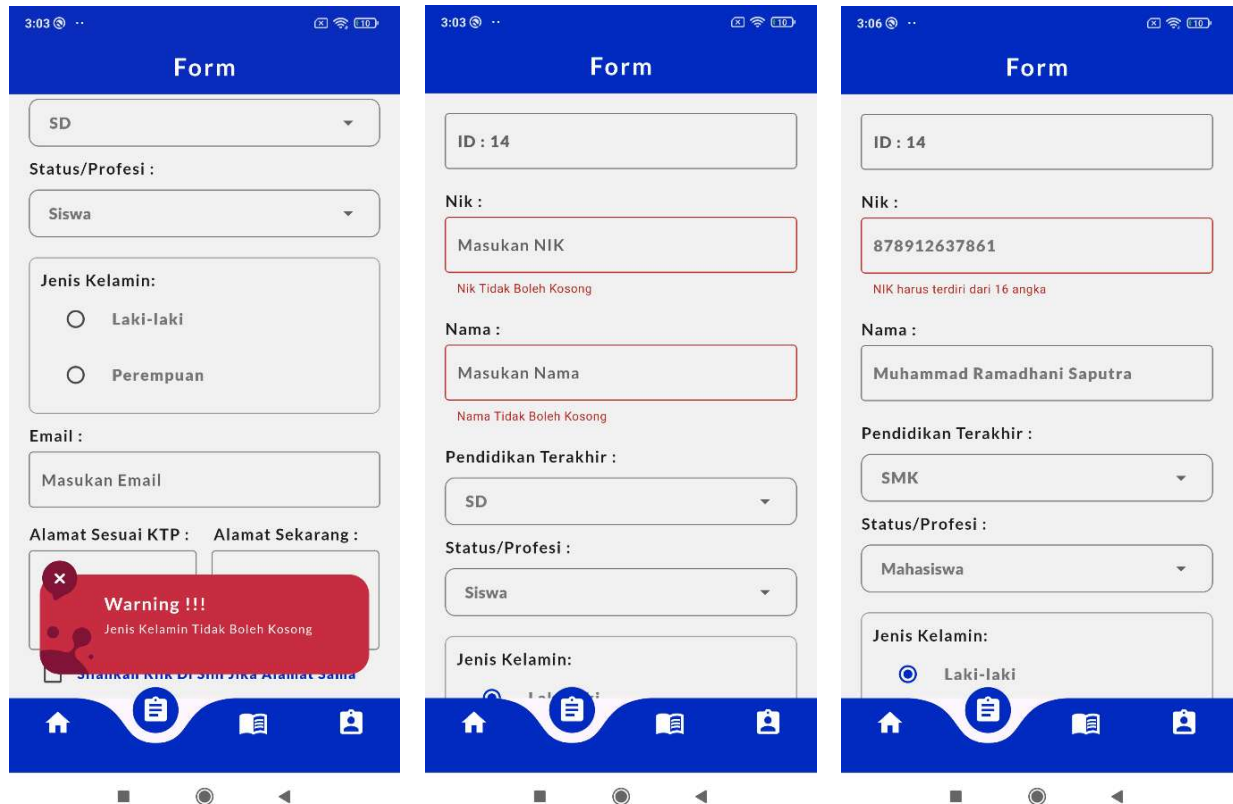
Gambar 3. 46 Tampilan Halaman *Home*

Gambar 3.47 menampilkan halaman formulir. Pada halaman ini pengguna menginput data sesuai dengan kriterianya dan juga terdapat tombol submit untuk menuju ke halaman confirm data, bisa di lihat pada gambar 3.50.



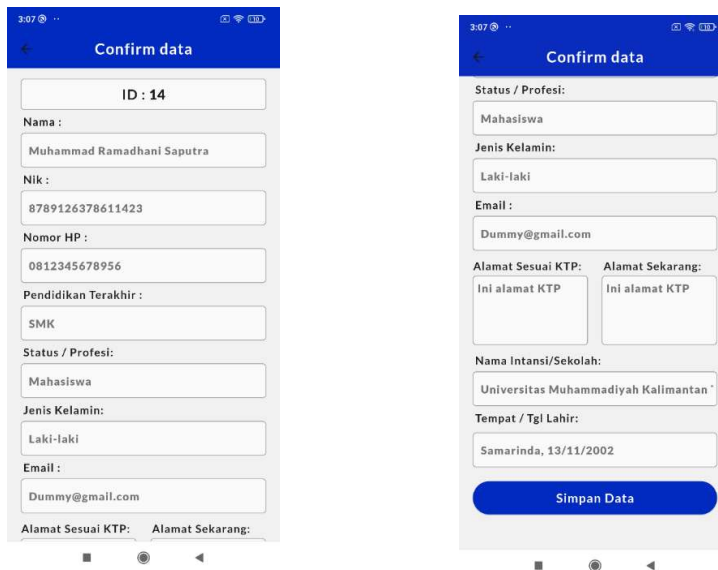
Gambar 3. 47 Tampilan Halaman Formulir

Gambar 3.48 menampilkan proses verifikasi pada halaman formulir. Jika pengguna belum mengisi jenis kelamin, sistem akan menampilkan pesan kesalahan. Selain itu, jika pengguna tidak mengisi seluruh kolom atau jika nomor identitasnya kurang dari 16 digit, sistem akan memberikan pemberitahuan kesalahan pada kolom input yang bersangkutan.



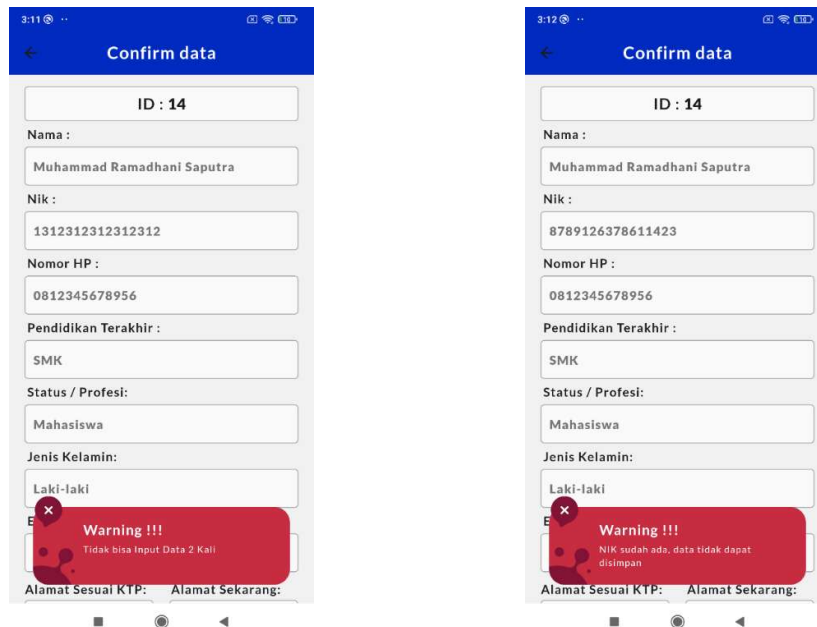
Gambar 3. 48 Tampilan Kesalahan Halaman Formulir

Gambar 3.49 menampilkan halaman konfirmasi data, yang berfungsi untuk memungkinkan pengguna untuk meninjau dan mengkonfirmasi data sebelum data tersebut dimasukkan ke dalam sistem.



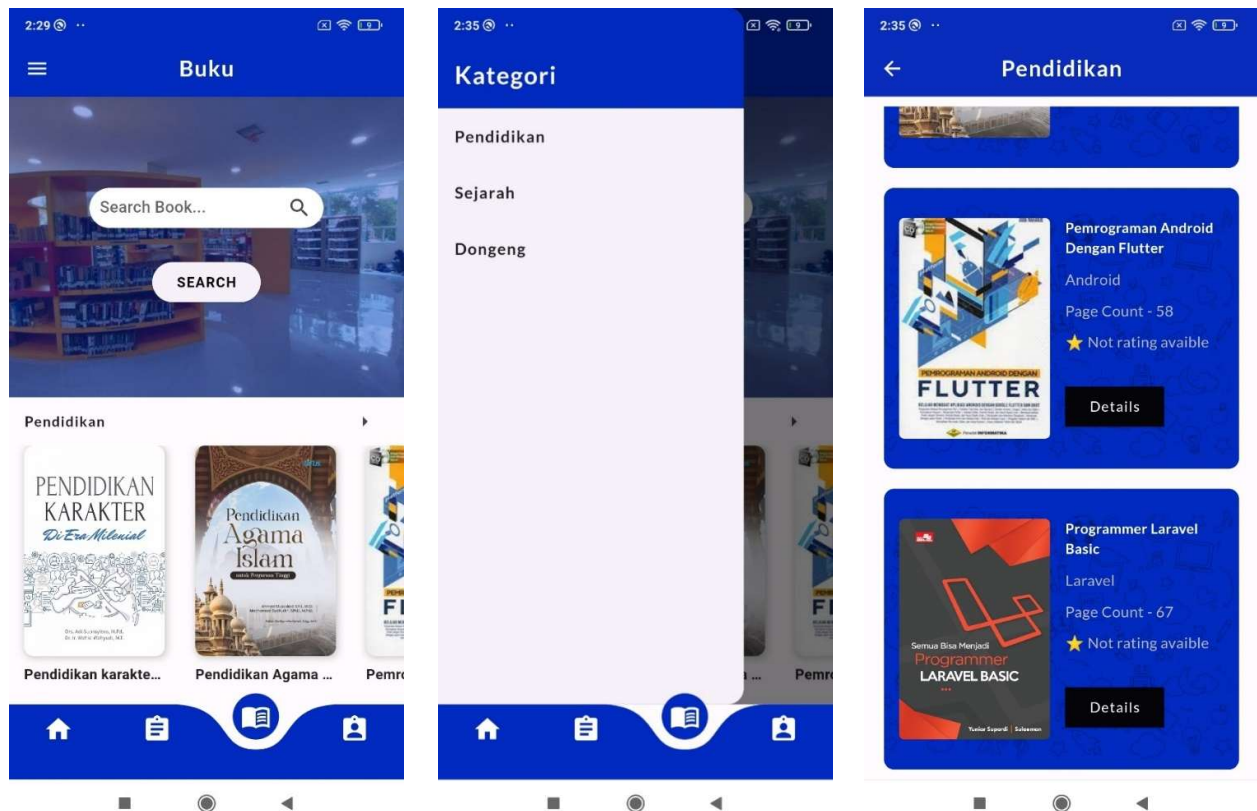
Gambar 3. 49 Tampilan Halaman Confirm Data

Gambar 3.50 menampilkan proses verifikasi pada halaman konfirmasi data. Jika pengguna telah menginputkan data sebelumnya, sistem akan menampilkan pesan kesalahan yang menyatakan bahwa pengguna tidak dapat menginputkan data lebih dari sekali. Selain itu, jika NIK (Nomor Induk Kependudukan) sudah terdaftar dalam database sistem, pesan kesalahan akan ditampilkan.



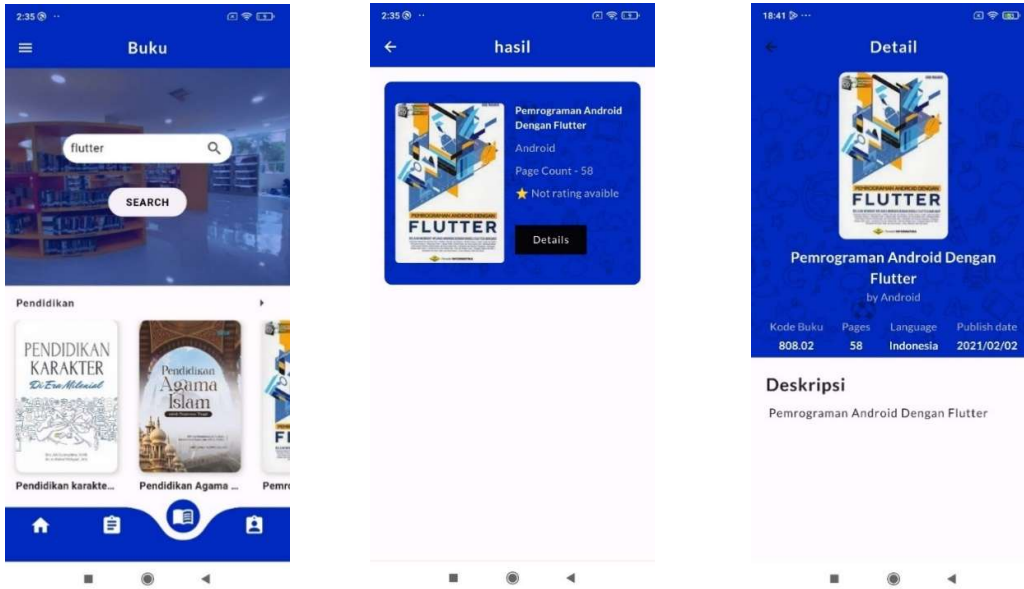
Gambar 3. 50 Tampilan Kesalahan Halaman Confirm Data

Gambar 3.51 menampilkan halaman buku dan kategori yang dilengkapi dengan sidebar untuk melihat kategori, fitur pencarian buku, dan tampilan buku dengan kategori pendidikan. Di samping itu, halaman tersebut menampilkan kategori buku seperti pendidikan, sejarah, dan dongeng. Selain itu, halaman hasil kategori menampilkan semua kategori yang dipilih oleh pengguna.



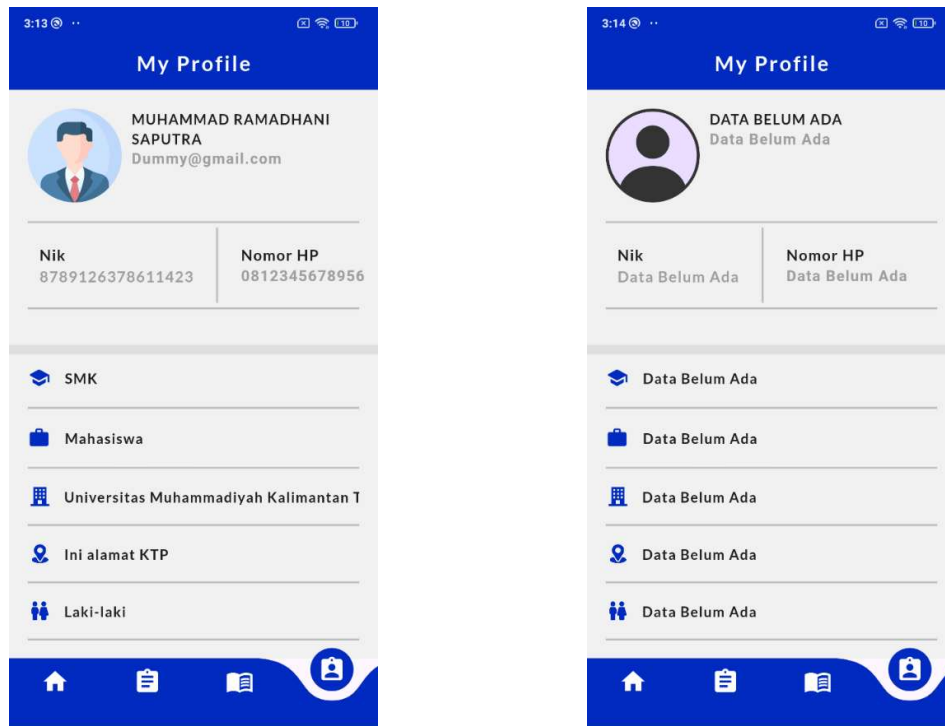
Gambar 3. 51 Tampilan Halaman Buku dan Halaman Kategori

Gambar 3.52 menampilkan hasil pencarian buku dengan kata kunci 'flutter' dan halaman detail buku. Di halaman hasil, terdapat buku-buku yang sesuai dengan kata kunci tersebut, masing-masing dilengkapi dengan tombol 'Details' untuk melihat detail lengkapnya. Halaman detail buku menampilkan informasi seperti gambar buku, judul, kode buku, jumlah halaman, bahasa, tanggal terbit, dan deskripsi buku.



Gambar 3. 52 Tampilan Search Buku dan Halaman Detail Buku

Gambar 3.53 menampilkan halaman profil. Data yang terdapat di halaman profil ini telah diinputkan sebelumnya pada halaman formulir, yang dapat dilihat di Gambar 3.41. Jika pengguna belum mengisi data pada halaman formulir, maka halaman profil akan menampilkan data kosong.



Gambar 3. 53 Tampilan Halaman Profil

3.4 Pengujian (*Testing*)

Penulis melakukan blackbox testing untuk mengevaluasi fungsionalitas fitur-fitur pada sistem website dan aplikasi Android. Pengujian aplikasi dilakukan oleh Pak Andri, staff otomasi perpustakaan berusia 38 tahun dari Samarinda.

3.4.1 Pengujian Sistem Website

Tabel 3. 12 Pengujian Sistem Website

No	Test case	Langkah uji	Hasil yang diharapkan	Hasil yang didapatkan	Keterangan
1	Login admin	Memasukan username dan password	Sistem menampilkan halaman dashboard	Sistem berhasil menampilkan dashboard	Berhasil
2	Login Admin	Memasukan username atau password salah	sistem menampilkan pemberitahuan bahwa usename dan password salah	Sistem berhasil menampilkan pemberitahuan kesalahan data	Berhasil
3	Lihat detail data	Klik button detail	Sistem akan menampilkan popup yang berisi detail data	Sistem berhasil menampilkan popup detail data	Berhasil
4	Hapus data	Klik button delete dan pilih ok	Sistem akan menampilkan popup hapus data dan data berhasil dihapus	Sistem berhasil menampilkan popup hapus data dan data berhasil dihapus	Berhasil
5	fitur search	Masukan keyword pencarian dan klik enter	Sistem akan menampilkan semua data yang sesuai dengan keyword	Sistem berhasil menampilkan data sesuai dengan keyword	Berhasil
6		Masukan keyword pencarian dengan data yang salah	Sistem tidak akan menampilkan data yang sesuai dengan keyword	Sistem berhasil untuk tidak menampilkan data keyword yang salah	Berhasil
7	Show entries	Pilih berapa entry yang akan ditampilkan di halaman	Sistem akan menampilkan entry sesuai dengan jumlah yang dipilih	Sistem berhasil menampilkan entry sesuai dengan jumlah yang dipilih	Berhasil
8	Logout	Klik button logout	Sistem akan menampilkan popup untuk konfirmasi logout dan jika berhasil akan menampilkan halaman login	Sistem berhasil menampilkan popup konfirmasi logout dan berhasil menuju halaman login	Berhasil

3.4.2 Pengujian Sistem Android

Tabel 3. 13 Pengujian Sistem Android

No	Test case	Langkah uji	Hasil yang diharapkan	Hasil yang didapatkan	Keterangan
1	Login user	Memasukan username dan password	Sistem menampilkan halaman home	Sistem berhasil menampilkan halaman home	Berhasil
2		Memasukan username dan password salah	sistem menampilkan pemberitahuan bahwa username dan password salah	Sistem berhasil menampilkan pemberitahuan kesalahan data	Berhasil
3	Register	Buka halaman register dan isi data yang diperlukan jika sudah klik simpan data	Sistem akan menampilkan notifikasi bahwa data berhasil register dan diarahkan ke halaman login	Sistem berhasil menampilkan notifikasi bahwa data berhasil register dan diarahkan ke halaman login	Berhasil
4		Masih ada data yang belum diisi	Sistem akan menampilkan notifikasi kesalahan	Sistem berhasil menampilkan notifikasi kesalahan	Berhasil
5	Mengisi formulir	Memasukan semua data yang diperlukan	Sistem akan menampilkan halaman confirm data	Sistem berhasil menampilkan halaman confirm data	Berhasil
6		Masih ada data yang belum diisi atau masih ada kriteria yang belum dipenuhi	Sistem akan menampilkan notifikasi kesalahan	Sistem berhasil menampilkan kesalahan	Berhasil
7	Confirm data	Data yang dikonfirmasi belum ada didatabase atau baru satu kali input data	Sistem akan menampilkan notifikasi bahwa data berhasil di daftar dan sistem akan mengembalikan ke halaman formulir	Sistem berhasil menampilkan notifikasi bahwa data berhasil di daftar dan menuju ke halaman formulir	Berhasil
8	Confirm data	Data yang dikonfirmasi adalah data yang sudah pernah diinput sebelumnya dan NIK sudah terdaftar di database.	Sistem akan menampilkan notifikasi bahwa nik sudah terdaftar dan data tidak bisa di input dua kali	Sistem berhasil menampilkan notifikasi bahwa nik sudah terdaftar dan data tidak bisa input dua kali	Berhasil
9	Search buku	Masukan keyword pencarian dan klik button search	Sistem akan menampilkan semua data yang sesuai dengan keyword	Sistem berhasil menampilkan data sesuai dengan keyword	Berhasil

No	Test case	Langkah uji	Hasil yang diharapkan	Hasil yang didapatkan	Keterangan
		Masukan keyword pencarian dengan data yang salah	Sistem tidak akan menampilkan data yang sesuai dengan keyword	Sistem berhasil untuk tidak menampilkan data keyword yang salah	Berhasil
10	Detail buku	Klik detail buku	Sistem akan menampilkan halaman detail buku yang dipilih	Sistem berhasil menampilkan detail buku yang dipilih	Berhasil
11	klik kategori buku	Pilih salah satu kategori buku	Sistem akan menampilkan semua data yang sesuai dengan kategori yang dipilih	Sistem berhasil menampilkan semua data yang sesuai dengan kategori yang dipilih	Berhasil
12	logout	Klik logout	Sistem akan menampilkan popup untuk konfirmasi logout dan jika berhasil akan menampilkan halaman login	Sistem berhasil menampilkan popup konfirmasi logout dan berhasil menuju halaman login	Berhasil

3.4.3 Pengujian TAM

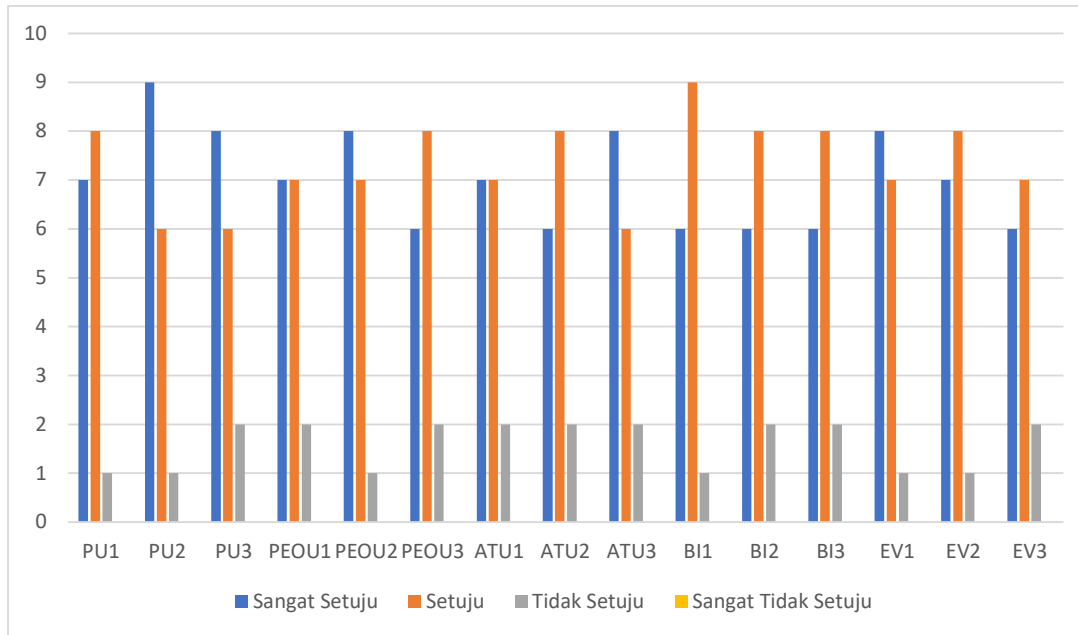
Pada pengujian ini bertujuan untuk mendapatkan respon yang berfokus pada 5 variabel yaitu, *Perceived Usefulness* (PU), *Perceived Ease of Use* (PEOU), *Attitude Toward Using* (ATU), *Behavioral Intention to Use* (BI), dan *External Variables* (EV). Berikut ini adalah rincian pertanyaan dari tiap variabel.

Tabel 3. 14 Kuesioner

Variable	Pertanyaan	Symbol
	Apakah Anda merasa bahwa menggunakan aplikasi pendaftaran anggota perpustakaan berbasis Android akan meningkatkan efisiensi proses pendaftaran?	PU1
Perceived Usefulness (PU)	Apakah Anda percaya bahwa aplikasi ini akan membuat proses pendaftaran anggota perpustakaan menjadi lebih cepat?	PU2
	Apakah Anda merasa bahwa aplikasi ini akan mengurangi kesalahan dalam pencatatan data anggota perpustakaan?	PU3
Perceived Ease of Use (PEOU)	Apakah Anda merasa bahwa aplikasi pendaftaran anggota perpustakaan berbasis Android ini mudah untuk dipelajari?	PEOU1

Variable	Pertanyaan	Symbol
	Apakah Anda merasa bahwa antarmuka aplikasi pendaftaran anggota perpustakaan ini intuitif dan mudah dimengerti?	PEOU2
	Apakah Anda merasa bahwa mengoperasikan aplikasi pendaftaran anggota perpustakaan ini tidak memerlukan banyak usaha?	PEOU3
	Apakah Anda merasa senang menggunakan aplikasi ini untuk mendaftar sebagai anggota perpustakaan?	ATU1
Attitude Toward Using (ATU)	Anda merasa puas menggunakan aplikasi pendaftaran anggota perpustakaan ini?	ATU2
	Anda memberikan nilai yang baik secara keseluruhan terhadap aplikasi pendaftaran anggota perpustakaan ini?	ATU3
	besar kemungkinan Anda akan menggunakan aplikasi pendaftaran anggota perpustakaan berbasis Android ini di masa mendatang?	BI1
Behavioral Intention to Use (BI)	besar keinginan Anda untuk merekomendasikan aplikasi pendaftaran anggota perpustakaan ini kepada orang lain?	BI2
	Anda akan memilih menggunakan aplikasi ini jika tersedia sebagai pilihan untuk pendaftaran atau layanan perpustakaan lainnya?	BI3
	infrastruktur teknologi yang tersedia (misalnya koneksi internet, smartphome) untuk mendukung penggunaan aplikasi pendaftaran anggota perpustakaan ini?	EV1
External Variables (EV)	Anda merasa bahwa lingkungan sosial (misalnya teman, keluarga, sesama pengguna perpustakaan) mendukung penggunaan aplikasi pendaftaran anggota perpustakaan berbasis Android?	EV2
	Anda percaya bahwa data pribadi Anda aman saat menggunakan aplikasi pendaftaran anggota perpustakaan ini?	EV3

Berdasarkan pertanyaan-pertanyaan di atas, berikut adalah ringkasan hasil jawaban responden mengenai kinerja sistem:



Gambar 3.54 Grafik Hasil Kuesioner

Berdasarkan hasil kuesioner dari 16 responden, dapat disimpulkan dengan menggunakan skala Likert bahwa responden memilih dari empat opsi jawaban dengan bobot sebagai berikut: Sangat Setuju (ST) = 4, Setuju (S) = 3, Tidak Setuju (TS) = 2, dan Sangat Tidak Setuju (STS) = 1.

Tabel 3.15 Hasil Nilai Persentase Tiap Pertanyaan Kuesioner

no	Pertanyaan	Frekuensi Jawaban				Jumlah Skor				Total Skor
		SS	S	TS	STS	SS	S	TS	STS	
1	PU1	7	8	1	0	44	57	0	0	101
2	PU2	9	6	1	0	48	51	2	0	101
3	PU3	8	6	2	0	36	57	4	0	97
4	PEOU1	7	7	2	0	32	60	4	0	96
5	PEOU2	8	7	1	0	40	54	4	0	98
6	PEOU3	6	8	2	0	36	57	4	0	97
7	ATU1	7	7	2	0	44	51	4	0	99
8	ATU2	6	8	2	0	44	54	2	0	100

no	Pertanyaan	Frekuensi Jawaban				Jumlah Skor				Total Skor
		SS	S	TS	STS	SS	S	TS	STS	
9	ATU3	8	6	2	0	36	57	4	0	97
10	BI1	6	9	1	0	52	48	2	0	102
11	BI2	6	8	2	0	36	57	4	0	97
12	BI3	6	8	2	0	40	54	4	0	98
13	EV1	8	7	1	0	40	57	2	0	99
14	EV2	7	8	1	0	40	57	2	0	99
15	EV3	6	7	2	0	40	51	6	0	97
Total Akhir Skor										798
Total Skor Tertinggi Tertinggi x Jumlah Responden x Jumlah Soal)										(Skor Skala 960
Persentase Rata-Rata Skor / Total Skor Tertinggi x 100)										(Total Akhrit 83%

Dari hasil persentase tabel 3.15 sistem pendaftaran anggota perpustakaan memiliki nilai persentase rata-rata sebesar 83%, yang berarti bahwa sebagian besar responden setuju atau sangat setuju atas implementasi sistem pendaftaran anggota perpustakaan berbasis android.

3.5 Pemeliharaan (*Maintenance*)

Tahap pemeliharaan biasanya meliputi perbaikan bugs dan peningkatan sistem untuk memastikan kinerja optimal dan relevansi. Namun, dalam konteks skripsi ini, aplikasi pendaftaran anggota perpustakaan dikembangkan hanya untuk tujuan akademis dan penelitian. Tidak direncanakan untuk implementasi dalam lingkungan produksi nyata atau penggunaan jangka panjang. Oleh karena itu, tidak ada rencana pemeliharaan yang disiapkan setelah periode pengujian dan evaluasi dalam rangka skripsi.

BAB IV PENUTUP

4.1 Kesimpulan

Berdasarkan hasil penelitian dan pengembangan aplikasi pendaftaran anggota perpustakaan berbasis Android menggunakan model waterfall pada Dinas Perpustakaan dan Kearsipan Kota Samarinda, dapat ditarik beberapa kesimpulan. (i) Aplikasi ini berhasil mengatasi permasalahan dalam proses pendaftaran anggota perpustakaan yang sebelumnya dilakukan secara manual. Dengan adanya aplikasi ini, calon anggota dapat melakukan pendaftaran secara online tanpa harus mengunjungi perpustakaan berulang kali, menghemat waktu proses pendaftaran. (ii) Fitur pencarian buku yang diimplementasikan dalam aplikasi memudahkan anggota perpustakaan untuk mencari ketersediaan buku tanpa harus datang langsung ke perpustakaan, mengurangi kemungkinan kunjungan yang sia-sia karena buku yang dicari tidak tersedia. Hasil kuesioner menunjukkan tingkat kepuasan yang tinggi, dengan nilai persentase rata-rata sebesar 83%, yang berarti bahwa sebagian besar responden setuju atau sangat setuju atas implementasi sistem pendaftaran anggota perpustakaan berbasis android.

4.2 Saran

Dalam penelitian ini, peneliti menyadari bahwa sistem tersebut belum sempurna. Untuk penelitian selanjutnya, diharapkan dapat mengembangkan aplikasi ini untuk mendukung penggunaan di perangkat iOS, fitur tambahan seperti peminjaman buku online dan notifikasi batas waktu pengembalian buku.

DAFTAR RUJUKAN

- Alfarisi, I. A., Priandika, A. T., & Puspaningrum, A. S. (2023). Penerapan Framework Laravel Pada Sistem Pelayanan Kesehatan (Studi Kasus: Klinik Berkah Medical Center). *Jurnal Ilmiah Computer Science*, 2(1), 1–9. <https://doi.org/10.58602/jics.v2i1.11>
- Anis, Y., Mukti, A. B., & Rosyid, A. N. (2023). KLIK: Kajian Ilmiah Informatika dan Komputer Penerapan Model Waterfall Dalam Pengembangan Sistem Informasi Aset Destinasi Wisata Berbasis Website. *Media Online*, 4(2), 1134–1142. <https://doi.org/10.30865/klik.v4i2.1287>
- Azrial, M. F. A., & Fadillah, N. (2020). Sistem Informasi Pengangkutan Pupuk Menggunakan Metode Waterfall (Studi Kasus PT. Pupuk Iskandar Muda, Aceh Utara). *J-ICOM - Jurnal Informatika Dan Teknologi Komputer*, 1(2), 75–81. <https://doi.org/10.33059/j-icom.v1i2.2897>
- Badrul, M. (2021). Penerapan Metode waterfall untuk Perancangan Sistem Informasi Inventory Pada Toko Keramik Bintang Terang. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 8(2), 57–52. <https://doi.org/10.30656/prosisko.v8i2.3852>
- Bimantoro, A., Pramesti, Wanda, A., Bakti, Satria, W., Samudra, M. A., & Amrozi, Y. (2021). Paradoks Etika Pemanfaatan Teknologi Informasi di Era 5.0. *Jurnal Teknologi Informasi*, 7(1), 58–68. <https://doi.org/10.52643/jti.v7i1.1425>
- Chandra, C., Sanjaya, D., Narabel, J., Vilano, N., & Budi, S. (2019). Analisis Estimasi Waktu Antrian dengan menggunakan Markov Chain dan Algoritma Pagerank. *Jurnal Teknik Informatika Dan Sistem Informasi*, 5(3), 406–414.
- Fadilah, N., Ikhwan, A., & Alda, M. (2023). Pengembangan Sistem Informasi Perpustakaan Pada Dinas Perpustakaan dan Kearsipan Kota Medan Berbasis Android. *J-SISKO TECH (Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD)*, 6(2), 298. <https://doi.org/10.53513/jsk.v6i2.8228>
- Handayani, D., & Salam, M. (2023). Aplikasi Sistem Informasi Simpan Pinjam Koperasi Berbasis Website Menggunakan Metode Waterfall. *Media Online*, 3(5), 425–434. <https://djournals.com/klik>
- Hanif, I. F., & Sinambela, G. M. (2020). Pembuatan Aplikasi E-Tatib Berbasis Android Menggunakan Bahasa Pemrograman Dart. *Jurnal Teknologi Dan Terapan Bisnis (JTTB)*, 3(2), 23–29.
- Hendrawan, S., Manuputty, A., & Haryanto, B. (2020). Design of Information Systems for Research Permit Application with Agile Method and Website Based Laravel Framework. *Journal of Information Systems and Informatics*, 2(1), 60–78. <https://doi.org/10.33557/journalisi.v2i1.45>
- Kahfi, A. H., Hasan, M., & Fazriansyah, A. (2023). Perancangan Program Pembayaran Administrasi Sekolah Berbasis Website Menggunakan Metode Waterfall. *Analisis Tingkat Kepuasan Pengguna Sistem Informasi Perpustakaan Menggunakan Pieces Framework*, 3(6), 1063–1069. <https://doi.org/10.30865/klik.v3i6.801>
- Mahdalena, D., Sari, V. N., Qurniati, N., & Prahasti, P. (2023). Perancangan Sistem Informasi Penjualan Pada Kedai Kopi Luwak Bengkulu Menggunakan Bahasa Pemrograman PHP dan Database MYSQL. *Digital Transformation Technology*, 3(2), 609–617. <https://doi.org/10.47709/digitech.v3i2.3094>

- Nugraha, F., Restendi, D., & Triyanto, A. (2020). Pengembangan Sistem Pelatihan Jarak Jauh Berbasis Moodle di Balai Diklat Keagamaan Bandung. *Andragogi: Jurnal Diklat Teknis Pendidikan Dan Keagamaan*, 8(2), 540–552. <https://doi.org/10.36052/andragogi.v8i2.166>
- Ridwansyah, R. (2023). Aplikasi Bimbingan Akademik berbasis Android di Jurusan Pendidikan Teknik Elektronika FT-UNM. *Jurnal Teknologi Informasi*, 9(1), 30–37. <https://doi.org/10.52643/jti.v9i1.3072>
- Rina Noviana. (2022). Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan Php Dan Mysql. *Jurnal Teknik Dan Science*, 1(2), 112–124. <https://doi.org/10.56127/jts.v1i2.128>
- Santoso, S., Surjawan, D. J., & Handoyo, E. D. (2020). Pengembangan Sistem Informasi Tukar Barang Untuk Pemanfaatan Barang Tidak Terpakai dengan Flutter Framework. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(3), 589–598. <https://doi.org/10.28932/jutisi.v6i3.3071>
- Siregar, S. R., & Pristiwanto, P. (2022). Penerapan Metode Waterfall Dalam Pengembangan Sistem Informasi Masjid. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, 3(1), 26–32. <https://djournals.com/klik>
- Supiana, N. (2022). Pengembangan Aplikasi Geolocation Untuk Monitoring Lokasi Mahasiswa Selama Pandemi Berbasis Android Menggunakan Metode Waterfall (Studi Kasus : Stmik Insan Pembangunan. *Jurnal Khatulistiwa Informatika*, 10(1), 74–80. <https://doi.org/10.31294/jki.v10i1.11741>
- Yosepha Pusparisa. (2020). *Pengguna Smartphone diperkirakan Mencapai 89% Populasi pada 2025*. Databoks.Katadata.Co.Id. <https://databoks.katadata.co.id/datapublish/2020/09/15/pengguna-smartphone-diperkirakan-mencapai-89-populasi-pada-2025>
- Zulfa, I., & Wanda, R. (2023). Klik: kajian ilmiah informatika dan komputer rancangan sistem informasi akademik berbasis website menggunakan php dan mysql. *Klik: Kajian Ilmiah Informatika Dan Komputer*, 3(4), 393–399. <https://djournals.com/klik/article/view/617>

RIWAYAT HIDUP



Muhammad Ramadhani Saputra atau biasanya dipanggil Dani, lahir di Samarinda pada tanggal 13 November 2002 dari pasangan Bapak Ramly Djafar dan Ibu Malarifjana penulis merupakan anak 2 dari 3 bersaudara. Penulis Berkebangsaan Indonesia dan beragama Islam. Penulis berasal dari Kota Samarinda dan bertempat tinggal di Jalan Marsda A Saleh Kelurahan Sidomulyo. Penulis menempuh pendidikan dari SD Negeri 015 Kota Samarinda (2008-2014), selanjutnya menempuh pendidikan pada sekolah pertama di SMP Negeri 34 Kota Samarinda (2014-2017), kemudian untuk pendidikan sekolah menengah atas penulis menyelesaikannya di SMK Negeri 9 Samarinda (2017-2020), dan untuk pendidikan perguruan tinggi penulis tempuh di Universitas Muhammadiyah Kalimantan Timur (UMKT) dari tahun 2020 sampai sekarang.

LAMPIRAN

Lampiran 1 Surat Izin Penelitian



UMKTI
Program Studi
Teknik Informatika
Fakultas Sains dan Teknologi

Telp. 0541-748511 Fax. 0541-766832

Website <http://informatika.umkt.ac.id>

email: informatika@umkt.ac.id



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 026-001/FST.1/A.7/C/2024
Lampiran : -
Perihal : Permohonan Izin Penelitian

Kepada Yth.
Dinas Perpustakaan dan Kearsipan Kota Samarinda
di -
Tempat

Assalamu'alaikum Warrahmatullahi Wabarrakatuh

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Aamiin.

Sehubungan dengan penyusunan Tugas Akhir/Skripsi Mahasiswa Program Studi Teknik Informatika Tahun Akademik 2023/2024, maka dengan ini kami mengajukan permohonan untuk dapat menerima mahasiswa dari Program Studi Teknik Informatika untuk melakukan penelitian di Dinas Perpustakaan dan Kearsipan Kota Samarinda, dengan nama mahasiswa:

Nama : Muhammad Ramadhani Saputra
NIM : 2011102441087
Program Studi : Teknik Informatika

Demikian permohonan ini dibuat. Atas perhatian dan kerjasamanya kami mengucapkan terima kasih.

Wassalamu'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 11 Sya'ban 1445 H
21 Februari 2024 M



Dosen Pembimbing : *[Signature]*

[Signature]
ansyah, S.Kom., M.TI
IDN. 1118019203

Kampus 1 : Jl. Ir. H. Juanda, No.15, Samarinda
Kampus 2 : Jl. Pelita, Pesona Mahakam, Samarinda

Lampiran 2 Surat Balasan Izin Penelitian



PEMERINTAH KOTA SAMARINDA
DINAS PERPUSTAKAAN DAN KEARSIPAN

Jl. Kesuma Bangsa 5, Samarinda, Kalimantan Timur
website: <https://perpustakaankearsipan.samarindakota.go.id> --- e-mail: dispursip.smr@gmail.com

Samarinda, 23 Juli 2024

Nomor : 400.14.5.4/0895/100.16
Sifat : Biasa
Lampiran : -
Hal : Keterangan Telah Melaksanakan Penelitian

Yth. Ketua Program Studi S-1 Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Muhammadiyah Kalimantan Timur
di
Samarinda

Sebagai tindak lanjut surat dari Universitas Muhammadiyah Kalimantan Timur, nomor: 026-001/FST.1/A.7/C/2024, tanggal 21 Februari 2024 tentang Permohonan Izin Penelitian, berkenaan dengan hal tersebut pada prinsipnya kami menerima Mahasiswa di bawah ini untuk melakukan penelitian dan pengambilan data sebagai bahan penyelesaian skripsi.

nama : Muhammad Ramadhani Saputra
NIM : 2011102441087
program studi : Teknik Informatika.

Demikian disampaikan, atas kerja sama yang baik diucapkan terima kasih.













Kepala Dinas Perpustakaan
dan Kearsipan Kota Samarinda,

Erham Yusuf, M.Pd.
Pembina Utama Muda IV/c
NIP 196808181998021004


Lampiran 3 Kartu Kendali Bimbingan

KARTU KENDALI BIMBINGAN LAPORAN KARYA ILMIAH

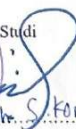
Nama : Muhammad Ramadhani Saputra
 NIM : 2011102441087
 Nama Dosen Pembimbing : Abdul Rahim, S.Kom., M.Cs
 Judul Penelitian : Pengembangan Aplikasi Perpustakaan Berbasis Android Menggunakan Model Waterfall Pada Dinas Perpustakaan Kota Samarinda

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1.	22/02/2024	Bimbingan Judul Skripsi	
2.	03/03/2024	Bimbingan Kelompok PTA	
3.	14/03/2024	Bimbingan canvas	
4.	24/04/2024	Bimbingan BAB 1	
5.	26/04/2024	Bimbingan bab 2	
6.	28/04/2024	Bimbingan revisi bab 1 dan bab 2	
7.	16/05/2024	Bimbingan untuk upload proposal ke Sumpel	
8.	28/05/2024	Bimbingan revisi review desk	
9.	7/06/2024	Bimbingan bab 3	
10.	17/06/2024	Bimbingan revisi bab 3	
11.			
12.			
13.			
14.			

Dosen Pembimbing


 (Abdul Rahim, S.Kom., M.Cs)




 (Abdul Rahim, S.Kom., M.TI)

Lampiran 4 Jadwal Penelitian

No	Jenis Penelitian	Bulan/2024					
		Feb	Mar	Apr	Mei	Jun	Jul
Tahap Pra Penelitian							
1	Menentukan Judul Penelitian						
2	Menentukan Permasalahan						
3	Mencari Studi Literatur						
4	Menyusun Metode Penelitian						
5	Menentukan Studi Kasus						
6	Menyusun Proposal Penelitian						
7	Review Desk Simpel						
Tahap Penelitian							
1	Analisis Kebutuhan (Requirement Analysis)						
2	Perancangan (Desain)						
3	Implementasi (Implementation)						
4	Pengujian (Testing)						
5	Pemeliharaan (Maintenance)						
Tahap Akhir Penelitian							
1	Penyusunan Laporan						
2	Seminar Hasil						

Lampiran 5 Code main.dart

```
import 'package:aplikasi_daftar_angota_perpus/screens/form/form.dart';
import 'package:flutter/material.dart';
import 'splashscreens.dart';
import 'screens/login/login.dart';
import 'package:flutter/services.dart';

void main() {
  WidgetsFlutterBinding.ensureInitialized();
  SystemChrome.setPreferredOrientations([
    DeviceOrientation.portraitUp,
  ]);
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      // theme: ThemeData(primaryColor: Colors.green),
      debugShowCheckedModeBanner: false,
      home: Loading(),
      routes: {
        '/form': (context) => MyFormPage(),
        '/login': (context) => LoginPage(),
        '/loading': (context) => Loading(),
      },
    );
  }
}
```

Lampiran 6 splashscreens.dart

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'screens/login/login.dart';
import 'package:percent_indicator/percent_indicator.dart';

class Loading extends StatefulWidget {
  // const Loading({Key? key}) : super(key: key);

  @override
  _LoadingState createState() => _LoadingState();
}
```

```

class _LoadingState extends State<Loading> {
  @override
  void initState() {
    super.initState();
    Timer(const Duration(seconds: 1), () {
      Navigator.of(context)
        .pushReplacement(MaterialPageRoute(builder: (_) => LoginPage()));
    });
  }

  @override
  Widget build(BuildContext context) {
    final MediaQuerywidth = MediaQuery.of(context).size.width;
    return Scaffold(
      backgroundColor:
        Colors.transparent, // Make the scaffold background transparent
      body: Container(
        width: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.white, Color(0xffff012ac0)],
            begin: Alignment.topRight,
            end: Alignment.bottomLeft),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
              ),
            Image.asset(
              'assets/splash_screens/registermember.png', // Replace with your image asset path
              height: 100, // Adjust the height of the image
            ),
          ],
        ),
      ),
    );
  }
}

```


Lampiran 7 Code login.dart

```
import 'package:aplikasi_daftar_angota_perpus/bottomnavbar/bottomnavbar.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:quickalert/quickalert.dart';
import 'package:shared_preferences/shared_preferences.dart';
import './register.dart';
import 'package:http/http.dart' as http;
// import '../Services/globals.dart';
import '../Services/snackbar.dart';
import 'package:aplikasi_daftar_angota_perpus/Services/auth_services.dart';
import 'dart:convert';
// import 'package:shared_preferences/shared_preferences.dart';

class LoginPage extends StatefulWidget {
  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  String _username = "";
  String _password = "";

  loading() {
    QuickAlert.show(
      context: context,
      type: QuickAlertType.loading,
      text: 'Loading...',
    );
  }

  error(String text) {
    QuickAlert.show(
      context: context,
      type: QuickAlertType.error,
      title: 'Oops...',
      titleColor: Colors.red,
      text: text,
    );
  }

  warning(String text) {
    QuickAlert.show(
      context: context,
      type: QuickAlertType.warning,
      titleColor: Colors.yellow.shade900,
      text: text,
    );
  }
}
```

```

}

loginPressed(BuildContext context) async {
  if (_username.isNotEmpty && _password.isNotEmpty) {
    // Menampilkan loading
    // loading();
    showDialog(
      context: context,
      barrierDismissible: false,
      builder: (BuildContext context) {
        return Center(
          child: SizedBox(
            width: 30,
            height: 30,
            child: CircularProgressIndicator(
              // color: Color(0xffff012ac0),
              color: Color(0xffff012ac0),
            ),
          ),
        );
      },
    );
    await Future.delayed(Duration(milliseconds: 500));
    http.Response response = await AuthServices.login(_username, _password);

    // Menutup dialog loading
    Navigator.pop(context);

    if (response.statusCode == 200) {
      Map<String, dynamic>? responseMap = jsonDecode(response.body);

      if (responseMap != null && responseMap.containsKey('user')) {
        SharedPreferences preferences = await SharedPreferences.getInstance();
        preferences.setInt('id', responseMap['user']['id']);
        preferences.setString('username', responseMap['user']['username']);
        preferences.setString('password', responseMap['user']['password']);
        preferences.setString('firstname', responseMap['user']['firstname']);
        preferences.setString('lastname', responseMap['user']['lastname']);
        preferences.setString('token', responseMap['token']);

        // Navigasi ke halaman Home
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            builder: (BuildContext context) => BottomNavbar(),
          ),
        );
      } else {
        errorSnackBar(context,

```

```

        'Respons dari backend tidak sesuai atau tidak mengandung objek user. ');
    }
  } else {
    error("Username atau Password Salah !!");
    // errorSnackBar(context, 'Username atau Password Salah !!');
  }
  } else {
    warning('Username dan Password tidak boleh kosong');
    // errorSnackBar(context, 'Masukkan semua kolom yang diperlukan');
  }
}
}

```

Lampiran 8 Code Register.dart

```

import 'dart:convert';
import 'package:aplikasi_daftar_angota_perpus/Services/auth_services.dart';
import 'package:aplikasi_daftar_angota_perpus/screens/login/login.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:http/http.dart' as http;
// import '../Services/globals.dart';
import '../Services/snackbar.dart';

class RegisterPage extends StatefulWidget {
  @override
  State<RegisterPage> createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  String _firstname = "";
  String _lastname = "";
  String _username = "";
  String _password = "";
  bool _isLoading = false;

  createAccountPressed(BuildContext context) async {
    bool nameValid = RegExp(r'^[a-zA-Z0-9]{3,16}$').hasMatch(_username);
    if (nameValid) {
      showDialog(
        context: context,
        barrierDismissible: false,
        builder: (BuildContext context) {
          return Center(
            child: SizedBox(
              width: 30,
              height: 30,
              child: CircularProgressIndicator(

```

```

        color: Color(0xffff012ac0),
      ),
    ),
  );
},
);

await Future.delayed(Duration(milliseconds: 1500));
http.Response response = await AuthServices.register(
  _username, _password, _firstname, _lastname);
Navigator.pop(context);
Map<String, dynamic> responseMap =
  jsonDecode(response.body) as Map<String, dynamic>;
if (response.statusCode == 200) {
  // Tampilkan snackbar setelah registrasi berhasil
  successSnackBar(context, 'Register Berhasil');
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (BuildContext context) => LoginPage(),
    ),
  );
} else {
  String errorMessage = extractErrorMessage(responseMap);
  errorSnackBar(context, errorMessage);
}
} else {
  errorSnackBar(context, 'Kolom Tidak Boleh Kosong');
}
}
}

```

Lampiran 9 globals.dart

```

String baseUrl = "http://10.20.2.241:8000/api/";
String baseUrlgambar = "http://10.20.2.241:8000/";
Map<String, String> headers = {
  "Content-Type": "application/json",
  'Accept': 'application/json',
};

```

Lampiran 10 code form.dart

```

import 'package:aplikasi_daftar_angota_perpus/Services/snackbar.dart';
import 'package:aplikasi_daftar_angota_perpus/loading/loadingform/loadingform.dart';
import 'package:aplikasi_daftar_angota_perpus/screens/form/controller/confirmdata.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgetform.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgettext.dart';

```

```

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:shared_preferences/shared_preferences.dart';

// import 'package:http/http.dart' as http;

class MyFormPage extends StatefulWidget {
  @override
  _MyFormPageState createState() => _MyFormPageState();
}

final GlobalKey<ScaffoldMessengerState> scaffoldMessengerKey =
  GlobalKey<ScaffoldMessengerState>();

int userId = 0;

class _MyFormPageState extends State<MyFormPage> {
  void initState() {
    super.initState();
    // Panggil metode untuk mengambil data dari SharedPreferences saat widget diinisialisasi
    getLoginData();
    _delayedPageLoad();
  }

  bool _isLoading = false;
  late String userName;
  late String userToken;

  final formKey = GlobalKey<FormState>();
  final TextEditingController _namaController = TextEditingController();
  final TextEditingController _nikController = TextEditingController();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _alamatKtp = TextEditingController();
  final TextEditingController _alamatSekarang = TextEditingController();
  final TextEditingController _namaSekolah = TextEditingController();
  final TextEditingController _nomorHp = TextEditingController();
  final TextEditingController _tglLahir = TextEditingController();
  String _listProfesi = 'Siswa';
  String? _gender;
  String _pendidikanTerakhir = 'SD';
  List<String> _schoolTypes = [
    'PAUD',
    'TK',
    'KB',
    'SD',
    'MI',
    'SMP',
    'MTS',
  ];

```

```

'SMA',
'SMK',
'MA',
'D3',
'S1',
'S2',
'S3',
];
List<String> _profesi = [
'Siswa',
'Mahasiswa',
'Guru',
'Dosen',
'Peneliti',
'PNS Daerah (Pusat)',
'Karyawan',
'Ti',
'Umum',
'Ibu Rumah Tangga',
];
Future<void> getLoginData() async {
  SharedPreferences preferences = await SharedPreferences.getInstance();
  // Ambil nilai ID, name, dan token dari SharedPreferences
  userId = preferences.getInt('id') ?? 0;
  userName = preferences.getString('name') ?? 'Nama Tidak Ditemukan';
  userToken = preferences.getString('token') ?? "";
  // Panggil setState untuk memperbarui tampilan
  if (mounted) {
    setState(() {});
  }
}

bool isPageLoaded = false;

void _delayedPageLoad() {
  Future.delayed(Duration(milliseconds: 1000), () {
    setState(() {
      isPageLoaded = true; // Setelah 1 detik, halaman dianggap sudah ter-load
    });
  });
}
}

```

Lampiran 11 profil.dart

```
import 'dart:io';
import 'package:aplikasi_daftar_angota_perpus/bottomnavbar/dialogexit.dart';
import 'package:aplikasi_daftar_angota_perpus/loading/loadingprofil/loadingprofil.dart';
import 'package:aplikasi_daftar_angota_perpus/screens/home.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgetprofil.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgettext.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import '../Services/auth_services.dart';
import 'dart:convert';
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:async';

class Profil extends StatefulWidget {
  const Profil({Key? key}) : super(key: key);

  @override
  State<Profil> createState() => _ProfilState();
}

late int userId;
late Map<String, dynamic> anggotaData =
  {}; // Menggunakan Map untuk satu anggota

class _ProfilState extends State<Profil> {
  void initState() {
    super.initState();
    // Panggil metode untuk mengambil data dari SharedPreferences saat widget diinisialisasi
    getUserData();
    _delayedPageLoad();
    // Panggil fetchData untuk mengambil data berdasarkan userId
    // fetchDataByIdFlutterUser(1);
  }

  Future<void> getUserData() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
    // Ambil nilai name, token, dan id dari SharedPreferences
    // userName = preferences.getString('username') ?? 'Nama Tidak Ditemukan';
    userId = preferences.getInt('id') ??
      -1; // Ganti -1 dengan nilai default yang sesuai
    // Panggil setState untuk memperbarui tampilan
    setState(() {});

    // Panggil fetchData dengan userId yang sudah didapat
    fetchDataByIdFlutterUser(userId);
  }
}
```

```

Future<void> fetchDataByIdFlutterUser(int idFlutterUser) async {
  try {
    final response = await API().getDataByIdFlutterUser(idFlutterUser);

    if (response.statusCode == 200) {
      final decodedData = json.decode(response.body);
      if (decodedData is Map<String, dynamic>) {
        // Tidak perlu mengonversi anggota menjadi List
        setState(() {
          anggotaData = decodedData['data'];
          // isLoading = false;
        });
      } else {
        throw Exception('Format respons tidak sesuai');
      }
    } else {
      throw Exception('Gagal mengambil data dari server');
    }
  } catch (error) {
    print('Error: $error');
  }
}

```

Lampiran 12 screenbuku.dart

```

import 'dart:io';
import 'package:aplikasi_daftar_angota_perpus/bottomnavbar/dialogexit.dart';
import 'package:aplikasi_daftar_angota_perpus/loading/loadingprofil/loadingprofil.dart';
import 'package:aplikasi_daftar_angota_perpus/screens/home.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgetprofil.dart';
import 'package:aplikasi_daftar_angota_perpus/widget/widgettext.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import '../Services/auth_services.dart';
import 'dart:convert';
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:async';

class Profil extends StatefulWidget {
  const Profil({Key? key}) : super(key: key);

  @override
  State<Profil> createState() => _ProfilState();
}

late int userId;
late Map<String, dynamic> anggotaData =

```



```

}); // Menggunakan Map untuk satu anggota

class _ProfilState extends State<Profil> {
  void initState() {
    super.initState();
    // Panggil metode untuk mengambil data dari SharedPreferences saat widget diinisialisasi
    getUserData();
    _delayedPageLoad();
    // Panggil fetchData untuk mengambil data berdasarkan userId
    // fetchDataByIdFlutterUser(1);
  }

  Future<void> getUserData() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
    // Ambil nilai name, token, dan id dari SharedPreferences
    // userName = preferences.getString('username') ?? 'Nama Tidak Ditemukan';
    userId = preferences.getInt('id') ??
      -1; // Ganti -1 dengan nilai default yang sesuai
    // Panggil setState untuk memperbarui tampilan
    setState(() {});

    // Panggil fetchData dengan userId yang sudah didapat
    fetchDataByIdFlutterUser(userId);
  }

  Future<void> fetchDataByIdFlutterUser(int idFlutterUser) async {
    try {
      final response = await API().getDataByIdFlutterUser(idFlutterUser);

      if (response.statusCode == 200) {
        final decodedData = json.decode(response.body);
        if (decodedData is Map<String, dynamic>) {
          // Tidak perlu mengonversi anggota menjadi List
          setState(() {
            anggotaData = decodedData['data'];
            // isLoading = false;
          });
        } else {
          throw Exception('Format respons tidak sesuai');
        }
      } else {
        throw Exception('Gagal mengambil data dari server');
      }
    } catch (error) {
      print('Error: $error');
    }
  }
}

```

Lampiran 13 code Login.dart

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
  <link rel="stylesheet" href="{{asset('csslogin/loginess.css')}}">
</head>
<body>
<div class="container" id="container">
  <div class="form-container sign-up-container">
    <form action="#">
      <h1>Create Account</h1>
      <div class="social-container">
        <a href="#" class="social"><i class="fab fa-facebook-f"></i></a>
        <a href="#" class="social"><i class="fab fa-google-plus-g"></i></a>
        <a href="#" class="social"><i class="fab fa-linkedin-in"></i></a>
      </div>
      <span>or use your email for registration</span>
      <input type="text" placeholder="Name" />
      <input type="email" placeholder="Email" />
      <input type="password" placeholder="Password" />
      <button>Sign Up</button>
    </form>
  </div>
  <div class="form-container sign-in-container">
    <form action="/loginproses" method="post">
      @csrf
      <h1>Sign in</h1>
      <br>
      <span>or use your account</span>
      @if($errors->any())
        <div class="alert alert-danger">
          <ul>
            @foreach ($errors->all() as $error)
              <li>{{ $error }}</li>
            @endforeach
          </ul>
        </div>
      @endif
      <input type="text" name="name" placeholder="Username" />
      <input type="password" name="password" placeholder="Password" />
      <br>
      <button>Sign In</button>
    </form>
  </div>
  <div class="overlay-container">
    <div class="overlay">
```

```

    <div class="overlay-panel overlay-left">
      <h1>Welcome Back!</h1>
      <p>To keep connected with us please login with your personal info</p>
      <button class="ghost" id="signIn">Sign In</button>
    </div>
    <div class="overlay-panel overlay-right">
      <h1>Selamat Datang</h1>
      <p>Silakan masukkan username dan password Anda untuk mengakses sistem ini.</p>
    </div>
  </div>
</div>
<script src="{{asset('csslogin/loginjs.js')}}"></script>
</body>
</html>

```

Lampiran 14 code dashboard.dart

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">
  <title>Dashboard</title>
  <!-- Custom fonts for this template-->
  <link href="{{asset('sbadmin/all.min.css')}}" rel="stylesheet" type="text/css">
  <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,
800i,900,900i" rel="stylesheet">
  <!-- Custom styles for this template-->
  <link href="{{asset('sbadmin/sb-admin-2.min.css')}}" rel="stylesheet">
  <link href="{{asset('sbadmin/dataTables.bootstrap4.min.css')}}" rel="stylesheet">
</head>
<body id="page-top">
  <!-- Page Wrapper -->
  <div id="wrapper">
    <!-- Sidebar -->
    <ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">
      <!-- Sidebar - Brand -->
      <a class="sidebar-brand d-flex align-items-center justify-content-center" href="">
        <div class="sidebar-brand-icon rotate-n-15">
          <i class="">MY</i>
        </div>
        <div class="sidebar-brand-text mx-2">Dashboard</div>
      </a>

```

```

<!-- Divider -->
<hr class="sidebar-divider my-0">
<!-- Nav Item - Dashboard -->
<li class="nav-item active">
  <a class="nav-link" href="/">
    <i class="fas fa-fw fa-tachometer-alt"></i>
    <span>Dashboard</span></a>
</li>
<hr class="sidebar-divider">
<li class="nav-item">
  <a class="nav-link" href="/daftaranggota">
    <i class="fas fa-fw fa-table"></i>
    <span><strong>Anggota Perpustakaan</strong></span>
  </a>
</li>

<li class="nav-item">
  <a class="nav-link" href="/daftarregister">
    <i class="fas fa-fw fa-table"></i>
    <span><strong>Data Calon Anggota</strong></span></a>
</li>

<li class="nav-item">
  <a class="nav-link" href="/daftarbuku">
    <i class="fas fa-fw fa-table"></i>
    <span><strong>Buku</strong></span></a>
</li>

<hr class="sidebar-divider">

```

Lampiran 15 code anggota perpustakaan

```

@extends('layout.dashboard')
@extends('tambahdata')
@extends('editdata')
@extends('detaildata')
@section('content')
<div class="container-fluid">
  <div class="card shadow mb-4">
    <div class="card-header py-3">
      <h3 class="m-0 font-weight-bold text-primary">
        <center>Data Anggota</center>
      </h3>
    </div>
    <div class="card-body">
      <!-- <a href="#" type="button" class="btn btn-warning mb-2" data-toggle="modal" data-
target="#tambahModal">Tambah +</a -->
      <div class="table-responsive">
        <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">

```

```

<thead>
  <tr>
    <th>no</th>
    <th>Nik</th>
    <th>Nama</th>
    <th>Jenis Kelamin</th>
    <th>No Telpon</th>
    <th>Dibuat</th>
    <th>Aksi</th>
  </tr>
</thead>
<tbody>
  @foreach ($data as $row)
  <tr>
    <th scope="row">{{ $loop->iteration }}</th>
    <td>{{ $row->nik }}</td>
    <td style="text-transform: uppercase;">{{ $row->nama }}</td>
    <td>{{ $row->jeniskelamin }}</td>
    <td>0{{ $row->notelpon }}</td>
    <td>{{ $row->created_at->diffForHumans() }}</td>
    <td>
      <!-- <a href="#" type="button" class="btn btn-info" data-toggle="modal" data-
target="#editModal-{{ $row->id }}">Edit</a> -->
      <a href="#" type="button" class="btn btn-success" data-toggle="modal" data-
target="#detailModal-{{ $row->id }}">Detail</a>
      <a href="#" type="button" class="btn btn-danger delete" data-id="{{ $row->id }}"
data-nama="{{ $row->nama }}">Delete</a>
      <!-- <a href="/delete/{{ $row -> id }}" type="button" class="btn btn-
danger">Delete</a> -->
    </td>
  </tr>
  @endforeach
</tbody>
</table>
</div>
<script src="sbadmin/bootstrap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
<script src="sbadmin/sweetalert.min.js"></script>
<!-- cdn jquery -->
<script src="sbadmin/jquery-3.7.1.slim.min.js" integrity="sha256-
kmHvs0B+OpCW5GVHUNjv9rOmY0IvSIRcf7zGUDTDQM8=" crossorigin="anonymous"></script>
<!-- cdn toastr -->
<script src="sbadmin/toastr.min.js" integrity="sha512-
VEd+mq25CkR676O+pLBnDW09R7VQX9Mdiiij052gVCp5yVH3jGtH70Ho/UUv4mJDsEdTvqRCFZg0N
KGiojGnUCw=" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script>
  $('<span>'.delete').click(function() {
    var anggotaid = $(this).attr('data-id')
  })
</script>

```

```

var nama = $(this).attr('data-nama')
swal({
  title: "Yakin ?",
  text: "Kamu akan menghapus data anggota dengan nama *" + nama + "**",
  icon: "warning",
  buttons: true,
  dangerMode: true,
})
.then((willDelete) => {
  if (willDelete) {
    window.location = "/delete/" + anggotaaid + ""
    swal("Data berhasil dihapus", {
      icon: "success",
    });
  } else {
    swal("Data tidak jadi dihapus");
  }
});
})
</script>
</div>
</div>
</div>
@endsection

```

Lampiran 16 code calon anggota

```

@extends('layout.dashboard')
@extends('tambahdata')

@extends('editdata')
@extends('detaildata')

@section('content')
<div class="container-fluid">
  <div class="card shadow mb-4">
    <div class="card-header py-3">
      <h3 class="m-0 font-weight-bold text-primary">
        <center> Data Calon Anggota</center>
      </h3>
    </div>
    <div class="card-body">
      <!-- <a href="#" type="button" class="btn btn-warning mb-2" data-toggle="modal" data-
target="#tambahModal">Tambah +</a> -->
    </div>
    <div class="table-responsive">
      <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
        <thead>

```

```

        <tr>
            <th>no</th>
            <th>username</th>
            <th>firstname</th>
            <th>lastname</th>
            <th>Dibuat</th>
            <th>Aksi</th>
        </tr>
    </thead>
    <tbody>
        @foreach ($data as $row)
        <tr>
            <th scope="row">{{ $loop->iteration }}</th>
            <td>{{ $row->username }}</td>
            <td>{{ $row->firstname }}</td>
            <td>{{ $row->lastname }}</td>
            <td>{{ $row->created_at->diffForHumans() }}</td>
            <td>
                <!-- <a href="#" type="button" class="btn btn-info" data-toggle="modal" data-
target="#editModal-{{ $row->id }}">Edit</a> -->
                <a href="#" type="button" class="btn btn-danger delete" data-id="{{ $row->id }}"
data-nama="{{ $row->username }}">Delete</a>
                <!-- <a href="/delete/{{ $row -> id }}" type="button" class="btn btn-
danger">Delete</a> -->
                {{-- <a href="#" type="button" class="btn btn-success" data-toggle="modal" data-
target="#detailModal-{{ $row->id }}">Detail</a> --}}
            </td>
        </tr>
        @endforeach
    </tbody>
</table>
</div>
<script src="sbadmin/bootstrap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
<script src="sbadmin/sweetalert.min.js"></script>
<!-- cdn jquery -->
<script src="sbadmin/jquery-3.7.1.slim.min.js" integrity="sha256-
kmHvs0B+OpCW5GVHUNjv9rOmY0IvSIRcf7zGUDTDQM8=" crossorigin="anonymous"></script>
<!-- cdn toastr -->
<script src="sbadmin/toastr.min.js" integrity="sha512-
VEd+nq25CkR676O+plBnDW09R7VQX9Mdii;052gVCp5yVH3jGtH70Ho/UUv4mJDsEdTvqRCFZg0N
KGiojGnUCw=" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
<script>
    $('<del>.delete</del>').click(function() {
        var anggotaid = $(this).attr('data-id')
        var nama = $(this).attr('data-nama')
        swal({
            title: "Yakin ?",

```

```

        text: "Kamu akan menghapus data anggota pegawai dengan nama *" + nama + "**",
        icon: "warning",
        buttons: true,
        dangerMode: true,
    })
    .then((willDelete) => {
        if (willDelete) {
            window.location = "/deleteregister/" + anggotaaid + ""
            swal("Data berhasil dihapus", {
                icon: "success",
            });
        } else {
            swal("Data tidak jadi dihapus");
        }
    });
    })
</script>
</div>
</div>
</div>
@endsection

```

Lampiran 17 code detail data

```

@foreach ($data as $row)
<div class="modal fade" id="detailModal-{{ $row->id }}" tabindex="-1" role="dialog" aria-
labelledby="tambahModalLabel"
aria-hidden="true">
<div class="modal-dialog modal-lg" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="tambahModalLabel">Detail Data</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form action="/updatedata/{{ $row->id }}" method="POST" enctype="multipart/form-data">
@csrf

<div class="form-row">
<div class="form-group col-md-6">
<label for="nik">NIK:</label>
<div class="input-group">
<input type="number" class="form-control" id="nik-{{ $row->nik }}" name="nik"
value="{{ $row->nik }}" readonly>
<div class="input-group-append">
<button type="button" class="btn btn-primary copy-button"

```



```

        onclick="copyToClipboard(this)">
        <i class="fas fa-copy"></i>
    </button>
</div>
</div>
</div>

```

Lampiran 18 code routes/api.php

```

Route::post('/auth/register', [AuthContoller::class, 'register']);
Route::post('/auth/login', [AuthContoller::class, 'login']);
Route::post('/test', function () {
    return response([
        'message' => "Api is working"
    ],200);
});

Route::get('/jumlah-anggota', [AnggotaApiController::class, 'jumlah']);
Route::get('/anggota/{id_flutteruser}', [AnggotaApiController::class, 'getDataByIdFlutterUser']);
Route::post('/anggota/{id_flutteruser}', [AnggotaApiController::class, 'anggota']);
Route::put('/anggota/{id}', [AnggotaApiController::class, 'update']);
Route::get('/daftaranggota/ceknik', [AnggotaApiController::class, 'ceknik']->name('ceknik'));

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});

Route::get('/daftar-buku', [BukuApiController::class, 'index']);
Route::get('/gambarbuku/{filename}', [BukuApiController::class, 'getGambar']);
Route::get('/daftarbuku/{kategori}', [BukuApiController::class, 'index']);
Route::get('/daftarbukuid/{id}', [BukuApiController::class, 'daftarbukuid']);
Route::get('/search', [BukuApiController::class, 'search']);

```

Lampiran 19 routes/web.php

```

Route::get('/', function () {
    $jumlahanggota = daftaranggota::count();
    $jumlahanggotacowo = daftaranggota::where('jeniskelamin','Laki-laki')->count();
    $jumlahanggotacewe = daftaranggota::where('jeniskelamin','Perempuan')->count();
    return view('welcome', compact('jumlahanggota','jumlahanggotacowo','jumlahanggotacewe'));
})->middleware('auth');
// Route::get('/login', function () {
//     return view('login.login');
// });

Route::get('/daftaranggota', [DaftaranggotaController::class, 'index']->name('daftaranggota')->middleware('auth'));

```

```

Route::get('/daftarregister', [DaftaranggotaController::class, 'daftarregister'])->name('daftarregister')->middleware('auth');
Route::get('/deleteregister/{id}', [DaftaranggotaController::class, 'deleteregister'])->name('deleteregister');
Route::post('/insertdata', [DaftaranggotaController::class, 'insertdata'])->name('insertdata');
Route::post('/updatedata/{id}', [DaftaranggotaController::class, 'updatedata'])->name('updatedata');
Route::get('/delete/{id}', [DaftaranggotaController::class, 'delete'])->name('delete');

Route::get('/login', [LoginController::class, 'login'])->name('login');
Route::post('/loginproses', [LoginController::class, 'loginproses'])->name('loginproses');
Route::get('/register', [LoginController::class, 'register'])->name('register');
Route::post('/registeruser', [LoginController::class, 'registeruser'])->name('registeruser');
Route::get('/logout', [LoginController::class, 'logout'])->name('logout');

Route::get('/daftarbuku', [DaftarbukuController::class, 'daftarbuku'])->name('daftarbuku');
Route::post('/insertdata', [DaftarbukuController::class, 'insertdata'])->name('insertdata');
Route::get('/deletebuku/{id}', [DaftarbukuController::class, 'deletebuku'])->name('deletebuku');

```

Lampiran 20 code app/http/controllers/logincontroller.php

```

<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Support\Str;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    public function login(){
        // return view('login.login');
        return view('login.logintest');
    }
    public function loginproses(Request $request){
        if(Auth::attempt($request->only('name', 'password'))){
            return redirect('/');
        }

        return redirect('login')->withErrors(['message' => 'Nama pengguna atau kata sandi salah']);
    }

    public function register(){

```

```

        return view('login.register');
    }
    public function registeruser(Request $request) {
        User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => bcrypt($request->password),
            'remember_token' => Str::random(60),
        ]);

        return redirect('/login');
        // dd($request->all());
    }
    public function logout() {
        Auth::logout();
        return redirect('login');
    }
}
}

```

Lampiran 21 code app/http/controllers/daftaranggotacontroller.php

```

<?php

namespace App\Http\Controllers;

use App\Models\daftaranggota;
use App\Models\FlutterUser;
use Illuminate\Http\Request;

class DaftaranggotaController extends Controller
{
    public function index()
    {
        $data = daftaranggota::all();

        return view('daftaranggota', compact('data'));
    }

    public function insertdata(Request $request)
    {
        daftaranggota::create($request->all());

        return redirect()->route('daftaranggota')->with('success', 'Data Berhasil Ditambah');
    }

    public function updatedata(Request $request, $id)
    {
        $data = daftaranggota::find($id);
    }
}

```

```

        $data->update($request->all());
        return redirect()->route('daftaranggota')->with('success', 'Data Berhasil Diupdate');
    }

    public function delete($id){
        $data = daftaranggota::find($id);
        $data->delete();
        return redirect()->route('daftaranggota')->with('success', 'Data Berhasil Dihapus');
    }

    public function daftarregister() {
        $data = FlutterUser::all();
        return view('daftarregister', compact('data'));
    }

    function deleteregister($id){
        $data = FlutterUser::find($id);
        $data->delete();
        return redirect()->route('daftarregister')->with('success', 'Data Berhasil Dihapus');
    }
}

```

Lampiran 22 code app/http/controllers/anggotaapicontroller.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\daftaranggota;
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;

class AnggotaApiController extends Controller
{
    public function index()
    {
        $Sanggota = daftaranggota::all();
        return response()->json(['message' => 'Success', 'data' => $Sanggota,]);
    }

    public function anggota(Request $request)
    {
        $request->validate([
            'id flutteruser' => 'unique:daftaranggotas,id flutteruser',

```

```

    // tambahkan aturan validasi lainnya sesuai kebutuhan
  });
  $Sanggota = daftaranggota::create($request->all());
  return response()->json(['message' => 'success', 'data' => $Sanggota]);
}

public function update(Request $request, $id)
{
  $Sanggota = daftaranggota::find($id);
  $Sanggota->update($request->all());
  return response()->json(['message' => 'success', 'data' => $Sanggota]);
}

public function getDataByIdFlutterUser(Request $request, $id_flutteruser)
{
  // Menggunakan join untuk mengambil data sesuai id_flutteruser
  $Sanggota = daftaranggota::join('flutterusers', 'daftaranggotas.id_flutteruser', '=', 'flutterusers.id')
    ->where('daftaranggotas.id_flutteruser', $id_flutteruser)
    ->select('daftaranggotas.*', 'flutterusers.username')
    ->first();

  if ($Sanggota) {
    return response()->json(['message' => 'Success', 'data' => $Sanggota]);
  } else {
    return response()->json(['message' => 'Data not found'], 404);
  }
}
}

```

Lampiran 23 wawancara

No	Pertanyaan	Jawaban
1	Bagaimana proses pendaftaran perpustakaan yang saat ini berlaku di Dinas Perpustakaan Kota Samarinda?	Putry Desintia Ramadhani menjelaskan bahwa proses pendaftaran anggota perpustakaan di Dinas Perpustakaan Kota Samarinda saat ini dilakukan secara manual. Calon anggota harus datang langsung ke perpustakaan, mengisi formulir pendaftaran, dan menyerahkan dokumen yang diperlukan seperti fotokopi KTP atau kartu identitas lainnya. Petugas perpustakaan kemudian memverifikasi data yang diberikan sebelum memberikan kartu anggota.
2	Apa saja kendala atau masalah yang sering dihadapi dalam proses pendaftrana anggota perpustakaan?	Menurut Putry Desintia Ramadhani, kendala utama yang sering dihadapi dalam proses pendaftaran anggota perpustakaan adalah kesalahan dalam pengisian formulir pendaftaran oleh calon anggota dan keterlambatan dalam proses verifikasi data oleh petugas. Terdapat beberapa masalah tambahan, seperti ketika kertas formulir habis, petugas perpustakaan harus mencetak kertas formulir terlebih dahulu, yang memakan banyak waktu. Selain itu, terdapat masalah ketika kertas print khusus untuk formulir habis, sehingga petugas harus mencarinya di gudang

- atau membelinya terlebih dahulu untuk mendapatkan kertas tersebut. Hal ini mengakibatkan berkurangnya efektivitas dan efisiensi terhadap layanan perpustakaan.
- 3 Setelah calon anggota mengisi fomulir dan di serahkan ke petugas, apakah fomulir itu di simpan untuk dibackup atau dibuang? Putry Desintia Ramadhani menerangkan bahwa formulir pendaftaran yang telah diisi oleh calon anggota tidak disimpan sebagai backup, melainkan dibuang setelah diproses.
 - 4 Fitur-fitur apa saja yang diharapkan dapat dimiliki oleh aplikasi pendaftaran anggota perpustakaan berbasis android ini? Putry Desintia Ramadhani mengharapkan aplikasi pendaftaran anggota perpustakaan berbasis Android ini memiliki berbagai fitur yang memudahkan proses pendaftaran dan pengelolaan data anggota. Fitur-fitur yang diharapkan antara lain adalah:
 - Login: Memungkinkan pengguna untuk masuk ke aplikasi dengan akun yang telah terdaftar.
 - Register: Memungkinkan calon anggota untuk mendaftar dan membuat akun baru.
 - Menginput data: Memungkinkan pengguna untuk mengisi dan menyimpan data pendaftaran anggota baru.
 - Menghapus data: Memungkinkan petugas untuk menghapus data anggota yang tidak valid atau tidak diperlukan lagi.
 - Mencari data: Memungkinkan petugas untuk mencari data anggota berdasarkan kriteria tertentu.
 - Melihat detail data: Memungkinkan petugas untuk melihat detail data anggota secara lengkap dan terperinci.
 - 5 Bagaimana sistem verifikasi dan validasi data calon anggota yang saat ini diterapkan di Dinas Perpustakaan Kota Samarinda? Putry Desintia Ramadhani menjelaskan bahwa sistem verifikasi dan validasi data calon anggota yang saat ini diterapkan di Dinas Perpustakaan Kota Samarinda dilakukan secara manual oleh petugas perpustakaan. Petugas memeriksa keaslian dokumen yang diserahkan oleh calon anggota dan memastikan bahwa semua informasi yang diberikan sesuai dengan persyaratan.
 - 6 Berapa rata-rata pendaftaran anggota baru di perpustakaan perhari? Menurut Putry Desintia Ramadhani, perpustakaan menerima 15 hingga 20 anggota baru setiap harinya.
-

SKRIPSI MUHAMMAD RAMADHANI SAPUTRA

by Teknik Informatika UMKT



Submission date: 16-Jul-2024 02:04PM (UTC+0800)

Submission ID: 2417648066

File name: 087_Muhammad_Ramadhani_Saputra_-_MUHAMMAD_RAMADHANI_SAPUTRA.docx (6.14M)

Word count: 8244

Character count: 52500

SKRIPSI MUHAMMAD RAMADHANI SAPUTRA

ORIGINALITY REPORT

22% SIMILARITY INDEX	21% INTERNET SOURCES	7% PUBLICATIONS	7% STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	-----------------------------

PRIMARY SOURCES

1	widuri.raharja.info Internet Source	2%
2	repository.ar-raniry.ac.id Internet Source	1%
3	Submitted to Universitas Pamulang Student Paper	1%
4	repository.ub.ac.id Internet Source	1%
5	repo.unikadelasalle.ac.id Internet Source	1%
6	repository.uin-suska.ac.id Internet Source	1%
7	ojs.trigunadharma.ac.id Internet Source	1%
8	repository.its.ac.id Internet Source	1%
9	123dok.com Internet Source	<1%