

**RANCANG BANGUN APLIKASI MANAJEMEN PERPUSTAKAAN
BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL UNTUK
MENINGKATKAN EFISIENSI PENGELOLAAN DATA BUKU**

SKRIPSI

Karya Tulis sebagai syarat memperoleh

Gelar Sarjana Komputer dari Fakultas Teknologi Informasi

Universitas Bale Bandung

Disusun oleh:

RANGGA KURNIAWAN

NPM. 301200024



PROGRAM STRATA 1

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

BANDUNG

2024

LEMBAR PERSETUJUAN PEMBIMBING

**RANCANG BANGUN APLIKASI MANAJEMEN PERPUSTAKAAN
BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL UNTUK
MENINGKATKAN EFISIENSI PENGELOLAAN DATA BUKU**

Disusun Oleh :

RANGGA KURNIAWAN

NPM. 301200024

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

Baleendah, Juli 2024

Disetujui oleh:

Pembimbing Utama

Pembimbing Pendamping

Yudi Herdiana, S.T., M.T.

NIK. 04104808008

Yaya Suharya, S.Kom., M.T.

NIK. 041043170007

LEMBAR PERSETUJUAN PENGUJI

**RANCANG BANGUN APLIKASI MANAJEMEN PERPUSTAKAAN
BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL UNTUK
MENINGKATKAN EFISIENSI PENGELOLAAN DATA BUKU**

Disusun Oleh :

RANGGA KURNIAWAN

NPM. 301200024

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

Baleendah, Juli 2024

Disetujui oleh:

Penguji 1

Penguji 2

Mohammad Bayu Anggara, S.Kom., M.Kom.

NIK. 04104823002

Sukiman, S.Tr.Kom., S.Pd., M.Kom., MCE.

NIK. 04104821001

LEMBAR PENGESAHAN PROGRAM STUDI

**RANCANG BANGUN APLIKASI MANAJEMEN PERPUSTAKAAN
BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL UNTUK
MENINGKATKAN EFISIENSI PENGELOLAAN DATA BUKU**

Disusun Oleh :

RANGGA KURNIAWAN

NPM. 301200024

SKRIPSI ini telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

Baleendah, Juli 2024

Disetujui oleh:

Mengetahui,
Dekan

Mengesahkan,
Ketua Program Studi

Yudi Herdiana, S.T., M.T.
NIK. 04104808008

Yusuf Muharam, S.Kom., M. Kom.
NIK. 04104820003

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini, saya:

Nama : Rangga Kurniawan
NIM : 301200024
Jurusan : Teknik Informatika
Fakultas : Fakultas Teknologi Informasi
Judul : Rancang Bangun Aplikasi
Perpustakaan Berbasis Web
Untuk Meningkatkan Efisiensi
Pengelolaan Data Buku
Menggunakan *Framework*
Laravel

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

Bandung, 28 Mei 2024

Materai 10.000

Rangga Kurniawan
NPM. 301200024

ABSTRAK

pemanfaatan teknologi informasi dan komunikasi sudah menjadi kunci untuk meningkatkan efektivitas pengelolaan perpustakaan. pengelolaan perpustakaan secara efisien sering kali menjadi tantangan bagi para pustakawan. Banyak perpustakaan yang belum mengadopsi aplikasi Pengelolaan yang mengandalkan teknologi salah satunya adalah SMP KP 2 Majalaya.. hal tersebut menimbulkan beberapa masalah diantaranya ketidakakuratan data, kesulitan dalam melacak inventaris, serta keterlambatan dalam menyajikan laporan dengan adanya penelitian ini diharapkan bisa mengatasi masalah tersebut.

penelitian ini bertujuan untuk merancang dan membangun aplikasi perpustakaan berbasis web yang dapat membantu dalam manajemen perpustakaan secara terkomputerisasi. Penelitian ini menggunakan metode pengumpulan data metode kualitatif yaitu Observasi, Interview dan Studi Pustaka, dengan *Waterfall* sebagai metode pengembangannya dengan tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian,. Alat bantu yang digunakan dalam perancangan sistem adalah *Unified Modelling Language* (UML) yang dimana meliputi use case diagram, *Activity* diagram, dan class diagram. Basis data yang digunakan adalah MySQL dengan struktur tabel yang dirancang sesuai dengan kebutuhan sistem. Sedangkan bahasa pemrograman yang digunakan adalah PHP dan *Framework Laravel*. Pada tahapan perancangan antarmuka, desain aplikasi dibuat menggunakan Figma, dengan desain antarmuka yang dapat mudah di pahami oleh pengguna dan tentunya dengan memperhatikan aspek *user experience*. Fitur-fitur utama aplikasi meliputi pengelolaan data buku, pengelolaan data anggota, pengelolaan peminjaman buku, pembuatan laporan. Pengujian aplikasi dilakukan dengan metode *black box* testing untuk memastikan fungsionalitas dan non-fungsionalitas aplikasi berjalan dengan baik serta menggunakan metode validitas untuk mendapatkan nilai kelayakan dari aplikasi yang telah dibuat.

Hasil dari pengujian menunjukkan dengan menggunakan metode *black box* didapatkan bahwa aplikasi dapat bekerja sesuai dengan fungsinya sedangkan hasil yang didapat dari kuesioner yang di berikan kepada pengguna mendapatkan nilai rata rata capaian untuk seluruh aspek adalah 95,83% sedangkan pengujian terhadap

anggota mendapatkan capaian untuk seluruh aspek sebesar 91.10%.dengan capaian tersebut menunjukkan bahwa aplikasi sudah berfungsi dengan baik.Hasil dari penelitian ini adalah aplikasi manajemen perpustakaan yang nantinya diharapkan dapat membantu petugas perpustakaan dalam meningkatkan efisiensi operasional, akurasi data yang lebih tinggi, kemudahan akses informasi, dan penyediaan laporan Pengelolaan yang lebih cepat dan terstruktur di perpustakaan SMP KP 2 Majalaya

Kata Kunci : aplikasi web, *Laravel*, MySQL, perpustakaan , PHP, UML , *Waterfall*

ABSTRACT

The utilization of information and communication technology has become the key to improving the effectiveness of library management. efficient library management is often a challenge for librarians. many libraries have not adopted management applications that rely on technology, one of which is SMP KP 2 Majalaya. this has caused several problems including data inaccuracies, difficulty in tracking inventory, and delays in presenting reports. with this research, it is hoped that it can overcome these problems.can overcome these problems. this research aims to design and develop a web-based library application that facilitates computerized library management. The research employs a qualitative data collection method, encompassing observation, interviews, and literature review. The Waterfall method serves as the development methodology, with phases including requirements analysis, system design, implementation, testing, and documentation. Unified Modeling Language (UML) tools, including use case diagrams, Activity diagrams, and class diagrams, aid in system design. MySQL serves as the database, with table structures tailored to system requirements. PHP and the Laravel Framework

This research aims to design and build web-based library applications that can assist in computerized library management. This research uses qualitative method data collection methods, namely observation, interview and literature study, with Waterfall as its development method with the stages of needs analysis, system design, implementation, and testing. The tools used in system design are Unified Modeling Language (UML) which includes use case diagrams, Activity diagrams, and class diagrams. The database used is MySQL with a table structure designed according to system requirements, while the programming language used is PHP and the Laravel Framework. At the interface design stage, the application design is made using Figma, with an interface design that can be easily understood by users and of course by paying attention to user experience aspects. The main features of the application include book data management, member data management, book borrowing management, report generation. Application testing is carried out using the black box testing method to ensure the functionality and non-functionality of the application runs well and uses the validity method to get the feasibility value of the application that has been made.

The results of the test show that using the black box method it is found that the application can work according to its function while the results obtained from the questionnaire given to users get an average value of achievement for all aspects of 95.83% while testing of members gets an achievement for all aspects of 91.10%. with these achievements showing that the application is functioning properly. The result of this research is a library management application that will be expected to help library staff in increasing operational efficiency, higher data accuracy, easy access to information, and providing faster and more structured management reports in the KP 2 Majalaya Junior High School library.

Keywords : *Web applicatio, Laravel, MySQL,library,PHP,UML, Waterfall*

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Allah SWT, yang telah memberikan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini tepat sesuai waktunya. Tak lupa shalawat serta salam kita curahkan kepada jungjunan Nabi Muhammad SAW.

Atas izin Allah SWT, penulis dapat menyelesaikan skripsi ini dengan judul **“RANCANG BANGUN APLIKASI MANAJEMEN PERPUSTAKAAN BERBASIS WEB MENGGUNAKAN *FRAMEWORK LARAVEL* UNTUK MENINGKATKAN EFISIENSI PENGELOLAAN DATA BUKU (STUDI KASUS PADA SMP KP 2 MAJALAYA)”** yang disusun untuk memenuhi persyaratan dalam mengikuti Skripsi. Untuk itu ucapkan terimakasih kepada Bapak Yusuf Muharam, S.Kom.,M. Kom.. selaku Kepala Program Studi Teknik Informatika.

Dalam penyusunan laporan ini, penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari kesempurnaan karena pengalaman dan pengetahuan penulis yang terbatas.oleh karena itu penulis memohon saran dan kritik dari teman teman maupun dosen demi tercapainya skripsi yang lebih baik

Penyusunan Laporan ini tidak lepas dari bantuan dan dukungan dari beberapa pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT yang telah memberikan kemudahan serta kelancaran sehingga penulis dapat menjalani proses demi proses dalam penelitian maupun penulisan laporan penelitian ini.
2. Orang tua yang selalu memberikan dukungan, bimbingan, do'a, serta nasihat kepada penulis.
3. Bapak Yudi Herdiana, S.T., M.T., selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung sekaligus pembimbing utama.
4. Bapak Yaya suharya, M.T selaku pembimbing pendamping
5. Bapak Yusuf Muharam, S. Kom., M.Kom, selaku Kepala Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung

6. Seluruh jajaran Dosen dan Staff Fakultas Teknologi Informasi Universitas Bale Bandung.
7. Rekan-rekan Fakultas Teknologi Informasi tahun 2020 yang selalu memberikan semangat serta bantuan kepada penulis dalam proses penelitian ataupun penyusunan laporan.
8. Muhammad Iqbal,Jujun Junaedi,Muhammad Anwar ibrahim,Narwin, Fadli Adhitia Mahardika,Sandi Rusman Hadi ,selaku sahabat yang turut memberikan semangat, apresiasi, do'a dan bantuan kepada penulis.
9. Sahabat seperjuangan kuliah grup “Pemuda Pencari Pahala” yaitu Dadan Hardiansah, Defit Taufik Ismail, Diana Salendra, Fahrul Zaman, Taufik Hidayat, dan Teja Kusumah. Terima kasih telah selalu membantu penulis dalam penyusunan laporan skripsi, menjadi tempat berbagi keluh kesah dan selalu menemani dalam suka maupun duka selama masa kuliah.
10. Diriku yang selalu berusaha berkembang melewati segala rintangan dan lika liku kehidupan serta menjadi versi terbaik dari diri sendiri.Kamu keren
11. Teman saya nurul karena telah membantu penulis dalam pencetakan tanpa bantuan mu skripsi ini hanya akan menjadi sebuah pdf.

DAFTAR ISI

ABSTRAK.....	vi
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvii
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Metodologi penelitian.....	3
1.5.1 Metode pengumpulan data.....	3
1.5.2 Metode pengembangan	3
1.6 Sistematika penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1 Landasan Teori	5
2.1.1 Acuan Penelitian Pertama	9
2.1.2 Acuan Penelitian Kedua.....	10
2.1.3 Acuan Penelitian Ketiga	10
2.2 Dasar Teori	11
2.2.1 Aplikasi	11
2.2.2 Sistem.....	11
2.2.3 Website	11

2.2.4 Perpustakaan	12
2.2.5 Basis data	13
2.2.6 Metode SDLC <i>Waterfall</i>	13
2.2.7 UML.....	15
2.2.8 XAMPP	19
2.2.9 Visual Studio Code	19
2.2.10 Figma.....	19
2.2.11 Draw.io	20
2.2.12 Mysql.....	20
2.2.13 PHP	21
2.2.14 Bootstrap.....	22
2.2.15 <i>Laravel</i>	22
2.2.16 Java Script	24
2.2.17 HTML.....	24
2.2.18 CSS	24
2.2.19 Metode pengujian penelitian	25
BAB III METODOLOGI PENELITIAN	26
3.1 Kerangka Berfikir	26
3.2 Deskripsi	27
3.1.1 Identifikasi Masalah.....	27
3.1.2 Pengumpulan Data	27
3.1.3 Analisis Kebutuhan.....	27
3.1.4 Perancangan	29
3.1.5 Implementasi.....	31
3.1.6 Pengujian	31
3.1.7 Pembuatan Laporan.....	31

BAB IV ANALISIS DAN PERANCANGAN	32
4.1 Analisis	32
4.1.1 Analisis masalah	32
4.1.2 Analisis Software	32
4.1.3 Analisis Pengguna.....	33
4.1.4 Desain antarmuka	33
4.1.5 Fitur-Fitur.....	33
4.1.6 Analisis Data.....	34
4.1.7 Analisis Biaya	35
4.2 Perancangan.....	36
4.2.1 Struktur Tabel	60
4.2.2 Desain Antarmuka	63
BAB V IMPLEMENTASI DAN PENGUJIAN.....	70
5.1 Implementasi	70
5.1.1 Listing Program	70
5.1.2 Implementasi sistem	75
5.1.3 Spesifikasi Sistem.....	75
5.1.4 Instalasi Sistem	76
5.1.5 Menjalankan Sistem.....	77
5.2 Pengujian Aplikasi.....	84
BAB VI KESIMPULAN	94
6.1 Kesimpulan.....	94
6.2 Saran....	95
DAFTAR PUSTAKA	96
LAMPIRAN.....	101

DAFTAR GAMBAR

Gambar 2. 1 <i>Waterfall</i>	14
Gambar 3. 1 Kerangka Berfikir.....	26
Gambar 4. 1 Use Case User	36
Gambar 4. 2 Use Case Diagram.....	37
Gambar 4. 3 <i>Activity</i> Diagram Login	40
Gambar 4. 4 <i>Activity</i> Diagram Registrasi.....	41
Gambar 4. 5 <i>Activity</i> Diagram Lupa Password	42
Gambar 4. 6 <i>Activity</i> Diagram Tambah Kelas	43
Gambar 4. 7 <i>Activity</i> Diagram Hapus Kelas	44
Gambar 4. 8 <i>Activity</i> Diagram Edit Kelas.....	45
Gambar 4. 9 <i>Activity</i> Diagram Tambah Buku.....	46
Gambar 4. 10 <i>Activity</i> Diagram Hapus Buku.....	47
Gambar 4. 11 <i>Activity</i> Diagram Edit Buku	48
Gambar 4. 12 <i>Activity</i> Diagram Tambah Anggota.....	49
Gambar 4. 13 <i>Activity</i> Diagram Hapus Anggota.....	50
Gambar 4. 14 <i>Activity</i> Diagram Edit Anggota	51
Gambar 4. 15 <i>Activity</i> Diagram Peminjaman	52
Gambar 4. 16 Pengembalian Buku.....	53
Gambar 4. 17 <i>Activity</i> Diagram Buku Masuk	54
Gambar 4. 18 <i>Activity</i> Diagram Cetak Laporan	55
Gambar 4. 19 <i>Activity</i> Diagram Buku Keluar	56
Gambar 4. 20 <i>Activity</i> Diagram Halaman Login User.....	57
Gambar 4. 21 <i>Activity</i> Diagram Peminjaman User	58
Gambar 4. 22 Class Diagram	59
Gambar 4. 23 Desain Antarmuka Login	64
Gambar 4. 24 Desain Antarmuka Reset Password.....	65
Gambar 4. 25 Desain Antarmuka Dashboard	65
Gambar 4. 26 Desain Antarmuka Anggota	65
Gambar 4. 27 Desain Antarmuka Buku	66
Gambar 4. 28 Desain Antarmuka Kelas.....	66
Gambar 4. 29 Desain Antarmuka Peminjaman.....	67

Gambar 4. 30 Desain Antarmuka Buku Masuk	67
Gambar 4. 31 Desain Antar Muka Buku Keluar.....	68
Gambar 4. 32 Desain Antarmuka User	68
Gambar 4. 33 Desain Antarmuka Dashboard User.....	69
Gambar 4. 34 Desain Antarmuka Peminjaman.....	69
Gambar 5. 1 Halaman Login.....	78
Gambar 5. 2 Halaman Register	78
Gambar 5. 3 Halaman Lupa Password.....	79
Gambar 5. 4 Halaman Dashboard	79
Gambar 5. 5 Halaman Buku.....	80
Gambar 5. 6 Halaman Kelas	80
Gambar 5. 7 Halaman Anggota.....	81
Gambar 5. 8 Halaman Peminjaman	81
Gambar 5. 9 Halaman Buku Masuk.....	82
Gambar 5. 10 Halaman Buku Keluar.....	82
Gambar 5. 11 Halaman Login User	83
Gambar 5. 12 Halaman Dashboard User.....	83
Gambar 5. 13 Halaman Peminjaman Anggota.....	84
Gambar 5. 14 Grafik Kuesioner 1	87
Gambar 5. 15 Grafik Kuesioner 2	88
Gambar 5. 16 Grafik Kuesioner 3.....	88
Gambar 5. 17 Grafik Kuesioner 4.....	88
Gambar 5. 18 Grafik Kuesioner 5	89
Gambar 5. 19 Grafik Kuesioner 6	89
Gambar 5. 20 Grafik Kuesioner 7	89
Gambar 5. 21 Grafik Kuesioner 8.....	90
Gambar 5. 22 Grafik Kuesioner 9	90
Gambar 5. 23 Grafik Kuesioner 10.....	90
Gambar 5. 24 Grafik Kuesioner 11	91
Gambar 5. 25 Grafik Kuesioner 12.....	91
Gambar 5. 26 Grafik Kuesioner 13	91
Gambar 5. 27 Grafik Kuesioner 14.....	92

Gambar 5. 28 Grafik Kuesioner 15	92
Gambar 5. 29 Grafik Kuesioner 16	92
Gambar 5. 30 Grafik Kuesioner 17	93
Gambar 5. 31 Grafik Kuesioner 18	93

DAFTAR TABEL

Tabel 2. 1 Acuan Jurnal Penelitian	5
Tabel 2. 2 Use Case Diagram.....	16
Tabel 2. 3 <i>Activity</i> Diagram	17
Tabel 2. 4 Class Diagram	18
Tabel 4. 2 Analisis Biaya	35
Tabel 4. 3 Deskripsi Actor	37
Tabel 4. 4 Deskripsi Use Case Diagram	38
Tabel 4. 5 Tabel User	60
Tabel 4. 6 Tabel Anggota.....	60
Tabel 4. 7 Tabel Buku	61
Tabel 4. 8 Tabel Buku Masuk	61
Tabel 4. 9 Tabel Buku Keluar	62
Tabel 4. 10 Tabel Kelas	62
Tabel 4. 11 Tabel Peminjaman	63
Tabel 5. 1 Spesifikasi Perangkat Keras.....	75
Tabel 5. 2 Spesifikasi Perangkat Lunak	76
Tabel 5. 3 Pengujian Aplikasi	84
Tabel 5. 4 Hasil Uji Oleh Petugas Perpustakaan.....	86
Tabel 5. 5 Hasil Uji Oleh Anggota.....	87

DAFTAR LAMPIRAN

Lampiran 1 : Dokumentasi.....	101
Lampiran 2 : Wawancara	102
Lampiran 3 : Tor (Term Of Reference).....	104
Lampiran 4 : Hasil Kuesioner	105
Lampiran 5 : Listing Program	107
Lampiran 6 : Riwayat Hidup.....	119

BAB I

PENDAHULUAN

1.1 Latar belakang

Perpustakaan sekolah adalah perpustakaan yang berada pada lembaga pendidikan sekolah sebagai pusat sumber belajar mengajar. Atas pernyataan tersebut, beberapa tugas perpustakaan sekolah adalah mengelola datadata perpustakaan, memberikan layanan pemimjaman buku, menyediakan tempat membaca dan mengembangkan agar mudah dikontrol dan diakses oleh anggota perpustakaan. Menyediakan sumber-sumber referensi untuk menambah pengetahuan siswa dan guru. Mendidik siswa agar dapat mencari dan mengumpulkan informasi (referensi) secara mandiri (Hartono, 2016) Pada era globalisasi saat ini, teknologi informasi dan komunikasi semakin canggih dan cepat. Oleh karena itu, perpustakaan harus dapat memanfaatkan teknologi informasi dan komunikasi dengan akurat, relevan dan tepat waktu (Saputra & Nugroho, 2017)

SMP KP 2 Majalaya adalah salah satu sekolah menengah pertama swasta yang ada di Jl-Wangisagara,desa Wangisagara,Kecamatan Majalaya,Kabupaten Bandung Jawa Barat yang mana memiliki fasilitas yang cukup lengkap salah satunya perpustakaan.selain memiliki fasilitas yang bisa dikatakan lengkap SMP KP 2 majalaya juga menjadi Salah satu sekolah yang memiliki akreditasi A yang mana membuat SMP KP 2 Menjadi salah satu sekolah favorit di wilayah majalaya,namun dengan banyak nya siswa siswi yang bersekolah di SMP KP 2 Majalaya menimbulkan beberapa masalah baru khususnya untuk Pengelolaan perpustakaan seperti ketidakakuratan data, kesulitan dalam melacak inventaris, serta keterlambatan dalam menyajikan laporan.

Perpustakaan SMP KP 2 Majalaya menghadapi kendala dalam pengelolaan buku karena sistem manual yang rentan kesalahan, sehingga data menjadi tidak akurat dan pelacakan inventaris sulit dilakukan. Hal ini berakibat pada keterlambatan pelaporan, informasi yang tidak akurat, pengambilan keputusan yang tidak tepat, dan penurunan minat baca siswa. Implementasi sistem atau aplikasi pengelolaan perpustakaan digital menjadi solusi untuk meningkatkan akurasi data, kemudahan pelacakan inventaris, otomatisasi pelaporan, efisiensi operasional, meningkatkan minat baca siswa.

Pada penelitian terdahulu telah menunjukkan bahwa adanya aplikasi perpustakaan dapat membantu petugas perpustakaan penelitian tersebut di antaranya Hendra Sopandi , Novita Anggraini (2024), kemudahan penelitian yang dilakukan Rahman Kurniadi, Cecep Riki, Milah Nurkamilah (2022) lalu ada penelitian yang dilakukan oleh Rosita Asnur, Bernadus Very Christioko (2022) namun dari 3 jurnal acuan diatas tidak ada yang menggunakan *Framework Laravel* dalam pembuatan code programnya. dalam penelitian yang penulis lakukan penulis membuat aplikasi yang memiliki design UI UX yang lebih intuitif dan fungsional serta aplikasi yang dibuat dapat memberikan informasi secara real time.

Dengan memanfaatkan fitur *reporting real-time*, pustakawan dapat melihat informasi mengenai ketersediaan buku jumlah peminjam dan informasi lainnya secara *real-time* tujuan penelitian ini diharapkan perpustakaan dapat mempermudah dalam aspek Pengelolaan buku di perpustakaan, serta diharapkan memperoleh manfaat yang signifikan seperti peningkatan efisiensi operasional, akurasi data yang lebih tinggi, kemudahan akses informasi, dan penyediaan laporan Pengelolaan yang lebih cepat dan terstruktur. oleh karena itu, dilakukan penelitian yang berjudul

“RANCANG BANGUN APLIKASI MANAJEMEN PERPUSTAKAAN BERBASIS WEB MENGGUNAKAN *FRAMEWORK LARAVEL* UNTUK MENINGKATKAN EFISIENSI PENGELOLAAN DATA BUKU”

1.2 Rumusan Masalah

Adapun rumusan masalah dari latar belakang yang telah di kemukakan diatas adalah sebagai berikut :

1. Bagaimana merancang dan membangun aplikasi yg dapat membantu pengelola perpustakaan dalam melacak inventaris buku
2. Bagaimana merancang dan membangun aplikasi yg dapat mempermudah dalam pembuatan laporan secara cepat
3. Bagaimana merancang dan membangun aplikasi yg dapat meningkatkan akurasi data pada pengelolaan data buku di perpustakaan

1.3 Batasan Masalah

Adapun batasan masalah yang ada agar permasalahan sesuai yang dituju maka perlu ditetapkan batasan masalah sebagai berikut :

1. Penelitian dilakukan di SMP KP 2 Majalaya di Jl Wangisagara, Desa wangisagara Kec. Majalaya, Kabupaten Bandung, Jawa Barat
2. Aplikasi manajemen perpustakaan ini meliputi data buku, data siswa, data transaksi, dan pembuatan laporan.
3. Metode yang digunakan dalam penelitian ini adalah *Waterfall* tahap tahap yang digunakan dalam penelitian ini diantaranya analisis kebutuhan, perancangan, implementasi dan pengujian.
4. Sistem informasi dibangun dengan memanfaatkan *FRAMEWORK Laravel*.

1.4 Tujuan Penelitian

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

4. Merancang dan membangun aplikasi yg dapat membantu pengelola perpustakaan dalam melacak inventaris buku
5. Merancang dan membangun aplikasi yg dapat mempermudah dalam pembuatan laporan secara cepat
6. Merancang dan membangun aplikasi yg dapat meningkatkan akurasi data pada pengelolaan data buku di perpustakaan

1.5 Metodologi penelitian

Dalam penelitian ini penulis melakukan pengumpulan data dan melakukan pengembangan dengan menggunakan metode metode berikut :

1.5.1 Metode pengumpulan data

Metode pengumpulan data yang dilakukan dalam pelaksanaan pengerjaan skripsi, penulis menggunakan beberapa metode untuk mendapatkan data yang akurat ,yaitu observasi langsung terhadap sistem pengelolaan buku di perpustakaan SMP KP 2 Majalaya, wawancara dengan pihak pustakawan mengenai pengelolaan buku di perpustakaan, dan studi Pustaka dari jurnal dan sumber terkait.

1.5.2 Metode pengembangan

Adapun metode pengembangan yang digunakan adalah metode pengembangan perangkat lunak yaitu *Waterfall*, *Waterfall* adalah pendekatan dalam pengembangan perangkat lunak di mana setiap fase harus diselesaikan sebelum melanjutkan ke fase berikutnya. Prosesnya mencakup analisis sistem,

perancangan, implementasi, pengujian, dan pemeliharaan. Hasil dari setiap fase menjadi dasar untuk fase selanjutnya, dan umumnya, perubahan besar tidak dapat dilakukan sampai fase sebelumnya selesai.

1.6 Sistematika penulisan

Sistematika penulisan bertujuan untuk memudahkan peneliti dalam menyusun skripsi yang sistematis. Adapun sistematika penulisan yang digunakan adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini mencakup latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

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

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metodologi penelitian yang dipakai dan kerangka penelitian atau tahap-tahap yang dilakukan atau dilalui dalam penelitian yang dilakukan

BAB IV ANALISIS, PERANCANGAN DAN HASIL

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

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi penyajian tahap pengembangan aplikasi yang akan dijelaskan tiap langkahnya dan menampilkan tampilan dari setiap fitur dari aplikasi yang dibuat serta penjelasan bagaimana aplikasi berjalan

BAB VI PENUTUP

Bab ini berisi kesimpulan serta saran untuk pengembangan aplikasi selanjutnya

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Pada penelitian ini, penyusun menjadikan pengetahuan yang telah di pelajari di kampus menjadi dasar pengetahuan dalam pembuatan Aplikasi Pengelolaan Perpustakaan Berbasis Web Menggunakan *FRAMEWORK Laravel* Pada SMP KP 2 Majalaya diantaranya sebagai berikut :

1. Teori Algoritma dan Pemograman 1 serta Algoritma dan Pemograman 2.

Dalam penelitian ini teori algoritma dan pemograman dipakai dalam pembuatan kode program karena algoritma merupakan kerangka dalam pembuatan kode program,

2. Teori Interaksi Manusia dan Komputer.

Berfungsi dalam pembuatan desain antar muka yang intuitif dan fungsional

3. Teori Pemograman Internet.

Dalam penelitian ini penulis membuat aplikasi yang berbasis web Dimana teori pemograman internet membantu dalam pembuatan aplikasi yang dibuat

4. Teori sistem basis data

Membantu dalam pembuatan basis data atau data base dalam mengelola data yg masuk maupun keluar

5. Teori Replika perangkat lunak

Membantu dalam pengembangan aplikasi seperti metode pengembangan apa yang akan digunakan

Selain menggunakan pengetahuan yang di dapat di kampus penulis juga mempelajari penelitian terdahulu sebagai acuan dalam penelitian, penelitian penelitian yang di jadikan acuan adalah sebagai berikut:

Tabel 2. 1 Acuan Jurnal Penelitian

No	Jurnal Penelitian	Masalah	Metode	Kesimpulan
1	Judul : Aplikasi perpustakaan	1. Kesulitan dalam Pengolahan	Metode <i>Waterfall</i>	Aplikasi perpustakaan

	<p>Berbasis Web Menggunakan <i>Framework</i> Codeigniter (Studi Kasus SMPN 3 Pacet)</p> <p>Nama Penulis : Hendra Sopandi , Novita Anggraini Tahun : 2024</p>	<p>Data: Semakin banyak transaksi yang terjadi, semakin sulit untuk mengelola dan menganalisis data secara manual</p> <p>2. Pencarian Buku yang Lambat: Proses pencarian buku yang dipinjam secara manual memakan waktu dan seringkali tidak efisien.</p> <p>3. Kesalahan dalam Laporan: Pembuatan laporan secara manual rentan terhadap kesalahan manusia, baik itu dalam penghitungan statistik maupun dalam mencatat informasi</p>		<p>mempermudah petugas dan siswa dalam transaksi peminjaman serta pengembalian buku. Dengan akses data buku secara langsung, pencarian buku jadi lebih cepat. Selain itu, penghitungan peminjaman dan pembuatan laporan menjadi lebih efisien berkat sistem terkomputerisasi.</p>
--	--	---	--	---

		transaksi dengan benar		
2	<p>Judul : Rancang Bangun Aplikasi Perpustakaan berbasis Web dengan Menggunakan <i>Framework</i> CodeIgniter</p> <p>Nama Penulis : Rahman Kurniadi, Cecep Riki, Milah Nurkamilah</p> <p>Tahun : 2022</p>	<p>1. Petugas perpustakaan bingung saat mencari data buku yang sudah ada di data atau yang belum masuk ke buku besar</p> <p>2. Waktu yang Tidak Efisien: Proses pencarian data yang memakan waktu sangat lama dapat menghambat produktivitas petugas. Misalnya, mencari informasi tentang siapa yang meminjam buku tertentu atau siapa yang belum mengembalikan buku dapat menjadi tugas</p>	<p>Metode Research and Development (R&D).</p>	<p>aplikasi perpustakaan berbasis web mampu menyelesaikan permasalahan yang ada seperti pencarian data transaksi yang tidak membutuhkan banyak waktu.</p> <p>Hasil pengujian aplikasi perpustakaan berbasis web yang diuji oleh ahli rekayasa perangkat lunak ternyata layak digunakan dengan nilai 3,71 dengan kategori baik. Kemudian hasil pengujian oleh petugas perpustakaan</p>

		yang memakan waktu.		mendapat skor 3,46 dengan kategori baik dan dinyatakan layak digunakan.
3	Judul : Sistem Informasi Perpustakaan Berbasis Web Menggunakan <i>Framework</i> Codeigniter Nama Penulis : Rosita Asnur, Bernadus Very Christioko Tahun : 2022	<p>1. Pembuatan Laporan yang Tidak Efisien: Pegawai harus mencatat semua data buku dan informasi peminjam secara manual untuk membuat laporan. Proses ini memakan waktu yang cukup lama dan kurang efisien.</p> <p>2. Kurangnya Akurasi Informasi: Karena pembuatan laporan masih dilakukan secara manual, kemungkinan kesalahan dalam pencatatan data buku dan nama siswa yang meminjam</p>	Metode <i>Waterfall</i>	Sistem Informasi Perpustakaan membantu dalam penginputan data buku, data anggota, data petugas, serta transaksi peminjaman dan pengembalian buku secara cepat dan efisien. Dengan sistem ini, kemungkinan kehilangan data akibat kerusakan atau kehilangan dokumen dapat diminimalisir karena semua data transaksi disimpan secara digital. Selain itu, proses perhitungan denda saat

		meningkat. Hal ini dapat mengakibatkan informasi yang disajikan menjadi kurang akurat.		pengembalian buku menjadi lebih cepat dan akurat karena dilakukan secara otomatis oleh sistem informasi perpustakaan yang dibangun. Hal ini membantu meningkatkan efisiensi dan akurasi dalam pengelolaan perpustakaan secara keseluruhan.
		3. Waktu yang Terbuang: Proses manual dalam mencatat data buku dan nama peminjam menghabiskan waktu yang berharga, yang seharusnya dapat digunakan untuk kegiatan lain yang lebih produktif dalam pengelolaan perpustakaan.		

2.1.1 Acuan Penelitian Pertama

Berdasarkan tabel 2.1 Acuan Penelitian nomor 1 dengan judul Aplikasi perpustakaan Berbasis Web Menggunakan *Framework* Codeigniter (Studi Kasus SMPN 3 Pacet) yang dilakukan oleh Hendra Sopandi , Novita Anggraini pada tahun 2024 pada penelitian ini memiliki persamaan dianatanya Menggunakan metode pengembangan *Waterfall* serta Aplikasi yang dibuat berbasis website Selain persamaan penelitian ini juga memiliki perbedaan yaitu dalam pembuatanya Menggunakan *Framework codeiniter* sedangkan penulis menggunakan *Framework Laravel* Metode penelitian yang gunakan adalah metode berorientasi pada objek yang mana ini dapat digunakan untuk

pengembangan perangkat lunak lebih mudah dipahami. Metodologi berorientasi objek adalah pendekatan pembangunan sistem perangkat lunak yang menggunakan pendekatan objek secara sistematis untuk mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi

2.1.2 Acuan Penelitian Kedua

Berdasarkan tabel 2.1 Acuan Penelitian nomor 1 dengan judul Rancang Bangun Aplikasi Perpustakaan berbasis Web dengan Menggunakan *Framework CodeIgniter* yang dilakukan oleh Rahman Kurniadi, Cecep Riki, Milah Nurkamilah pada tahun 2022 memiliki beberapa kesamaan diantaranya Pengujian dilakukan dengan menggunakan metode black box testing kemudian dalam perancangannya Menggunakan UML

Selain persamaan penelitian ini juga memiliki perbedaan yaitu dalam pembuatannya Menggunakan *Framework codeiniter* sedangkan penulis menggunakan *Framework Laravel* kemudian metode pengembangan aplikasi Menggunakan Metode Research and Development (R&D) Yaitu suatu proses atau langkah-langkah untuk mengembangkan produk baru atau menyempurnakan produk yang sudah ada, yang dapat dipertanggungjawabkan (Sukmadinata, 2005).

2.1.3 Acuan Penelitian Ketiga

Berdasarkan tabel 2.1 Acuan Penelitian nomor 1 dengan judul Sistem Informasi Perpustakaan Berbasis Web Menggunakan *Framework Codeigniter* yang dilakukan oleh Rosita Asnur, Bernadus Very Christioko pada Tahun 2022 memiliki kesamaan diantaranya metode pengembangan yang dipakai adalah metode pengembangan *Waterfall* kemudian metode pengumpulan data yang digunakan adalah observasi, wawancara dan studi pustaka

Selain persamaan penelitian ini juga memiliki perbedaan yaitu pada penelitian ini menggunakan *Framework codeigniter* dalam pembuatan sistem informasi perpustakaan berbasis web

Dari ketiga jurnal yang menjadi acuan dapat disimpulkan bahwa perbedaan antara jurnal acuan dengan penelitian yang dilakukan adalah adanya perbedaan *Framework* yang digunakan dalam pembuatan program yang dimana *Framework* yang digunakan dalam penelitian ini adalah *Laravel* sedangkan jurnal terdahulu menggunakan *codeigniter*

2.2 Dasar Teori

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

2.2.1 Aplikasi

App merupakan bentuk singkatan dari kata application . Aplikasi adalah program perangkat lunak yang dirancang untuk menjalankan fungsi tertentu secara langsung bagi pengguna atau, dalam beberapa kasus, untuk program perangkat lunak lain. Istilah aplikasi dipopulerkan pada masa-masa awal ponsel pintar , setelah Apple memperkenalkan iPhone pada tahun 2007 dan kemudian meluncurkan App Store pada tahun 2008. Pada saat itu, istilah tersebut diterapkan terutama pada aplikasi seluler , yang ringan dan berfokus pada tujuan atau tujuan tertentu. tugas, seperti memeriksa cuaca, mencatat, melakukan perhitungan atau membaca berita. (Rahul Awati. 2024)

2.2.2 Sistem

Sistem dapat didefinisikan sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Kumpulan elemen terdiri dari manusia, mesin, prosedur, dokumen, data atau elemen lain yang terorganisir dari elemen-elemen tersebut. Elemen sistem disamping berhubungan satu sama lain, juga berhubungan dengan lingkungannya untuk mencapai tujuan yang telah ditentukan sebelumnya. Terdapat beberapa definisi sistem yaitu :

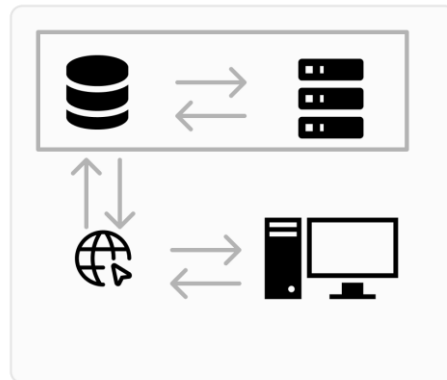
Gordon B. Davis (1984) “ Sebuah sistem terdiri dari bagian-bagian saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud”

Raymond Mcleod (2001) “ Sistem adalah himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk suatu kesatuan yang utuh dan terpadu “.

2.2.3 Website

Website adalah kumpulan halaman web yang memiliki sebuah domain yang mengandung informasi. Domain website adalah nama unik atau yang tidak duplikat yang dimiliki oleh sebuah institusi sehingga bisa di akses melalui internet, misalnya: ephi.id, yahoo.com, google.com dan lain-lain.

(Roosdianto et al., 2021)



gambar 2. 1 Arsitektur Web

(sumber : Wardani, Sirojul Hadi, Jian Budiarto,2021)

menunjukkan perancangan arsitektur aplikasi web yaitu dengan menghubungkan antara server dengan komputer pengguna melalui jaringan internet. Data sensor suhu dan kelembaban akan di kirim ke database serta di operasikan pada VPS. Tidak jauh beda dengan aplikasi web, hanya saja aplikasi mobile view dirancang agar dapat di operasikan pada aplikasi android, aplikasi mobile view juga melakukan request ke server untuk mendapatkan data suhu terakhir

2.2.4 Perpustakaan

Perpustakaan sekolah adalah perpustakaan yang berada pada lembaga pendidikan sekolah sebagai pusat sumber belajar mengajar. Atas pernyataan tersebut, beberapa tugas perpustakaan sekolah adalah mengelola data data perpustakaan, memberikan layanan pemimjaman buku, menyediakan tempat membaca dan mengembangkan agar mudah dikontrol dan diakses oleh anggota perpustakaan. Menyediakan sumber-sumber referensi untuk menambah pengetahuan siswa dan guru. Mendidik siswa agar dapat mencari dan mengumpulkan informasi (referensi) secara mandiri (Hartono., 2016).

Dalam UU No. 43 Tahun 2007 tentang Perpustakaan, disebutkan bahwa perpustakaan adalah lembaga yang mengelola koleksi karya tulis, karya cetak, dan/atau karya rekam secara profesional dengan sistem yang baku. Perpustakaan adalah kumpulan bahan cetak dan non cetak dan/atau sumber informasi pada

komputer yang disusun secara sistematis untuk kepentingan pemakai. Perpustakaan adalah ruangan yang digunakan untuk menyimpan buku dan publikasi lainnya yang biasanya disimpan dalam urutan tertentu untuk dibaca, bukan untuk dijual (S. Basuki, 2003)

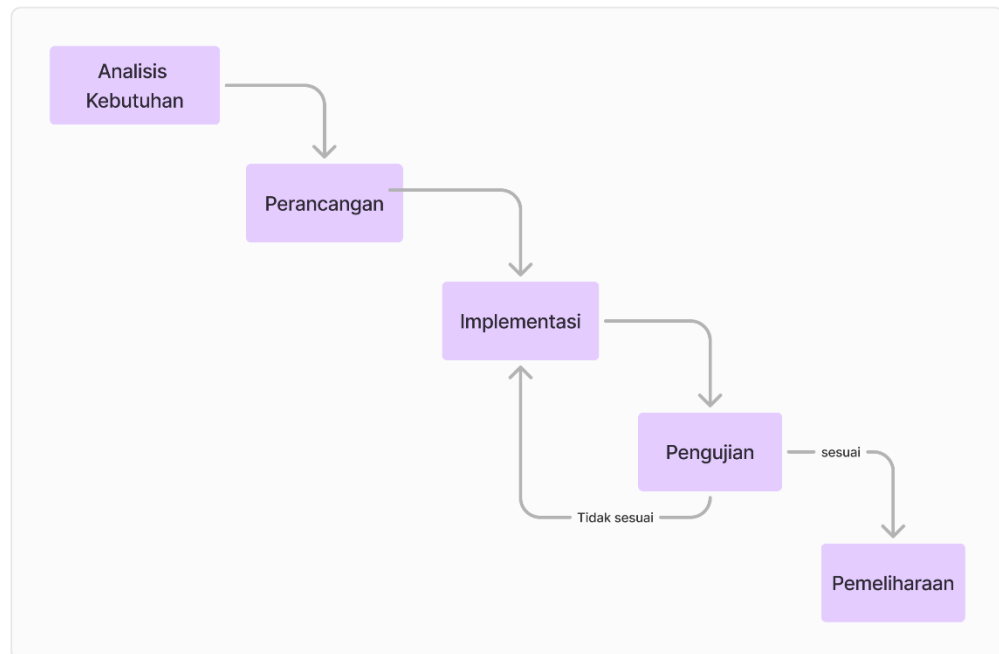
2.2.5 Basis data

Basis data terdiri dari 2 kata yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempang bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (Pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya. Yang diwujudkan dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya. (Suliyanti, 2019)

2.2.6 Metode SDLC *Waterfall*

System Development Life Cycle (SDLC) adalah proses mengembangkan atau mengubah suatu sistem dengan menggunakan model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem sebelumnya. Sementara itu, model *Waterfall* sering juga disebut sekuensial linier (Sequential linier) atau alur hidup klasik. Metode *Waterfall* adalah model yang digunakan untuk mengembangkan sistem informasi secara sistematis dan sekuensial. Salah satu metode SDLC, watefall, memiliki karakteristik bahwa setiap tahap pengerjaan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Ini berarti bahwa karena tidak ada pekerjaan paralel, fokus pada masing-masing tahap dapat dimaksimalkan. Metode *Waterfall* adalah urutan proses pengembangan perangkat lunak yang berurutan, tidak dapat dikembalikan, dan berurutan yang terdiri dari beberapa langkah yang harus dilakukan secara berurutan. Metode *Waterfall* ini memandang pengembangan perangkat lunak sebagai sebuah proses berurutan yang melibatkan tahap ke tahap selanjutnya, mulai dari analisis kebutuhan, perencanaan, desain, implementasi, pengujian, hingga pemeliharaan. Setiap tahapan harus diselesaikan sebelum dapat melanjutkan ke tahap berikutnya dan tidak dapat kembali ke tahap sebelumnya. Metode *Waterfall* digunakan untuk mengontrol dan mengelola risiko dalam pengembangan perangkat lunak dengan menguraikan setiap tahap menjadi tahapan yang terdefinisi dengan baik dan

mengikuti urutan yang telah ditentukan. Dengan metode ini, proyek pengembangan perangkat lunak dapat dijadwalkan dengan baik dan manajemen dapat memantau progres pengembangan dengan lebih mudah (Carolus Ningki, Noviyanti.P.2023)



gambar 2. 2 Waterfall

(Sumber : Muharir,M., dan Alamsyah,N.,2019)

Model *Waterfall* menyediakan pendekatan alur hidup sistem secara sekuensial atau terurut dimulai dari:

1) Analisis kebutuhan (*Analyst*)

Analisis kebutuhan adalah mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi dalam pelaksanaan. Tentunya juga mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan, yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan – perbaikannya.

2) Perancangan (*Design*)

Perancangan adalah persiapan rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dapat dibentuk yang berupa gambaran, perencanaan dan pembuatan sketsa atau pengaturan dari

beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

3) Implementasi (*implementation*)

Tahap implementasi, yaitu desain program diterjemahkan ke dalam kode – kode instruksi yang akan dijalankan komputer. Menerjemahkan hasil proses perancangan menjadi sebuah bentuk program komputer yang dimengerti oleh mesin komputer. Penerjemahan ke bahasa komputer menggunakan bahasa pemrograman dalam menjalankan instruksi yang disusun.

4) Pengujian (*Testing*)

Uji coba program merupakan elemen yang kritis dari SQA (*Software Quality Assurance*) dan mempresentasikan tinjauan ulang yang menyeluruh terhadap spesifikasi, desain dan pengkodean. Uji coba mempresentasikan ketidak normalan yang terjadi pada pengembangan software.

5) Pemeliharaan Sistem (*Maintenance*)

Pada tahap pemeliharaan sistem ini setelah aplikasi berhasil dan siap pakai maka harus melakukan pembenaran dan penyempurnaan software.

2.2.7 UML

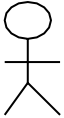


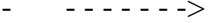
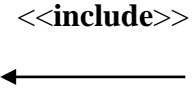
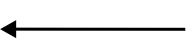
UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan *Software* berbasis *Object-Oriented*. UML sedniri juga memberikan standar penulisan sebuah sistem yang meliputi konsep bisnis proses, penulisan kelas – kelas dalam bahasa program yang spesifik, skema database, dan komponen – komponen yang diperlukan dalam sistem software. Beberapa jenis UML yang digunakan dalam perancangan aplikasi perpustakaan yaitu :

a. Use Case Diagram

”*Use Case Diagram* merupakan diagram yang harus dibuat pertama kali saat pemodelan pemograman perangkat lunak berorientasi objek

dilakukan. *Use Case Diagram* akan menggambarkan apa yang dikerjakan oleh aktor. Yang disebut aktor disini adalah pengguna aplikasi, sama seperti pembangunan perangkat lunak terstruktur saat membuat DFD, untuk menggambar *Use Case* mengacu pada proses sebelumnya, yaitu analisis kebutuhan pada Rekayasa Perangkat Lunak” (Maria, S. Dan Listiana, 2019).


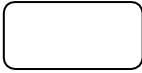
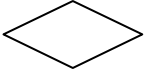


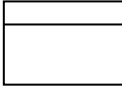
Tabel 2. 2 *Use Case Diagram*

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

b. Activity Diagram

Mengilustrasikan cara kerja diagram aktivitas Diagram ini berfungsi mirip dengan diagram alir dalam beberapa hal, namun perbedaan utama antara diagram ini dan notasi diagram alir adalah bahwa diagram ini mengizinkan tindakan parallel (Sukrianto, D. dan Agustina, S, 2018).

Tabel 2. 3 Activity Diagram



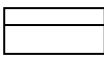

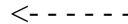
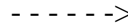
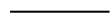
No	Simbol	Nama	Keterangan
1		Status Awal	Yaitu sebuah diagram aktivitas memiliki keadaan awal.
2		Aktivitas	Yaitu aktivitas yang dilakukan oleh sistem, aktivitas biasanya dimulai dengan kata kerja.
3		Percabangan / Decision	Yaitu cabang dimana ada pilihan aktivitas yang lebih dari satu tindakan.
4		Percabangan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

a. Class Diagram

Class diagram mempresentasikan struktur statis dari sistem dengan menggambarkan kelas-kelas yang ada dalam sistem tersebut beserta atribut-atribut yang menjadi properti kelas-kelas tersebut serta operasi-operasi atau metode-metode yang mewakili tindakan yang dapat dilakukan oleh setiap kelas. Atribut adalah variabel yang khusus dimiliki oleh sebuah kelas untuk menyimpan informasi yang relevan dengan kelas tersebut, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas untuk

melakukan tindakan-tindakan tertentu, yang dapat mengubah atribut-atribut atau menghasilkan output yang diperlukan dalam konteks sistem yang direpresentasikan oleh class diagram (Sukrianto, D. dan Agustina, S, 2018).

Tabel 2. 4 Class Diagram

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

2.2.8 XAMPP

XAMPP adalah singkatan dari X (salah satu dari empat sistem operasi). Ini adalah perangkat lunak gratis yang mendukung banyak sistem operasi dan merupakan kumpulan dari banyak aplikasi. Apache, MYSQL, PHPP, dan Perl adalah beberapa contohnya. Xampp adalah sebuah utilitas yang membundel beberapa produk perangkat lunak ke dalam satu paket. Sudah ada Apache (Web Server), Mysql (Database), PHP (server side scripting), Perl, FTP server, PhpMyAdmin, dan beberapa library tambahan lainnya yang sudah termasuk di dalam paket Xampp (Siregar, 2018).

2.2.9 Visual Studio Code

Visual Studio Code merupakan Editor kode pertama, dan cross-platform pertama, visual studio code merupakan Editor kode yang kuat dan cepat yang bagus untuk sehari-hari untuk pengkodean yang serius. Visual Studio Code memiliki fitur yang kaya dan dapat disesuaikan, termasuk penyorotan sintaksis, pemeriksaan kode, debugging, dan integrasi dengan berbagai ekstensi dan alat pengembangan yang berbeda. Salah satu keunggulan utama dari Visual Studio Code adalah kemampuannya untuk diintegrasikan dengan berbagai bahasa pemrograman dan kerangka kerja, sehingga cocok digunakan dalam pengembangan berbagai jenis aplikasi, mulai dari web development hingga pengembangan perangkat lunak. (The phpMyAdmin devel team Revision, 2020).

2.2.10 Figma

Figma adalah editor grafis vektor dan alat prototyping dengan berbasis web serta fitur offline tambahan yang diaktifkan oleh aplikasi desktop untuk Mac OS dan Windows. Aplikasi pendamping Figma Mirror untuk Android dan iOS memungkinkan untuk melihat prototype Figma pada perangkat seluler. Rangkaian fitur Figma berfokus pada penggunaan dalam antarmuka pengguna dan desain pengalaman pengguna dengan penekanan pada kolaborasi waktu nyata (real-time). Sederhananya, Figma adalah desain digital dan alat prototyping. Ini adalah aplikasi desain UI dan UX yang dapat Anda gunakan untuk membuat situs web, aplikasi, atau komponen antarmuka pengguna yang lebih kecil yang dapat diintegrasikan ke dalam proyek lain. Dengan alat berbasis vektor yang hidup di cloud, Figma memungkinkan para penggunanya untuk bekerja di mana saja dari

browser.Figma didukung oleh komunitas desainer dan pengembang yang kuat yang berbagi plugin untuk meningkatkan fungsionalitas dan mempercepat alur kerja. Siapa pun dapat berkontribusi dan berbagi. Figma juga digunakan oleh beberapa merek besar termasuk Slack, Twitter, Zoom, Dropbox, dan Walgreens. Nama-nama itu saja menunjukkan bahwa alat ini cukup kuat untuk memberi daya pada hampir semua proyek. (Agus Muhyidin et al., 2020)

2.2.11 Draw.io

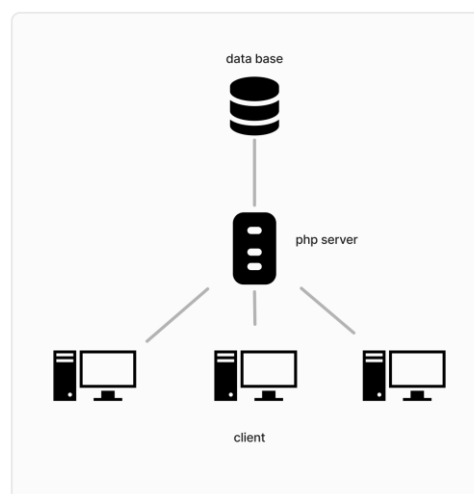
Draw.io adalah aplikasi diagram sumber terbuka dan gratis yang dapat Anda gunakan online di app.diagrams.net , dan offline (draw.io desktop).Sebagai aplikasi pembuatan diagram yang mengutamakan keamanan untuk tim, kami menyediakan fungsionalitas pembuatan diagram, dan Anda memilih tempat menyimpan data diagram. Ada banyak integrasi berbeda dengan platform dan aplikasi lain , termasuk Atlassian Confluence Cloud , Google Documents, GitHub, Microsoft Word.Editor diagram draw.io kami terlihat sama dan menawarkan fitur utama yang sama di mana pun Anda menggunakannya. Anda dapat membuka diagram yang dibuat dalam satu integrasi di integrasi lainnya. (Hendrawan1, Meisel2, and Sari3 2022)

2.2.12 Mysql

MySQL merupakan sebuah sistem manajemen basis data relasional (RDBMS) yang bersifat open-source. Ia dirancang untuk memberikan kinerja tinggi dan skalabilitas, mampu menangani beban kerja yang besar dan mendukung aplikasi yang memerlukan kecepatan tinggi dalam pengolahan data. Dalam konteks ilmiah, MySQL dapat diartikan sebagai sebuah perangkat lunak yang digunakan untuk mengelola dan menyimpan data dalam bentuk tabel yang terhubung melalui relasi atau hubungan. RDBMS seperti MySQL memungkinkan pengguna untuk mengatur, menyimpan, mengambil, dan memanipulasi data dengan menggunakan bahasa query seperti SQL (Structured Query Language). MySQL banyak digunakan dalam pengembangan perangkat lunak, aplikasi web, dan sistem informasi untuk menyimpan dan mengelola data secara efisien, MySQL banyak digunakan dalam pengembangan aplikasi web dan layanan online, serta menjadi bagian dari stack LAMP yang populer (Safitri & Nirmala, 2019).

2.2.13 PHP

PHP atau Hypertext Preprocessor adalah bahasa pemrograman script server-side yang digunakan untuk membangun atau mengembangkan aplikasi web. PHP sangat cocok untuk membuat halaman web dinamis dengan konektivitas ke berbagai basis data, salah satunya adalah MySQL yang sangat banyak digunakan dalam pengembangan web menggunakan PHP. PHP sangat mudah dipelajari dan berjalan dengan efisien di server-side. PHP (Hypertext PHP adalah bahasa pemrograman skrip yang dapat disisipkan ke dalam kode HTML. PHP telah merasionalisasi singkatannya menjadi 'PHP: Hypertext Preprocessor.' Bahasa pemrograman ini memiliki kemampuan yang mumpuni dalam mengembangkan situs web dinamis dan memungkinkan para pengembang untuk membangun sistem manajemen konten (Content Management System, CMS). Untuk menjalankan PHP, diperlukan infrastruktur berupa Web Server seperti Apache, PHP Interpreter, dan seringkali MySQL sebagai sistem basis data yang mendukung (Ferdinandus et al., n.d.).



gambar 2. 3 Arsitektur PHP

(sumber : Martinus Raditia Sigit Surendra, 2014)

Klien mengirimkan permintaan ke server web, yang kemudian menerima dan meresponsnya. Server web yang umumnya digunakan adalah Apache dan Nginx. Interpreter PHP bertugas menerjemahkan kode PHP menjadi kode mesin yang dapat dieksekusi oleh server web. Script PHP, yang ditulis dalam bahasa

pemrograman PHP, dapat menghasilkan teks HTML, mengakses database, dan menyelesaikan tugas-tugas lainnya. Data yang diperlukan oleh aplikasi web disimpan dalam database, yang bisa berupa MySQL, PostgreSQL, atau Oracle, dengan PHP memiliki kemampuan untuk terhubung ke berbagai jenis database tersebut. Selain itu, PHP juga mampu mengakses sumber daya sistem seperti file, direktori, dan proses.

2.2.14 Bootstrap

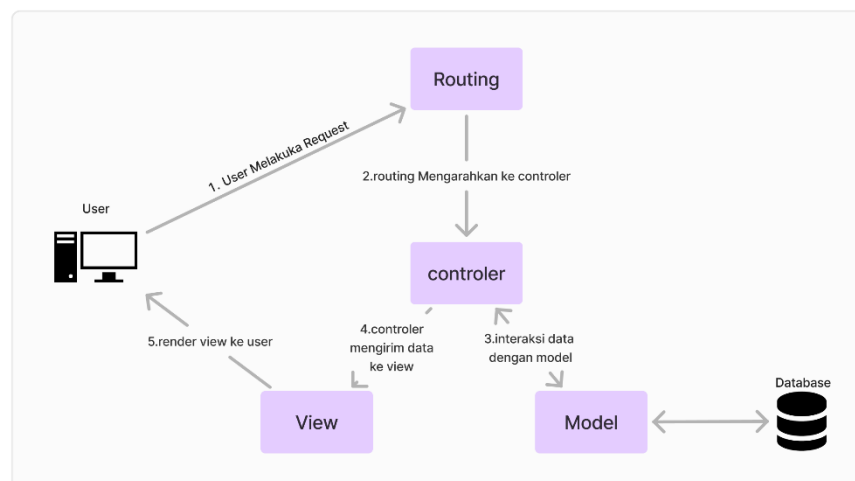
Bootstrap merupakan *Framework* untuk membangun desain web secara responsif. Bootstrap adalah kerangka kerja CSS yang sumber terbuka dan bebas untuk merancang situs web dan aplikasi web. Kerangka kerja ini berisi templat desain berbasis HTML dan CSS untuk tipografi, formulir, tombol, navigasi, dan komponen antarmuka lainnya, serta juga ekstensi opsional JavaScript. Dengan menggunakan Bootstrap, pengembang dapat dengan mudah membuat website yang responsif dan estetis tanpa harus menulis kode CSS atau JavaScript dari awal (Husein Alatas, 2013).

2.2.15 Laravel

Laravel merupakan *Framework* berbasis pemrograman PHP yang dapat digunakan dalam proses pengembangan website supaya lebih maksimal. Penggunaan *Laravel* akan membuat website yang dihasilkan lebih maksimal dan dinamis. Kehadiran *Framework Laravel* membuat pemrograman PHP menjadi maksimal dan lebih powerfull. Sebab, kehadiran *Framework Laravel* lebih baik dan fitur-fiturnya pun lebih ringan dibandingkan *Framework* lain. *Framework Laravel* ini menggunakan struktur Model View Controller atau MVC. MVC sendiri adalah model aplikasi pemisah antara tampilan komponen aplikasi dan data. Dengan menggunakan model MVC, para pengguna *Laravel* jadi lebih mudah memahami dan mempelajari *Laravel*. Dan juga proses pembuatan aplikasi website tentu akan menjadi lebih cepat.

Laravel adalah kerangka kerja yang dibangun dengan basis bahasa pemrograman PHP. *Laravel* memiliki komunitas dan pengguna yang terus berkembang hingga sekarang, *Laravel* sendiri tidak terlepas dari konsep MVC (Model, View, Controller) dimana merupakan sebuah konsep modern yang memisahkan bagian tampilan (front-end) dan juga bagian pengelolaan data atau

biasa disebut controller (back-end). Ada pula beberapa kelebihan yang dimiliki oleh *Laravel* diantaranya yaitu, *Laravel* memiliki fitur untuk mengatur rute atau disebut dengan routing, fitur ini akan mengatur rute request yang ditentukan berdasarkan kebutuhan nya. Selanjutnya *Laravel* juga memiliki query builder dan ORM yang dapat memberikan kemudahan dalam melakukan operasi database dan mendukung berbagai database. lalu, *Laravel* juga sudah terhubung dengan composer yang tentu akan mempermudah pengembang ketika membutuhkan modul atau library tambahan dan composer dapat melakukan pembaruan pada modul tersebut. *Laravel* juga dikenal memiliki komunitas yang sangat aktif sehingga *Laravel* sendiri memiliki banyak library tambahan yang dapat di install dan disesuaikan dengan kebutuhan aplikasi. Selain itu, masih banyak fitur lain yang dapat mempermudah proses perancangan aplikasi . (Adivia Gilang Prakarsa, Ari Sujarwo.2021),



gambar 2. 4 Arsitektur Laravel

(Sumber : Oky Dwi Arianto, Yerymia Alfa Susetyo.2022).

Ketika berinteraksi dengan *Laravel*, browser akan mengirimkan request kepada web server kemudian akan diteruskan ke sistem routing *Laravel*. Router *Laravel* akan memproses request kemudian mengalihkannya ke masing-masing class dan method sesuai dengan request url yang telah didefinisikan sebelumnya. Oleh Controller, terjadi komunikasi dengan model jika diperlukan data yang berhubungan dengan database. Dalam beberapa kasus, controller akan

melakukan render view yang nantinya akan dikonversi menjadi HTML dan dikirim kembali ke browser.

2.2.16 Java Script

JavaScript adalah bahasa skrip (bahasa yang kodenya ditulis menggunakan teks biasa) yang ditempelkan pada dokumen HTML dan diproses pada sisi klien”. Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi lebih luas. Selain itu, dengan menggunakan JavaScript juga dimungkinkan untuk mengimplementasikan tugas yang bersifat interaktif tanpa berhubungan dengan server. (Kadir, A. 2009).

2.2.17 HTML

Dalam jurnal Fitri Ayu dan Nia Permata Sari (ISSN: 2549-0222), disebutkan bahwa HTML digunakan untuk membuat struktur dasar halaman web dengan mendefinisikan elemen-elemen seperti teks, gambar, tautan, dan lainnya menggunakan tag-tag yang dikelilingi oleh tanda kurung sudut (< >). HTML bekerja secara hierarkis di mana elemen-elemen dapat dikelompokkan menjadi bagian-bagian yang lebih besar atau kecil sesuai dengan struktur logis halaman web. Selain itu, HTML mendukung atribut-atribut untuk memberikan informasi tambahan tentang elemen, seperti warna latar belakang atau tautan ke halaman lain. Sebagai teknologi dasar dalam pengembangan web, HTML menjadi tulang punggung dari struktur konten setiap halaman web di internet, memungkinkan pembangunan halaman web yang konsisten diakses, diindeks, dan diinterpretasikan oleh berbagai perangkat dan peramban web (Hidayatullah dan Kawistara, 2015).

2.2.18 CSS

CSS (*Cascading Style Sheet*) adalah bahasa style sheet yang mengatur format tampilan halaman web yang ditulis dengan penanda markah songage. CSS sering digunakan untuk membuat halaman HTML dan XHTML, tetapi sekarang dapat digunakan untuk dokumen XML apa pun, termasuk SVG, XUL, dan bahkan Android. CSS merupakan bahasa pemrograman yang digunakan untuk mengatur tampilan dan gaya elemen-elemen pada halaman web. Dengan menggunakan CSS, pengembang web dapat mengontrol warna, font, ukuran, letak, dan berbagai properti visual lainnya dari elemen HTML pada halaman web. CSS memisahkan

presentasi dari struktur konten pada halaman web, memungkinkan pengembang untuk membuat tampilan yang konsisten dan menarik secara visual. CSS bekerja dengan prinsip kaskade, yang berarti gaya-gaya yang dideklarasikan lebih bawah dalam kode CSS akan menggantikan gaya-gaya yang dideklarasikan sebelumnya. Selain itu, CSS juga mendukung konsep pewarisan (*inheritance*), yang memungkinkan elemen-elemen dalam hierarki HTML untuk mewarisi gaya-gaya dari elemen-elemen induknya. CSS dapat disertakan dalam halaman web secara internal (*inline*), eksternal (melalui file terpisah), atau bahkan menggunakan prinsip pemrograman berbasis kerangka kerja (*Framework*) seperti Bootstrap atau Tailwind CSS. Dengan kemampuan untuk mengontrol tampilan dan gaya elemen-elemen pada halaman web, CSS merupakan salah satu teknologi kunci dalam pengembangan web modern. (Safitri & Nirmala, 2019).

2.2.19 Metode pengujian penelitian

Ada beberapa metode pengujian yang dapat digunakan untuk menguji hasil penelitian tentang sistem manajemen perpustakaan , di antaranya:

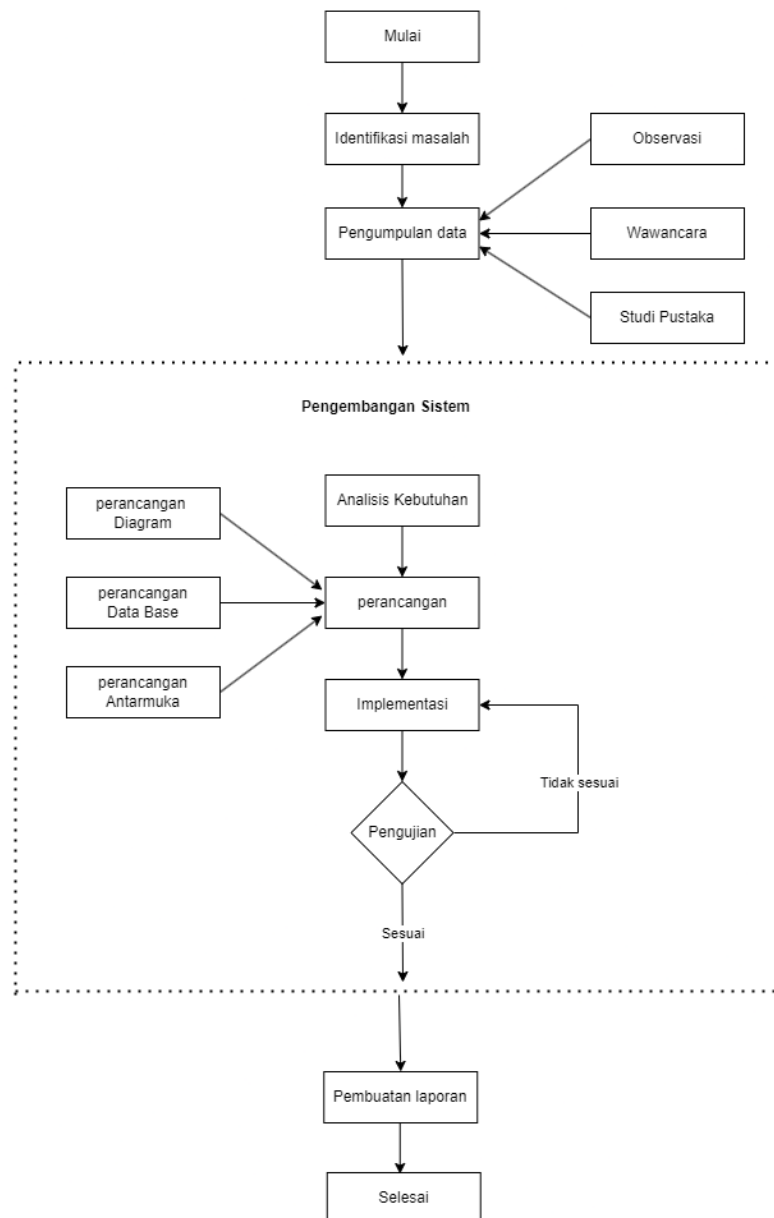
- 1) Uji Validitas Metode ini digunakan untuk mengukur seberapa validnya instrumen penelitian yang digunakan dalam penelitian. Ada beberapa jenis uji validitas, antara lain uji validitas isi, uji validitas konstruk, dan uji validitas kriteria.
- 2) Uji Reliabilitas Metode ini digunakan untuk mengukur seberapa konsisten hasil pengukuran yang diperoleh dari instrumen penelitian. Ada beberapa jenis uji reliabilitas, antara lain uji reliabilitas internal, uji reliabilitas eksternal, dan uji reliabilitas test-retest berupa kuisisioner. Pengujian reliabilitas pada kuisisioner mengukur sejauh mana alat ukur tersebut konsisten dalam mengukur apa yang seharusnya diukur.
- 3) Analisis Perbandingan Metode ini digunakan untuk membandingkan hasil penelitian dengan penelitian sebelumnya atau dengan standar yang telah ditetapkan. Analisis perbandingan dapat membantu dalam memperoleh pemahaman yang lebih baik tentang hasil penelitian dan mengevaluasi keefektifan dari strategi pengelolaan persediaan barang yang telah diimplementasikan

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Berfikir

Kerangka pemikiran adalah dasar pemikiran dari penelitian yang disusun dari dasar fakta-fakta, observasi, dan kajian kepustakaan. Berikut adalah kerangka pikir yang merupakan langkah-langkah yang akan dilakukan dalam penelitian ini:



gambar 3. 1 kerangka berfikir

3.2 Deskripsi

3.1.1 Identifikasi Masalah

Langkah pertama yang dilakukan dalam pembuatan aplikasi perpustakaan adalah dengan mengidentifikasi masalah atau menetapkan perumusan masalah. Dalam hal ini penulis meminta izin kepada pihak sekolah untuk melakukan penelitian, serta melakukan observasi dan wawancara guna memahami masalah yang dihadapi oleh pengelola perpustakaan maupun murid murid. Permasalahan yang ditemukan di perpustakaan tersebut adalah pengelolaan data buku yang masih manual, yang mana seringkali menyebabkan ketidak akuratan data, sering terjadinya kehilangan atau kerusakan buku dan kesulitan dalam pembuatan laporan secara cepat

3.1.2 Pengumpulan Data

Dalam penelitian ini penulis menggunakan beberapa metode dalam pengumpulan data yang dibutuhkan dalam pembuatan aplikasi perpustakaan berbasis web diantaranya sebagai berikut :

1) Observasi

Penulis mengamati langsung sistem perpustakaan yang ada di SMP KP 2 Majalaya untuk mendapatkan informasi yang diperlukan dan akurat dalam membuat aplikasi Pengelolaan perpustakaan.

2) Wawancara (Interview)

Penulis melakukan proses tanya jawab terhadap pihak yang bersangkutan di SMP KP 2 Majalaya mengenai sistem kerja dan proses Pengelolaan perpustakaan.

3) Studi Pustaka

Penulis mengumpulkan data dari beberapa jurnal, sumber yang bersangkutan dengan aplikasi Pengelolaan perpustakaan berbasis web di SMP KP 2 Majalaya

3.1.3 Analisis Kebutuhan

Pada tahap ini penulis melakukan analisis kebutuhan terkait aplikasi yang akan dibangun, meliputi kebutuhan fungsional dan non-fungsional, serta pemahaman tentang proses pengelolaan perpustakaan. Berikut adalah analisis yang dilakukan

1) Analisis Sistem yang berjalan

Penulis melakukan analisis terhadap sistem yang berjalan di perpustakaan SMP KP 2 Majalaya, Saat ini sistem yang diterapkan masih menggunakan cara konvensional, yaitu pencatatan atau pengelolaan data buku masih dilakukan secara manual. Yang mana seringkali menyebabkan masalah, seperti keakuratan data yang rendah, sering terjadi nya kehilangan atau kerusakan buku dan kesulitan dalam pembuatan laporan secara cepat

2) Analisis Kebutuhan Sistem

a. Kebutuhan Fungsional

1. Data buku : fitur ini akan berfokus pada penambahan buku bila ada buku baru ,penghapusan buku untuk buku buku yang tidak layak pakai lalu menampilkan buku buku yang ada di perpustakaan dan bisa di pinjam
2. Data anggota : fitur ini berfokus dalam penambahan user dan juga penghapusan user serta menampilkan user yang dapat meminjam buku
3. Transaksi : fitur ini memungkinkan user untuk meminjam atau mengembalikan buku serta dapat melihat daftar transaksi yang telah terjadi
4. Laporan : fitur ini yang mana memungkinkan pengguna untuk melihat atau mencetak laporan mengenai perpustakaan mulai dari ada berapa buku yang masuk maupun rusak, ada berapa anggota baru yang masuk dan ada berapa yang keluar serta dapat mengetahui siswa mana yang masih belum mengembalikan buku yang di pinjamnya

b. Kebutuhan non fungsional

1. Perangkat Keras

Perangkat keras yang digunakan memiliki spesifikasi sebagai berikut:

Table III.1 kebutuhan perangkat keras

No	Item	Spesifikasi
1	Processor	Kecepatan 2.10GHz
2	Harddisk	500 GB

3	Memory	4 GB
4	VGA	500 Mb
5	Monitor	Resolusi 1366x 768

2. Perangkat Lunak

Perangkat lunak yang digunakan dapat dilihat pada table berikut :

Table III.1 kebutuhan perangkat keras

Sistem Operasi	Window 10
Code editor	Visual Studio code
Web Framework	Laravel
Database	Mysql
Diagram	Draw.io
UI Design	Figma

3.1.4 Perancangan

1. Perancangan Diagram

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

1) *Use case diagram*

digunakan untuk menggambarkan secara umum hubungan antara aktor dan kasus pengguna dalam suatu sistem aplikasi. Diagram ini membantuk dalam pemahaman tentang interaksi antara aktor (entitas yang berinteraksi dengan sistem) dan berbagai fungsi (*use case*) yang ada dalam sistem tersebut. Dalam konteks ini, terdapat dua aktor yaitu Admin dan juga user.admin dapat menambahkan atau menghapus anggota,admin dapat menghapus atau menambahkan buku admin dapat melihat transaksi serta admin dapat mencetak laporan.sedangkan untuk user hanya bisa mengakses data buku untuk melihat

ketersediaan buku yang ada di perpustakaan berikut uraian dari usecase yang akan di buat

- a) Login
- b) Dashboard
- c) Buku masuk
- d) Buku keluar
- e) Data buku
- f) Data anggota
- g) Data kelas
- h) Laporan

2) *Activity diagram*

Activity diagram yang akan dibuat adalah sebagai berikut

- a) *Activity diagram* tambah anggota
- b) *Activity diagram* hapus anggota
- c) *Activity diagram* daftar anggota
- d) *Activity diagram* tambah buku
- e) *Activity diagram* hapus buku
- f) *Activity diagram* daftar buku
- g) *Activity diagram* pinjam buku
- h) *Activity diagram* kembalikan buku
- i) *Activity diagram* daftar transaksi
- j) *Activity diagram* Laporan

2. Perancangan data base

Pada tahapan perancangan database menggunakan Entity Relationship Diagram (ERD), dengan daftar entitas yang akan dibuat adalah sebagai berikut:

- a) Buku
- b) Peminjaman
- c) Anggota
- d) Buku keluar
- e) Buku masuk
- f) User
- g) kelas

3. Perancangan Antar muka

Pada tahapan perancangan antarmuka, desain aplikasi dibuat menggunakan Figma, dengan desain antarmuka yang dapat mudah di pahami oleh pengguna dan tentunya dengan memperhatikan aspek *user experience*

3.1.5 Implementasi

Tahap ini melibatkan mengubah semua desain yang telah dibuat menjadi kode program sesuai dengan spesifikasi yang telah ditetapkan dalam tahap sebelumnya. Aplikasi akan dibuat menggunakan bahasa pemrograman PHP dengan *Laravel*, serta database yang digunakan adalah *Mysql*.

3.1.6 Pengujian

Pada tahap ini dilakukan pengujian aplikasi yang telah dibuat untuk memastikan bahwa setiap fitur berfungsi sesuai dengan baik dan tidak ada bug atau kesalahan. Jika terdapat masalah, perbaikan dilakukan kembali pada tahap implementasi hingga semua berjalan dengan baik sesuai yang diharapkan. Pengujian sistem menggunakan metode *Black Box Testing*,

3.1.7 Pembuatan Laporan

Tahapan terakhir adalah penyusunan laporan hasil penelitian yaitu Menjelaskan secara detail mulai dari langkah-langkah yang dijalankan dalam penelitian hingga hasil yang diperoleh dalam penelitian ini. Pembuatan Laporan ini bertujuan untuk secara sistematis menyampaikan informasi kepada pembaca tentang isi dari penelitian yang telah dilakukan.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis

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

4.1.1 Analisis masalah

Berdasarkan observasi dan wawancara dengan narasumber didapati bahwa perpustakaan SMP KP 2 Majalaya memiliki sebuah permasalahan yaitu belum adanya sistem atau aplikasi untuk mengelola perpustakaan seperti dalam pencatatan inventaris buku, transaksi peminjaman, pengembalian, pembuatan laporan dan lainnya. karena sistem manual yang rentan kesalahan, sehingga data menjadi tidak akurat dan pelacakan inventaris sulit dilakukan. Hal ini berakibat pada keterlambatan pelaporan, informasi yang tidak akurat, pengambilan keputusan yang tidak tepat, dan penurunan minat baca siswa.

4.1.2 Analisis Software

Berdasarkan analisis kebutuhan maka untuk memenuhi kebutuhan dalam pengembangan aplikasi manajemen perpustakaan memerlukan software sebagai alat penunjang pengembangan aplikasi manajemen perpustakaan berbasis web. Adapun software yang dibutuhkan sebagai alat penunjang pada penelitian ini adalah sebagai berikut:

1. Sistem operasi windows 10
2. *Framework Laravel* digunakan untuk mengembangkan sistem dengan mudah dan cepat.
3. Visual Studio Code sebagai text editor digunakan untuk melakukan pengkodean dalam pembuatan Aplikasi manajemen perpustakaan
4. XAMPP digunakan untuk web server
5. Web browser digunakan untuk melihat tampilan sistem yang sudah dibuat
6. Draw.io digunakan untuk membuat/merancang diagram UML
7. Figma digunakan untuk membuat rancangan antarmuka Aplikasi manajemen perpustakaan

4.1.3 Analisis Pengguna

Analisis pengguna adalah yang berkaitan dengan yang akan memakai aplikasi manajemen perpustakaan berbasis web. Terdapat satu pengguna utama yaitu bagian pustakawan atau pengelola perpustakaan sebagai admin. Bertanggung jawab atas pengelolaan perpustakaan di sekolah dengan akses penuh terhadap sistem. Yang mana dapat menambah, mengedit, dan menghapus data buku dan data anggota serta dapat mencatat peminjaman serta pengembalian buku, memantau status atau stok buku secara real-time, serta membantu dalam membuat laporan.

4.1.4 Desain antarmuka

Desain antarmuka merupakan salah satu hal yang penting dalam sebuah aplikasi, desain antarmuka memiliki pengaruh besar terhadap kenyamanan pengguna saat menggunakannya. Pembuatan desain antarmuka pada aplikasi manajemen perpustakaan dibuat dengan desain antarmuka yang dapat mudah dipahami oleh pengguna dan tentunya dengan memperhatikan aspek *user experience*

Berikut adalah desain antarmuka dari aplikasi manajemen perpustakaan, meliputi:

1. Desain antarmuka halaman login
2. Desain antarmuka halaman dashboard
3. Desain antarmuka halaman buku
4. Desain antarmuka halaman kelas
5. Desain antarmuka halaman anggota
6. Desain antarmuka halaman peminjaman
7. Desain antarmuka halaman buku masuk
8. Desain antarmuka halaman buku keluar
9. Desain antarmuka halaman laporan

4.1.5 Fitur-Fitur

Fitur-fitur yang digunakan dalam aplikasi manajemen perpustakaan ini dibuat agar user dapat dengan mudah untuk mendapatkan informasi dan mengetahui siapa saja yg meminjam buku dan apakah buku yang di pinjam sudah dikembalikan atau belum serta dapat melihat siapa saja yang dapat meminjam buku dan buku apa saja

yang tersedia. Berikut adalah fitur-fitur yang disediakan aplikasi manajemen perpustakaan:

1. halaman dashboard

Berfungsi untuk menampilkan informasi informasi untuk pembuatan laporan seperti jumlah anggota, jumlah buku, jumlah peminjam laki laki, jumlah peminjam Perempuan jumlah buku masuk, jumlah buku keluar

2. halaman kelas

berfungsi untuk menambahkan atau membuat kelas sesuai dengan kelas yang ada fitur ini dibuat dikarenakan jumlah murid yang daftar ke SMP KP 2 Majalaya berbeda beda jadi sering terjadi perbedaan jumlah kelas tiap Angkatan

3. halaman Buku

berfungsi untuk melihat buku buku yang ada di perpustakaan termasuk stoknya bisa juga digunakan untuk menambahkan stok untuk buku yang sudah terdaftar.

4. halaman buku masuk

Berfungsi untuk menambahkan buku baru Ketika ada buku buku baru yang masuk ke perpustakaan

5. halaman buku keluar

Berfungsi untuk menghapus buku yang sudah tidak layak pakai

6. halaman anggota

Berfungsi untuk menambahkan atau pun menghapus anggota yang bisa meminjam buku.

7. halaman peminjaman

Berfungsi untuk meminjam atau mengembalikan buku serta bisa digunakan untuk melihat transaksi pinjam Kembali buku yang dilakukan

8. Halaman laporan

Berfungsi untuk mencetak laporan yang berisi informasi yang dibutuhkan

4.1.6 Analisis Data

Untuk mendukung pembuatan aplikasi manajemen perpustakaan, penulis memerlukan beberapa data sebagai bahan penelitian dan pembuatan aplikasi.

Adapun data-data yang digunakan dan diolah pada aplikasi manajemen perpustakaan ini adalah sebagai berikut:

1. Data transaksi

Analisis data transaksi akan memberikan detail mengenai peminjaman buku, termasuk siapa yang meminjam, barang apa yang dipinjam, tanggal peminjaman, dan tanggal pengembalian. Dengan analisis ini, pihak pengelola perpustakaan dapat mengetahui berapa jumlah peminjam laki laki dan berapa jumlah peminjam Perempuan dan siapa saja siswa yang meminjam buku akan tetapi tidak mengembalikan buku.

2. Data buku

Analisis data buku akan menghasilkan informasi mengenai jumlah buku yang tersedia. Informasi ini ditujukan agar pengelola perpustakaan dapat memberikan informasi yang akurat mengenai ketersediaan buku.

3. Data anggota

Analisis data anggota akan menghasilkan informasi mengenai siapa saja yang dapat meminjam buku di perpustakaan serta memberikan informasi mengenai anggota seperti kelas berapa dan beralamat dimana. Informasi ini ditujukan agar pengelola perpustakaan dapat memastikan siapa saja yang bisa meminjam buku dan apabila ada anggota yang belum mengembalikan buku dapat mengetahui informasi mengenai peminjam buku tersebut.

4.1.7 Analisis Biaya

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

tabel 4. 1 Analisis Biaya

No	Jenis Kebutuhan	Biaya
1	Biaya ATK	Rp. 2.550.000
2	Internet	Rp. 3.000.000
3	Analisis	Rp. 2.500.000
4	<i>Programming</i>	Rp. 5.500.000
Jumlah		Rp. 13.550.000

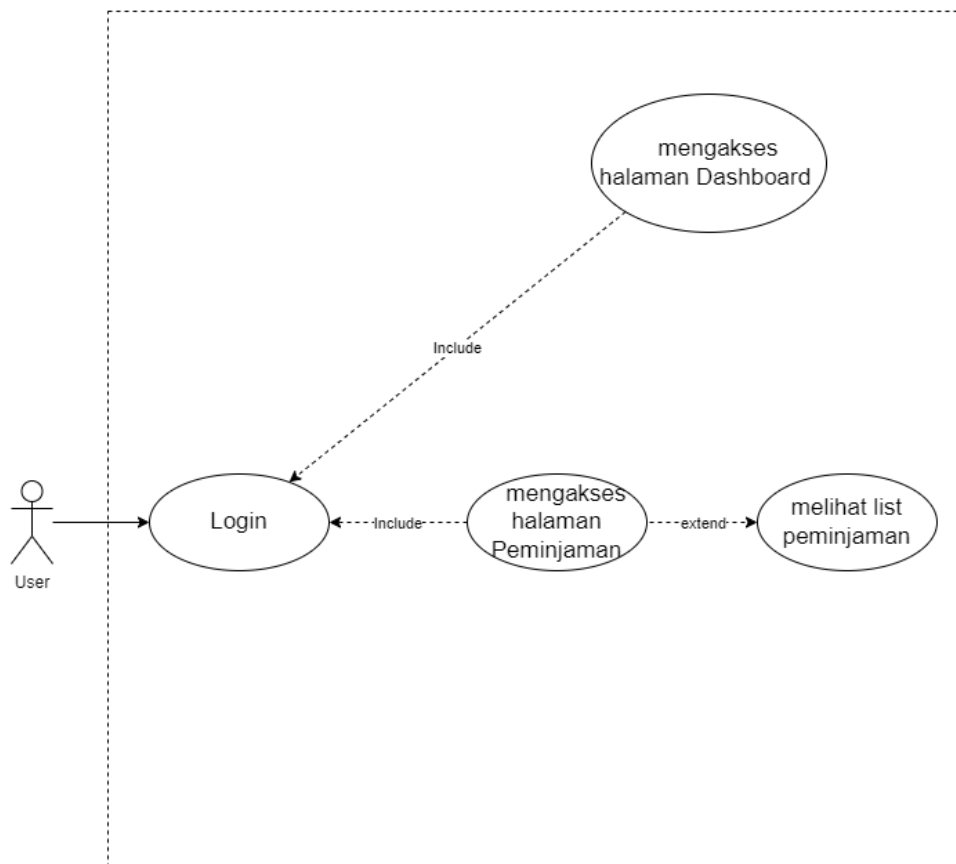
4.2 Perancangan

Membuat aplikasi manajemen perpustakaan yang efektif membutuhkan perencanaan matang untuk memastikan kemudahan penggunaan dan fungsionalitas yang optimal.. Perancangan ini melibatkan pembuatan perangkat lunak dalam bentuk Unified Modelling Language (UML) Perancangan meliputi perencanaan berbagai aspek dari perangkat lunak, termasuk desain, arsitektur perangkat lunak, dan struktur database. Perancangan yang akan dilakukan yaitu menggunakan *Unified Modeling Language* (UML), atau *Entity Relationship Diagram* (ERD).

1. Use Case Diagram

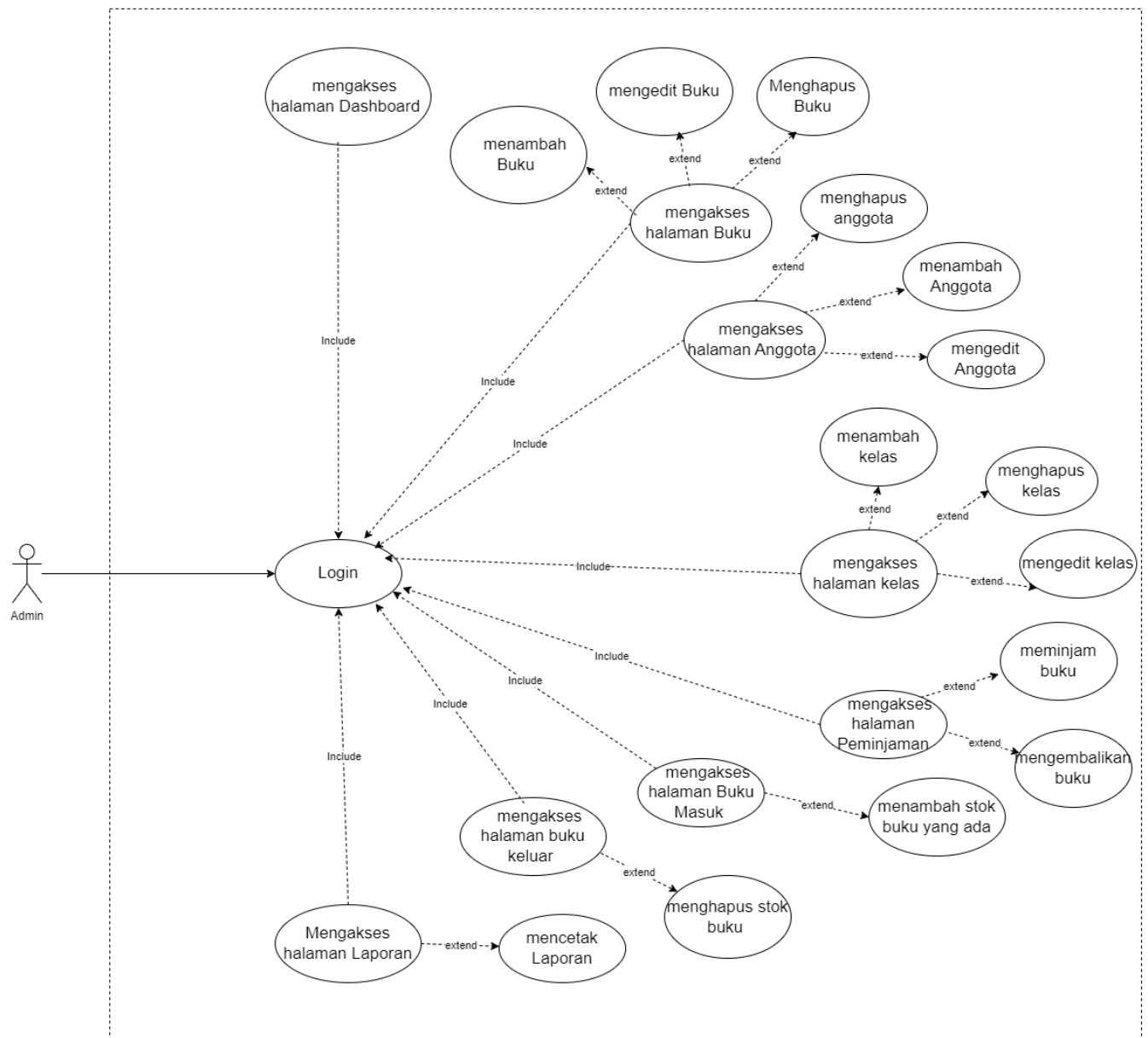
Use Case Diagram menggambarkan fungsi yang ada pada sistem tersebut dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut ini adalah penjelasan tentang Use Case diagram aplikasi manajemen perpustakaan berbasis web:

1) Use case diagram User



Gambar 4. 1 Use case User

2) Use case Diagram Admin



Gambar 4. 2 Use Case Diagram

1. Deskripsi Actor

tabel 4. 2 Deskripsi Actor

Actor	Deskripsi
Admin	Actor merupakan pengguna yang dapat melakukan berbagai operasi mulai dari login yang Dimana setelah login akan di arahkan ke 7 menu utama yaitu ada

	<p>dashboard yang Dimana berisikan informasi untuk pembuatan laporan seperti jumlah anggota, jumlah buku, jumlah peminjam laki laki, jumlah peminjam Perempuan jumlah buku masuk, jumlah buku keluar kemudian ada halaman kelas yang Dimana didalamnya bisa membuat kelas menghapus kelas ataupun mengedit kelas ,kemudian ada halaman buku yang didalamnya bisa menambahkan buku baru, menghapus buku ataupun mengedit buku, kemudian ada halaman anggota yang didalamnya bisa menambahkan anggota menghapus anggota dan mengedit anggota, kemudian ada menu peminjaman yang digunakan untuk meminjam ataupun mengembalikan buku, kemudian ada menu buku masuk dan buku keluar yang digunakan untuk menambah atau menghapus stok buku yang ada. dan yang terakhir adalah pencetakan laporan</p>
User	<p>User merupakan pengguna yang memiliki akses terbatas hanya bisa melihat data peminjaman yang dilakukan oleh dirinya</p>

2. Deskripsi Use Case Diagram

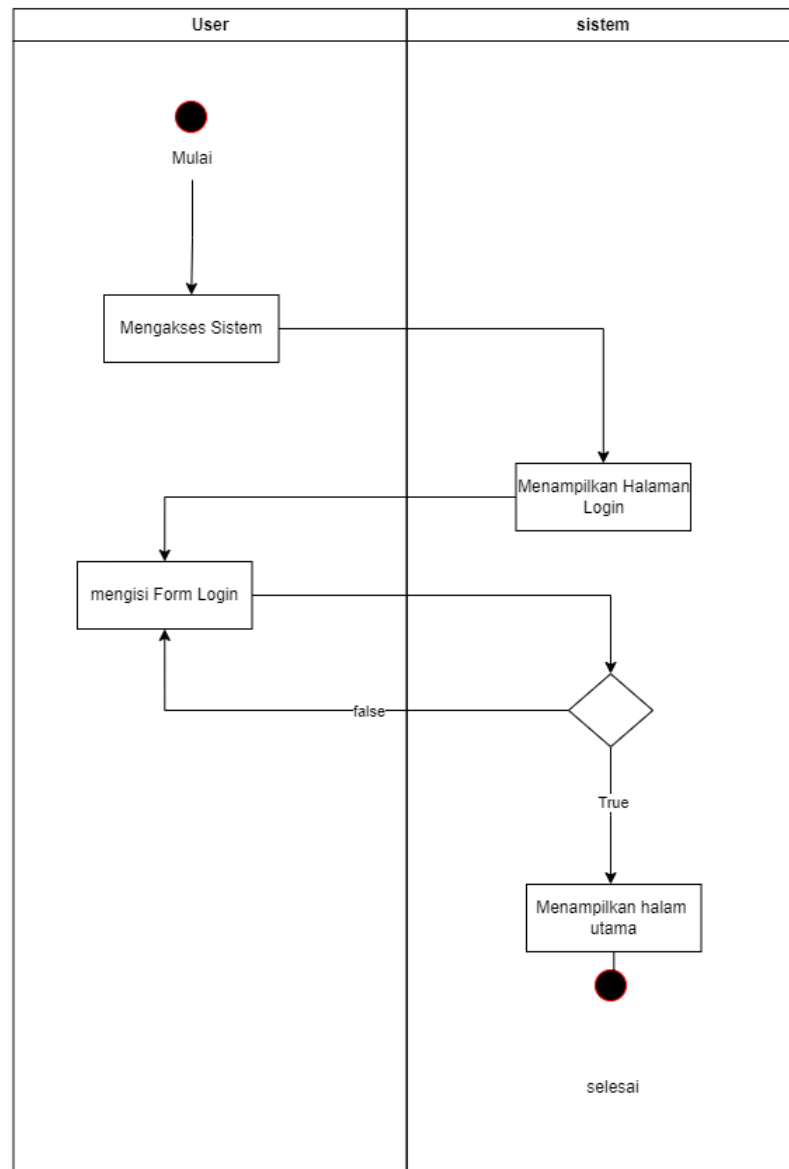
tabel 4. 3 Deskripsi Use Case Diagram

Use Case	Deskripsi
Login user dan admin	Proses awal yang dilakukan sebelum mengakses aplikasi dengan menginputkan <i>username</i> dan <i>password</i> dari pengguna.
Dashboard	Menu utama yang menampilkan tampilan dasar dari aplikasi manajemen perpustakaan berisikan informasi informasi untuk pembuatan laporan

kelas	Proses yang digunakan untuk mengakses data kelas yang Dimana didalamnya bisa melakukan proses tambah kelas,hapus kelas,dan edit kelas
Anggota	Proses pengelolaan data anggota yang bisa meminjam buku di perpustakaan yang mana digunakan untuk menambah,menghapus,mengedit atau melihat list data anggota
Buku	Proses yang digunakan untuk mengelola data buku termasuk melihat list buku menambah buku baru ataupun menghapus buku yang ada
Peminjaman	Proses yang digunakan untuk melakukan transaksi peminjaman buku dan pengembalian buku
Buku masuk	Proses yang digunakan untuk menambah stok buku yang telah terdaftar atau telah ada di perpustakaan
Buku keluar	Proses yang digunakan untuk mengurangi stok buku yang ada
Laporan	Proses yang digunakan untuk pencetakan laporan laporan yang dibutuhkan oleh petugas perpustakaan
Peminjaman User	Proses yang digunakan untuk melihat list peminjaman user

2. Activity Diagram

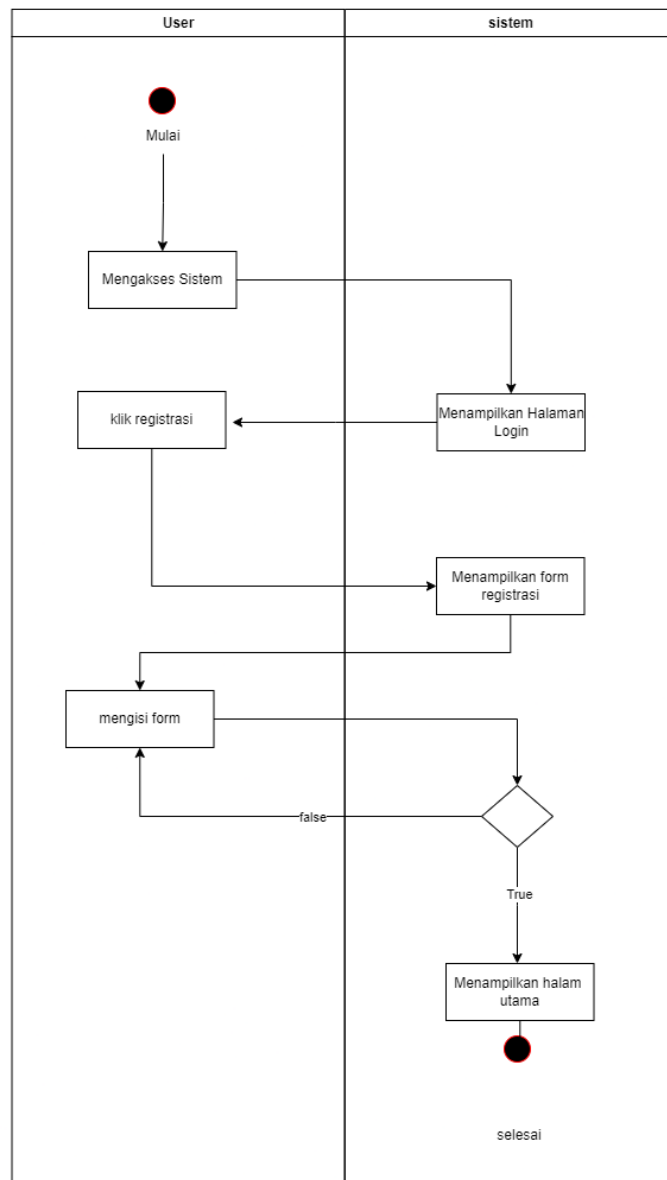
Activity Diagram merupakan diagram yang Menunjukkan alur proses bisnis aplikasi secara detail, termasuk langkah-langkah yang terlibat dalam setiap aktivitas, seperti peminjaman buku, pengembalian buku, dan pengelolaan data anggota. Diagram ini membantu memahami urutan proses dan mengidentifikasi potensi inefisiensi. Berikut adalah *Activity* diagram pada aplikasi manajemen perpustakaan.

1) *Activity Diagram Login*

Gambar 4. 3 Activity Diagram Login

Pada gambar 4.3 menjelaskan bagaimana alur Ketika user atau admin ingin mengakses aplikasi manajemen perpustakaan. Pada *Activity* diagram login diawali dengan mengakses sistem kemudian sistem menampilkan form login yang harus di isi oleh user kemudian form yang di isi user akan di verifikasi oleh sistem apabila login berhasil akan diarahkan ke halaman utama atau dashboard

2) Activity Diagram Registrasi

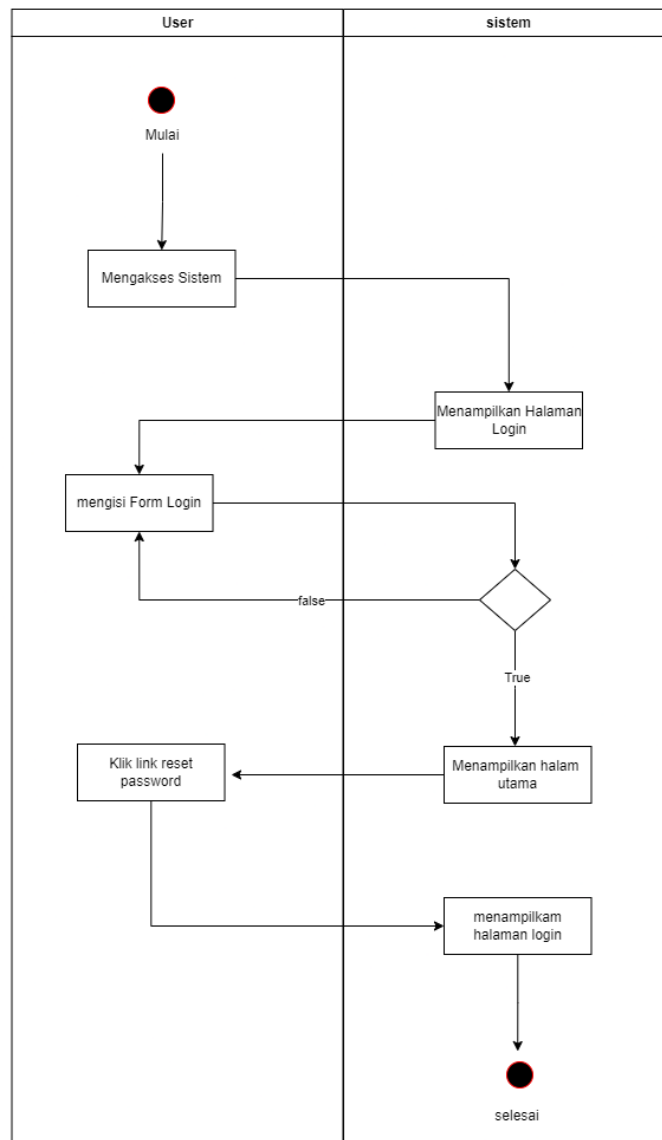


Gambar 4. 4 Activity Diagram Registrasi

Pada gambar 4.4 menjelaskan bagaimana alur Ketika user atau admin ingin mengakses aplikasi manajemen perpustakaan. Pada Activity diagram registrasi diawali dengan user yang mengakses sistem kemudian user akan di arahkan kehalaman login dan user mengklik tombol registrasi kemudian user mengisi form registrasi dan sistem akan memvalidasi apabila validasi gagal akan di kembalikan ke

halaman form registrasi sedangkan apabila berhasil akan diarahkan ke halaman login

3) *Activity Diagram Lupa Password*

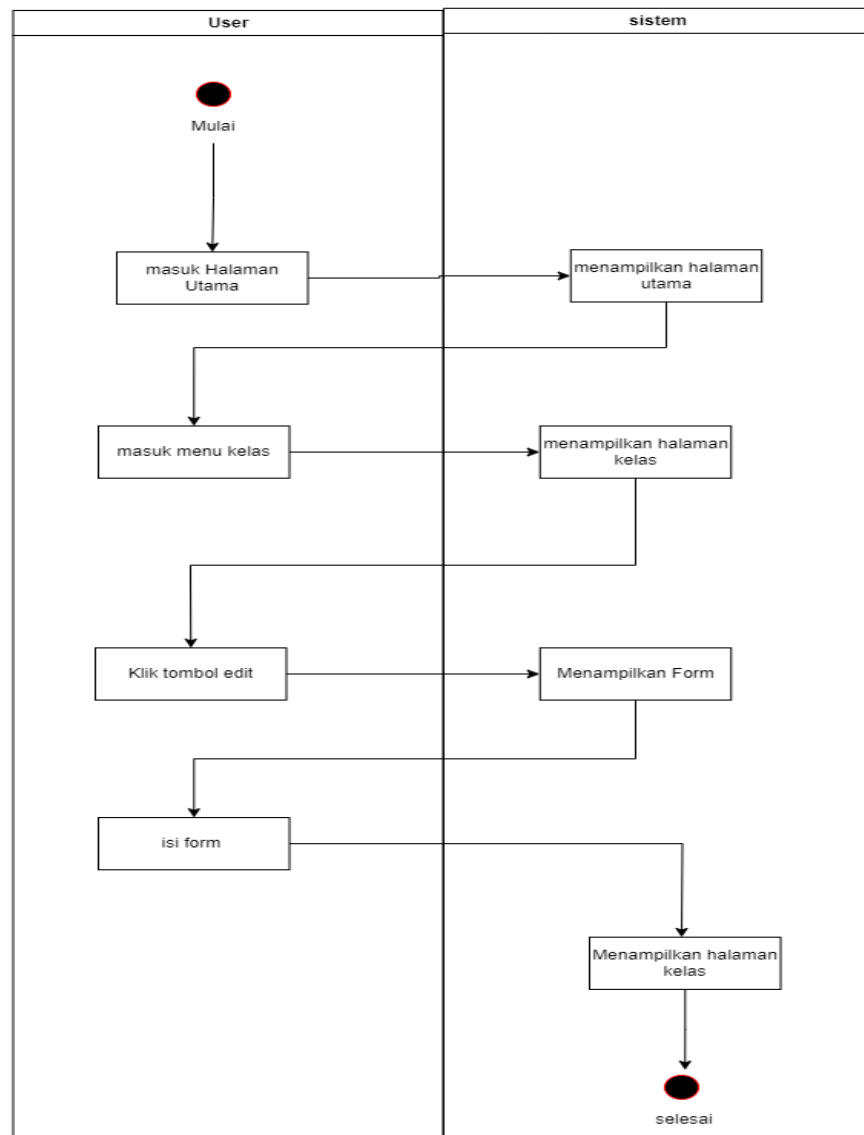


Gambar 4. 5 Activity Diagram Lupa Password

Pada gambar 4.4 menjelaskan bagaimana alur Ketika user atau admin ingin mengakses aplikasi manajemen perpustakaan. Pada *Activity* diagram lupa password diawali dengan user yang mengakses sistem dan sistem akan mengarahkan user ke halaman login lalu user mengklik tombol lupa password kemudian user akan menampilkan form yang harus di isi dengan

email user kemudian sistem akan memvalidasi apakah akun terdaftar atau tidak apabila tidak terdaftar akan muncul pop up gagal apabila berhasil link reset password akan di kirim ke email user

4) Activity Diagram Tambah Kelas

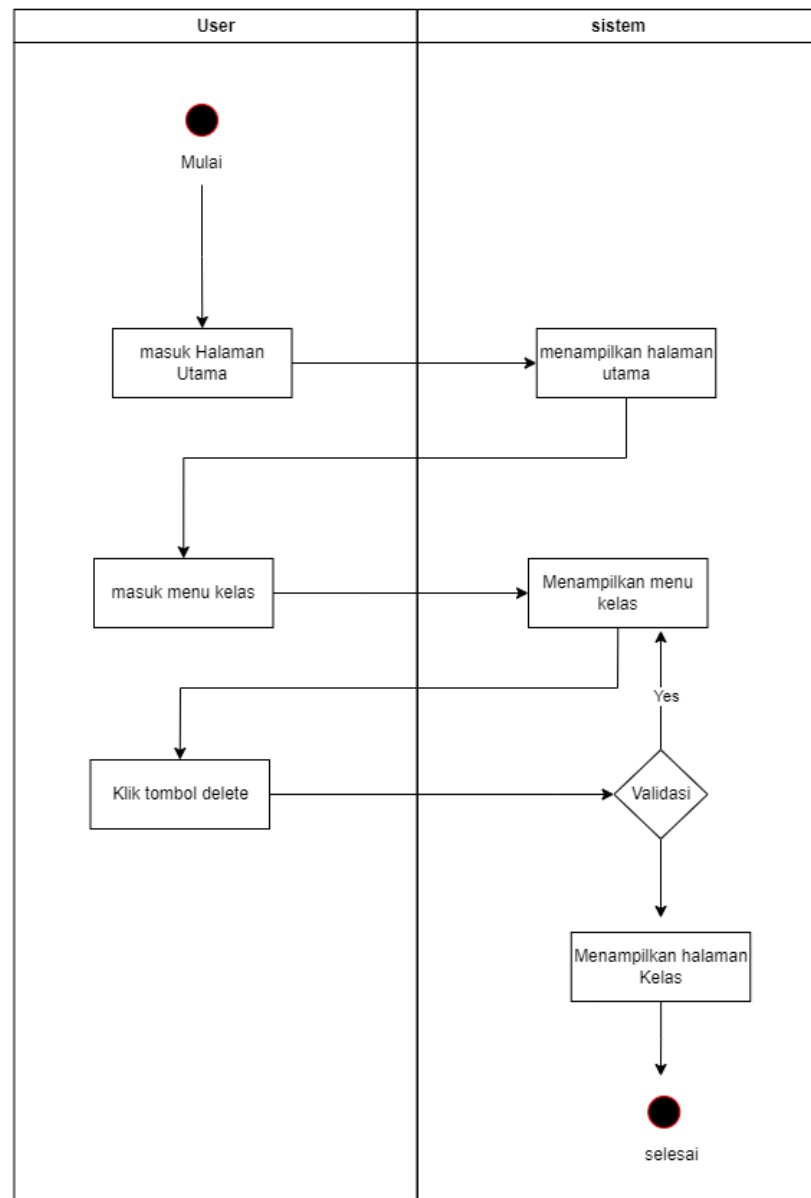


Gambar 4. 6 Activity Diagram Tambah Kelas

Pada gambar 4.4 ini menggambarkan bagaimana proses penambahan atau pembuatan kelas yang mana nantinya akan menjadi salah satu data diri untuk anggota perpustakaan Pada Activity diagram Tambah kelas dimulai dengan mengakses sistem

kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu kelas lalu user memilih tombol create kemudia mengisi form setelah selesai akan di arahkan ke halaman kelas.

5) Activity Diagram Hapus Kelas

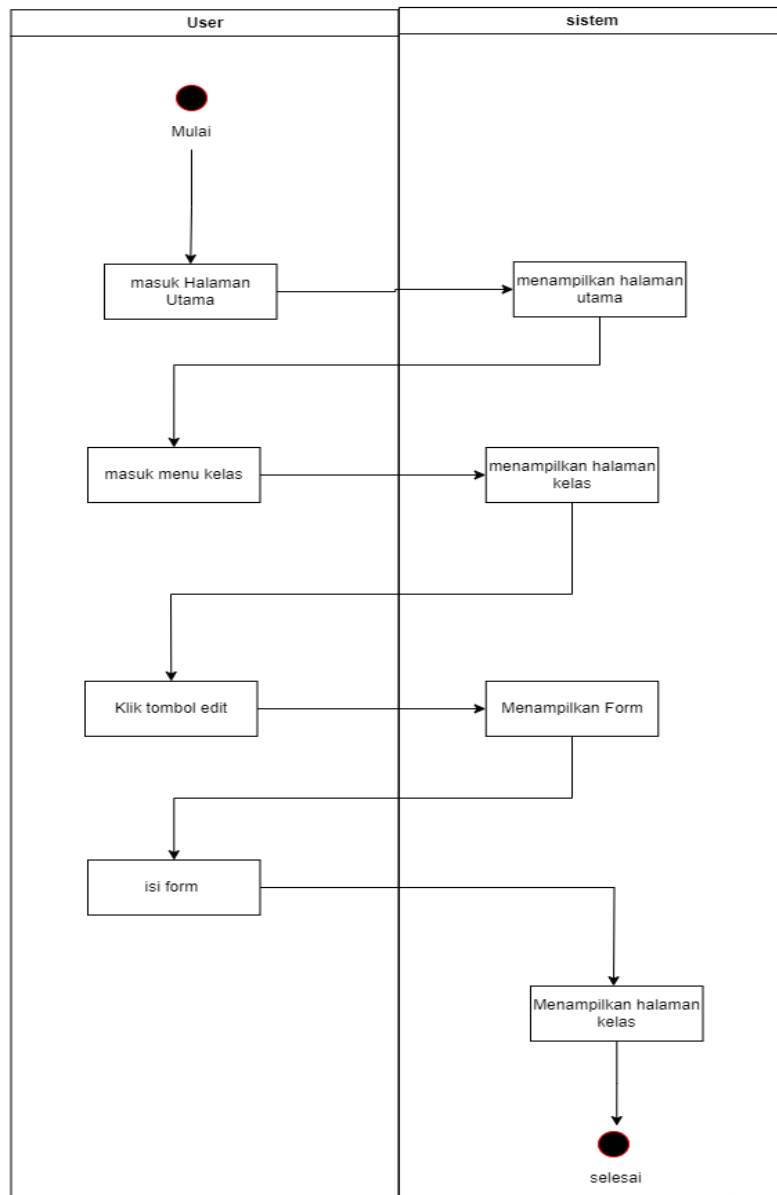


Gambar 4. 7 Activity Diagram Hapus Kelas

Pada gambar 4.5 menggambarkan alur penghapusan kelas dikarenakan setiap tahun atau setiap Angkatan memiliki jumlah kelas yang berbeda hal .Pada Activity diagram hapus kelas dimulai dengan

mengakses sistem kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu kelas lalu user memilih tombol delete pada kelas yang ingin dihapus

6) *Activity Diagram Edit Kelas*

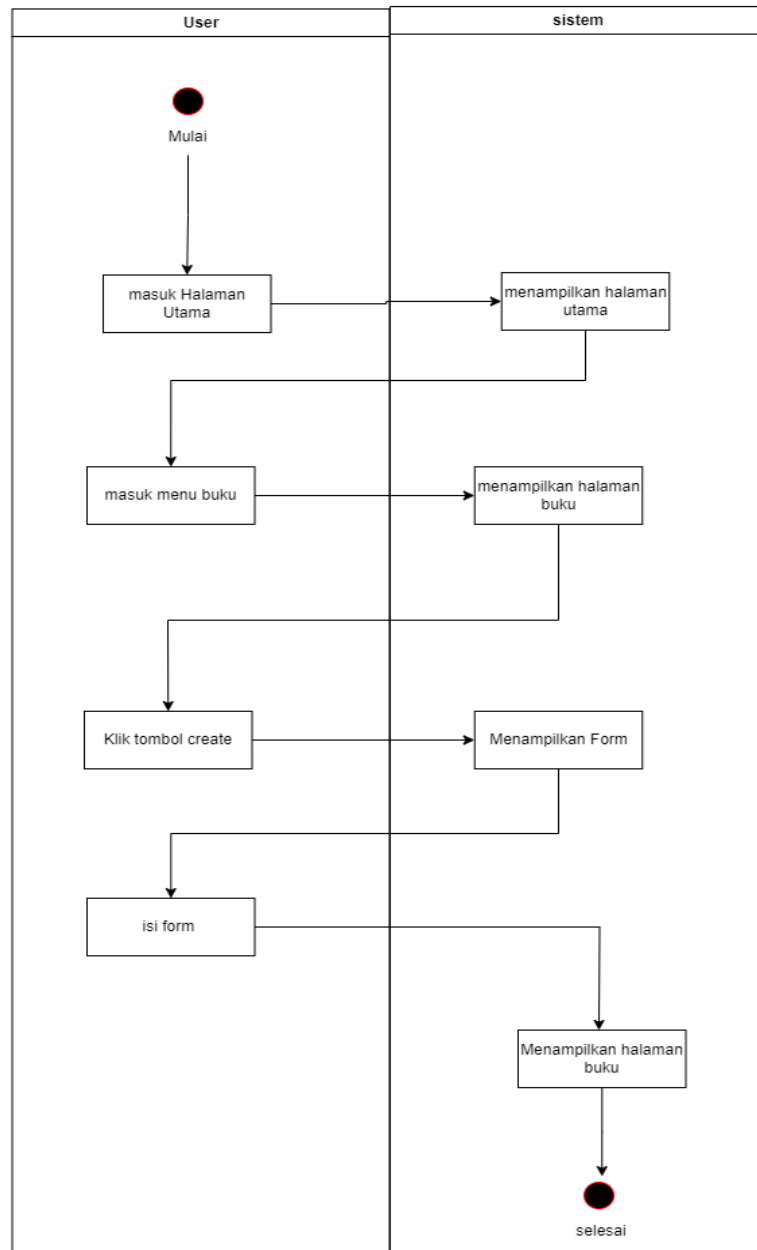


Gambar 4. 8 Activity Diagram Edit Kelas

Pada *Activity* diagram edit kelas dimulai dengan mengakses sistem kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu kelas

lalu user memilih tombol edit kemudian user akan diarahkan untuk mengisi form setelah selesai akan di arahkan ke halaman kelas.

7) Activity Diagram Tambah Buku

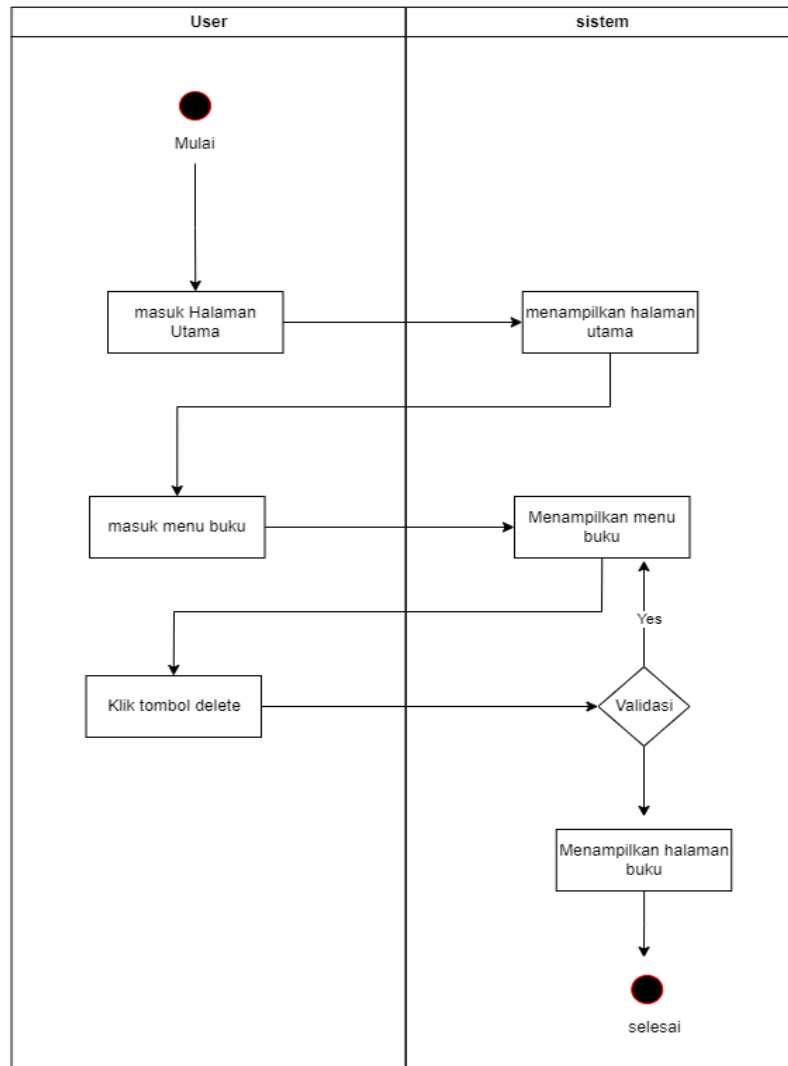


Gambar 4. 9 Activity Diagram Tambah Buku

Pada gambar 4.7 menggambarkan alur penambahan buku yang mana biasanya setiap tahun ajaran baru akan banyak buku baru yang masuk ke perpustakaan. Pada Activity diagram Tambah Buku dimulai dengan mengakses sistem kemudian akan di arahkan

kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu kelas lalu user memilih tombol create kemudian User akan diarahkan untuk mengisi form yang berisi data data mengenai buku setelah user selesai mengisi form akan di arahkan ke halaman kelas.

8) Activity Diagram Hapus Buku

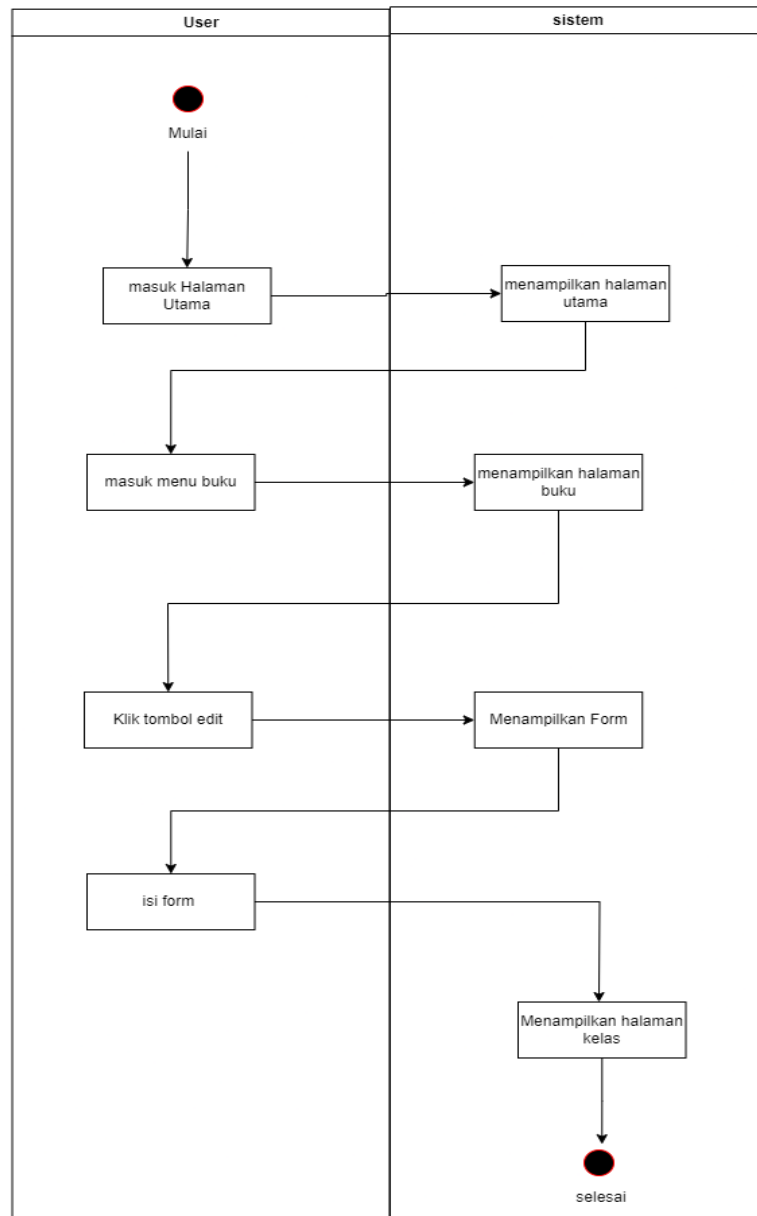


Gambar 4. 10 Activity Diagram Hapus Buku

Pada Activity diagram ini menggambarkan bagaimana proses penghapusan buku apabila ada buku yang rusak atau yang dilang di perpustakaan, Pada Activity diagram hapus buku dimulai dengan mengakses sistem kemudian akan di arahkan kehalaman utama

setelah sistem menampilkan halaman utama kemudian user memilih menu buku lalu user memilih tombol delete pada buku yang ingin dihapus

9) Activity Diagram Edit Buku

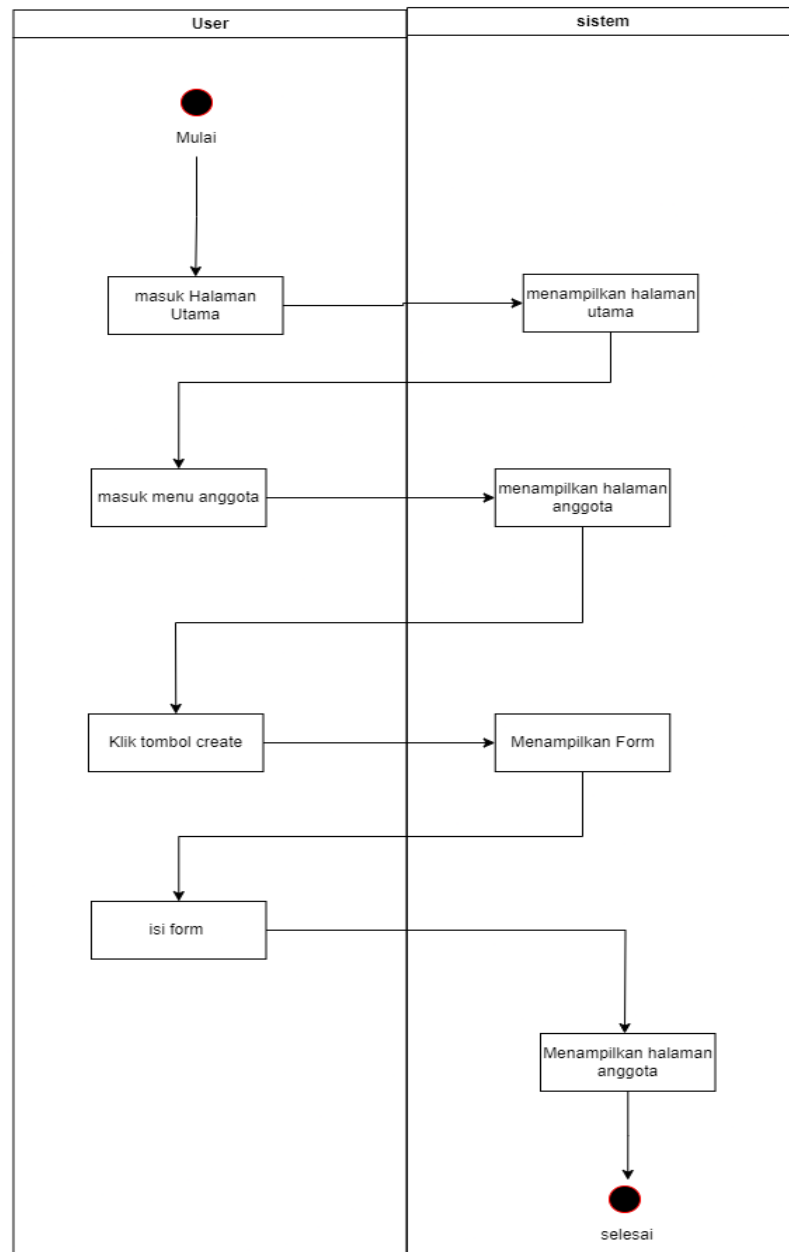


Gambar 4. 11 Activity Diagram Edit Buku

Pada gambar 4.9 menunjukkan alur dari edit buku .Pada Activity diagram edit kelas dimulai dengan mengakses sistem kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman

utama kemudian user memilih menu kelas lalu user memilih tombol edit kemudian user akan diarahkan untuk mengisi form setelah selesai akan di arahkan ke halaman kelas.

10) Activity Diagram Tambah Anggota

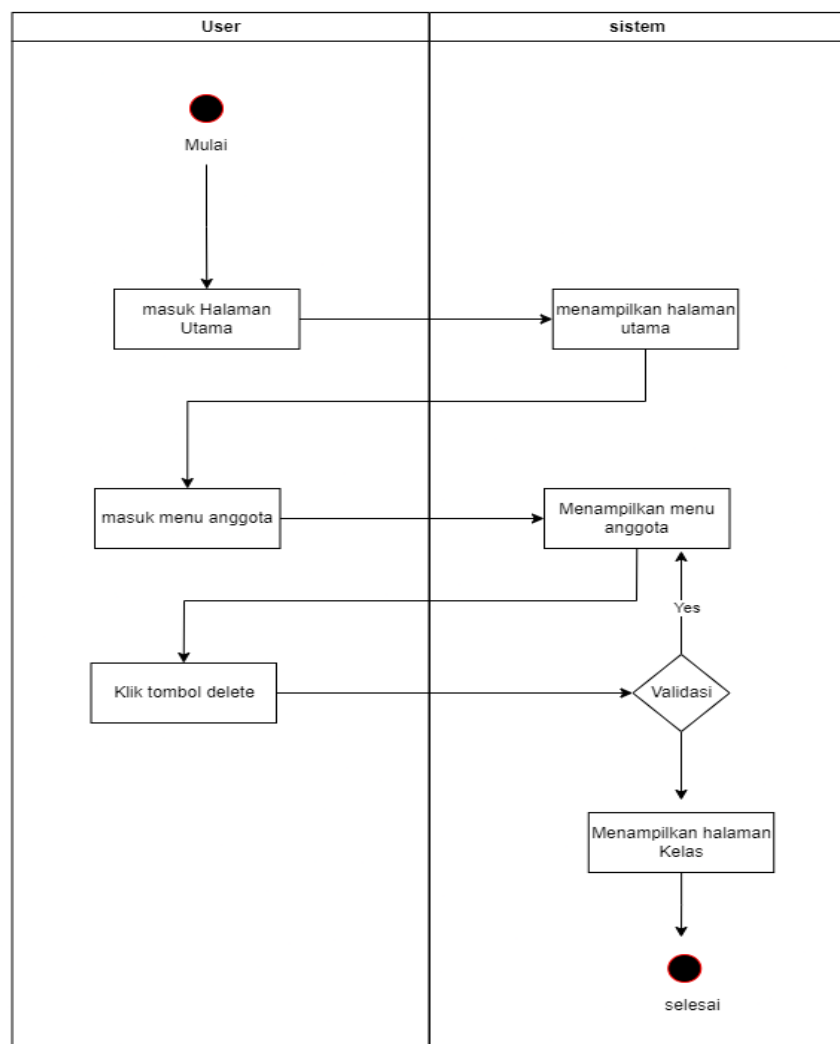


Gambar 4. 12 Activity Diagram Tambah Anggota

Pada gambar 4.10 menunjukkan bagaimana alur dari tambah anggota yang mana biasanya setiap ajaran baru aka nada siswa baru yang masuk atau ke SMP KP 2 Majalaya yang membuat ini menjadi salah

satu fitur penting . Pada *Activity* diagram Tambah anggota dimulai dengan mengakses sistem kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu anggota lalu user memilih tombol create kemudian User akan diarahkan untuk mengisi form yang berisi data data mengenai anggota setelah user selesai mengisi form akan di arahkan ke halaman kelas.

11) *Activity* Diagram Hapus Anggota

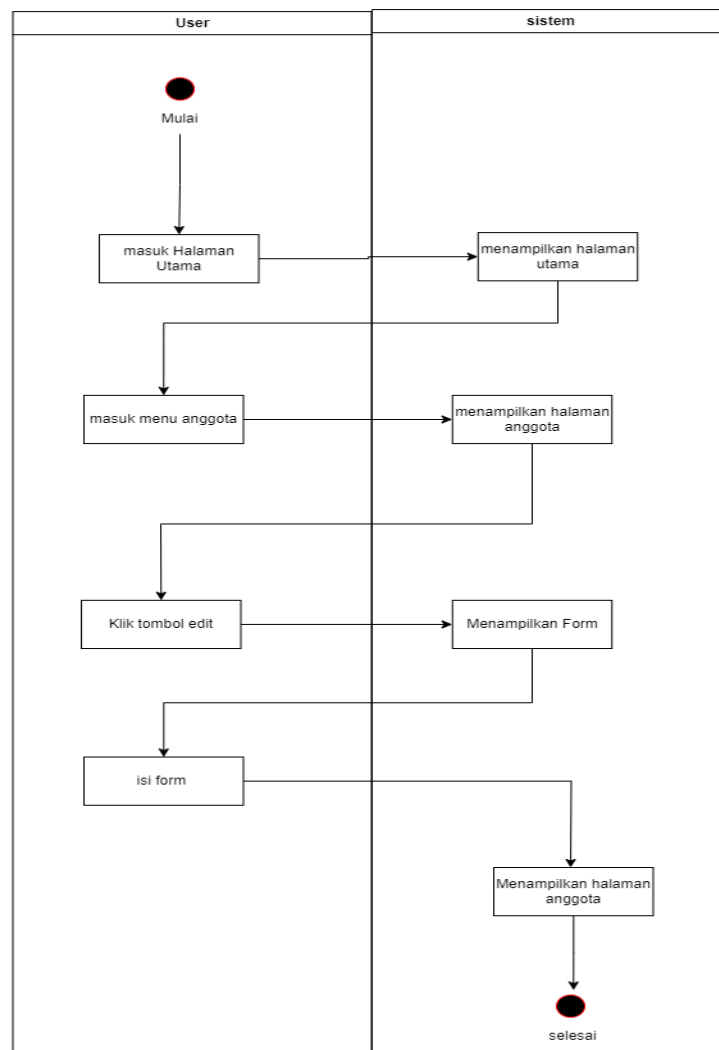


Gambar 4. 13 Activity Diagram Hapus Anggota

Pada gambar 4.11 menunjukan alur hapus anggota dikarenakan setiap tahun siswa akan lulus dari sekolah maka fitur ini di butuhkan agar data anggota dapat terkeklola dengan baik. Pada *Activity*

diagram hapus anggota dimulai dengan mengakses sistem kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu anggota lalu user memilih tombol delete pada anggota yang ingin dihapus

12) Activity Diagram Edit Anggota

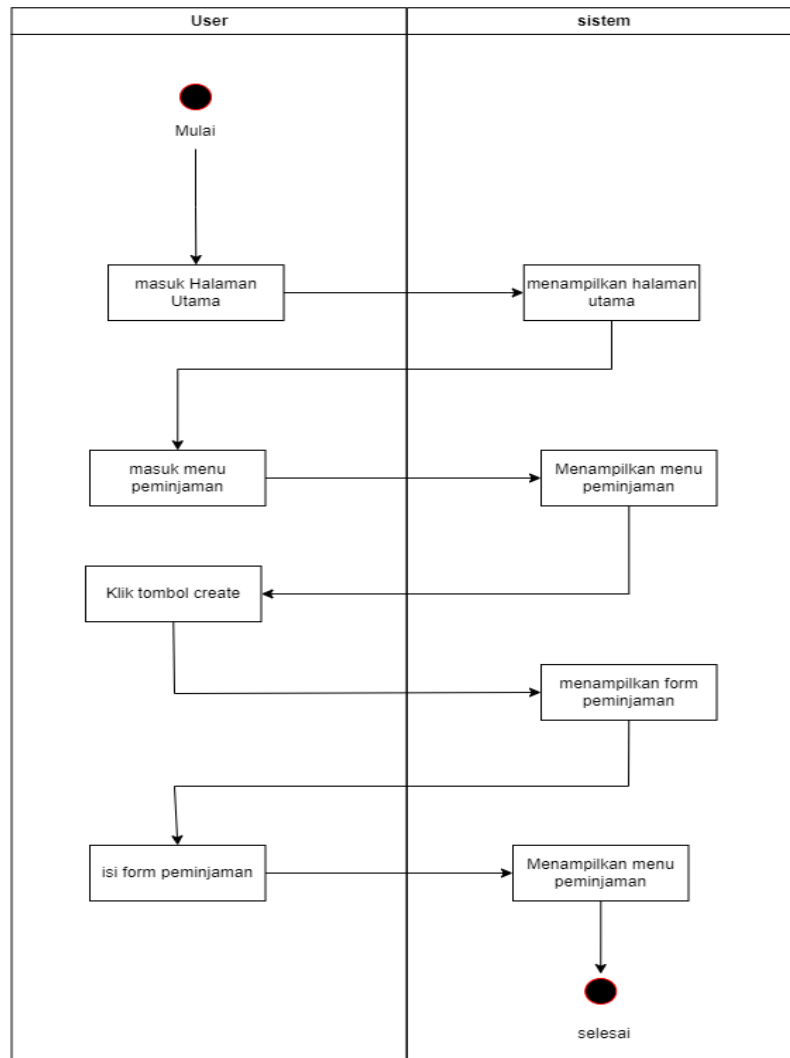


Gambar 4. 14 Activity Diagram Edit Anggota

Pada gambar 4.14 menggambarkan alur dari edit anggota dikarenakan bisa terjadi penginputan saat pembuatan data anggota maka fitur ini diperlukan untuk mengedit apabila ada data yang salah Pada Activity diagram edit kelas dimulai dengan mengakses sistem kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu anggota

lalu user memilih tombol edit kemudian user akan diarahkan untuk mengisi form setelah selesai akan di arahkan ke halaman anggota.

13) *Activity Diagram Peminjaman*

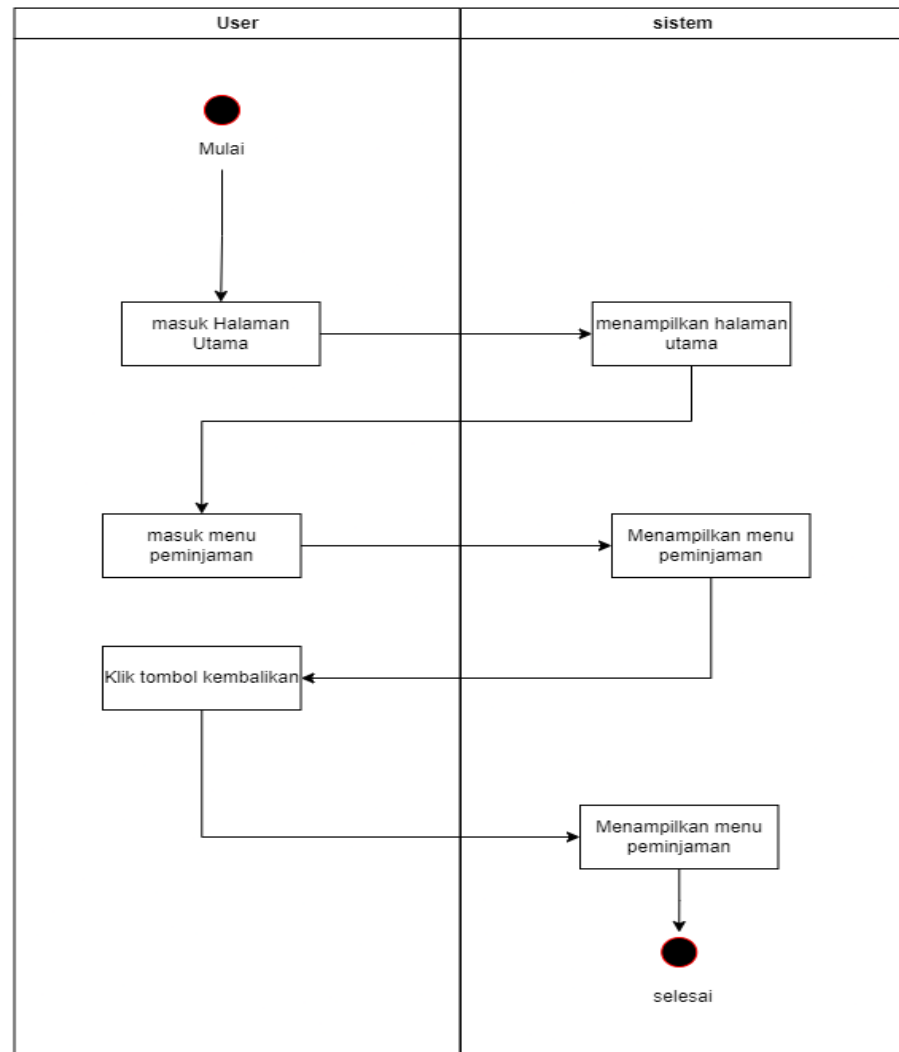


Gambar 4. 15 Activity Digram Peminjaman

Pada gambar 4.15 menunjukkan alur dari peminjaman .Pada *Activity diagram* kelas dimulai dengan mengakses sistem kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu peminjaman yang nantinya akan di arahkan ke halaman peminjaman kemudian user mengklik tombol create untuk melakukan peminjaman buku lalu user mengisi form peminjaman lalu sistem akan memverifikasi apakah stok nya

tersedia atau tidak kalau tersedia peminjaman berhasil dan akan di arahkan ke halaman peminjaman apabila gagal akan di arahkan Kembali untuk mengisi form

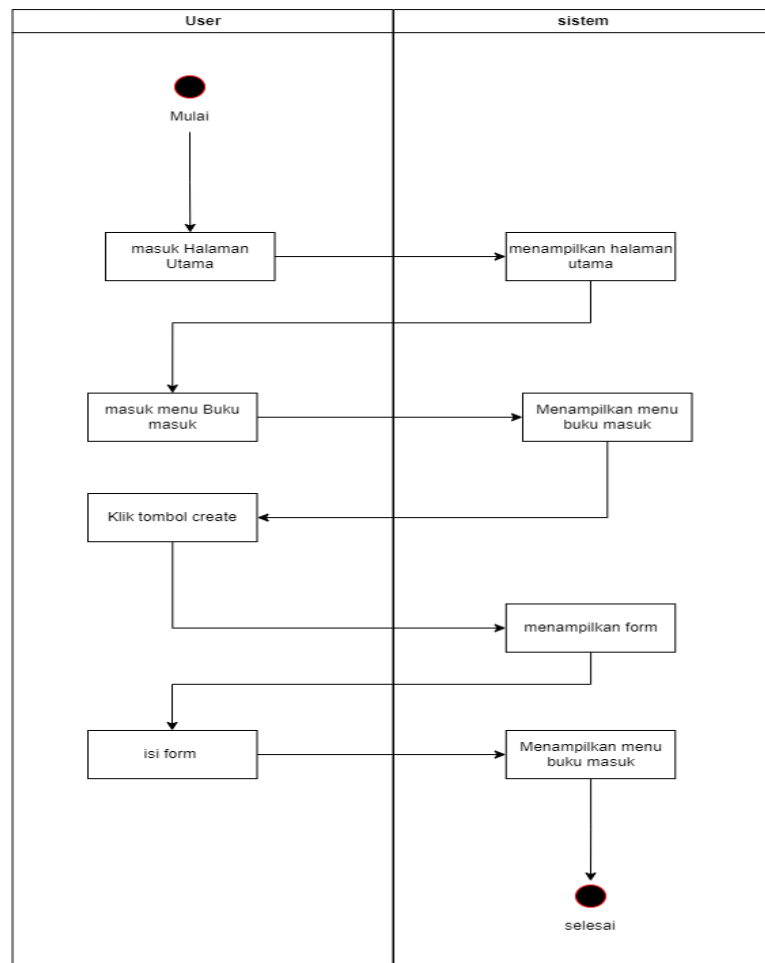
14) Activity Diagram Pengembalian



Gambar 4. 16 Pengembalian Buku

Pada 4.16 menunjukan alur pengembalian buku. Pada Activity diagram anggota dimulai dengan mengakses sistem kemudian akan di arahkan ke halaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu peminjaman yang nantinya akan di arahkan ke halaman peminjaman lalu klik tombol kembalikan setelah itu user akan di arahkan ke menu peminjaman

15) Activity Diagram Buku Masuk

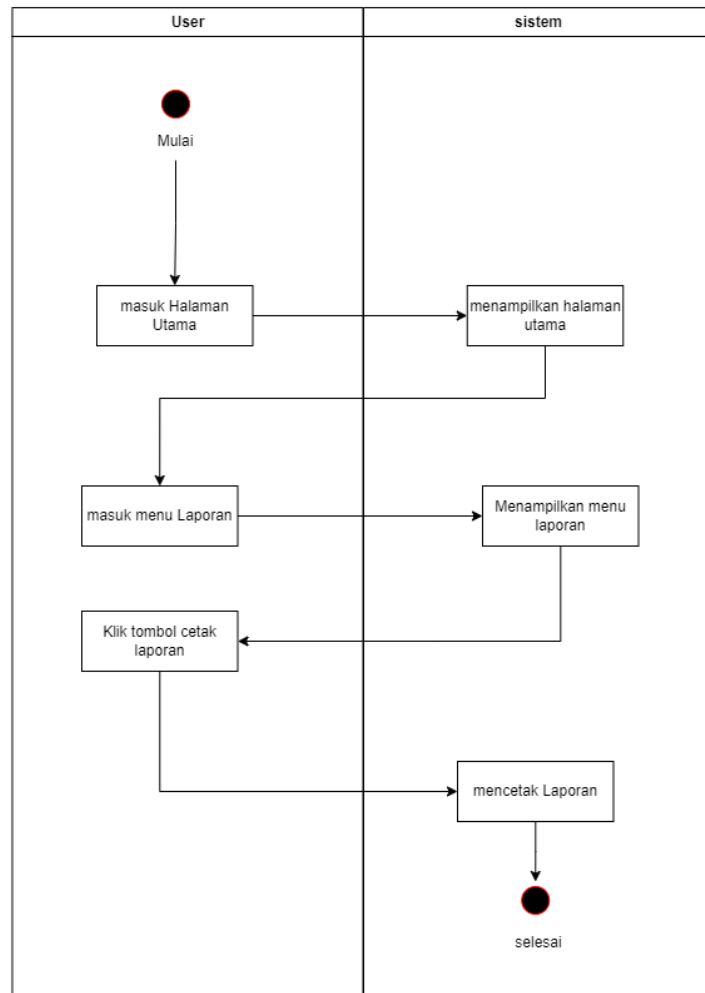


Gambar 4. 17 Activity diagram Buku masuk

Pada gambar 4.17 menggambarkan alur dari *Activity* buku masuk *Activity* ini dilakukan Ketika ada buku baru yang masuk tetapi sudah terdapat di perpustakaan atau bisa dikatakan menambah stok maka menggunakan fitur buku masuk .Pada *Activity* diagram buku masuk dimulai dengan mengakses sistem kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu Buku Masuk yang nantinya akan di arahkan ke halaman Buku masuk yang kemudian user mengklik tombol create lalu sistem akan menampilkan form setelah user mengisi form akan diarahkan Kembali ke menu buku masuk fitur ini

digunakan untuk menambahkan buku baru yang belum ada di perpustakaan.

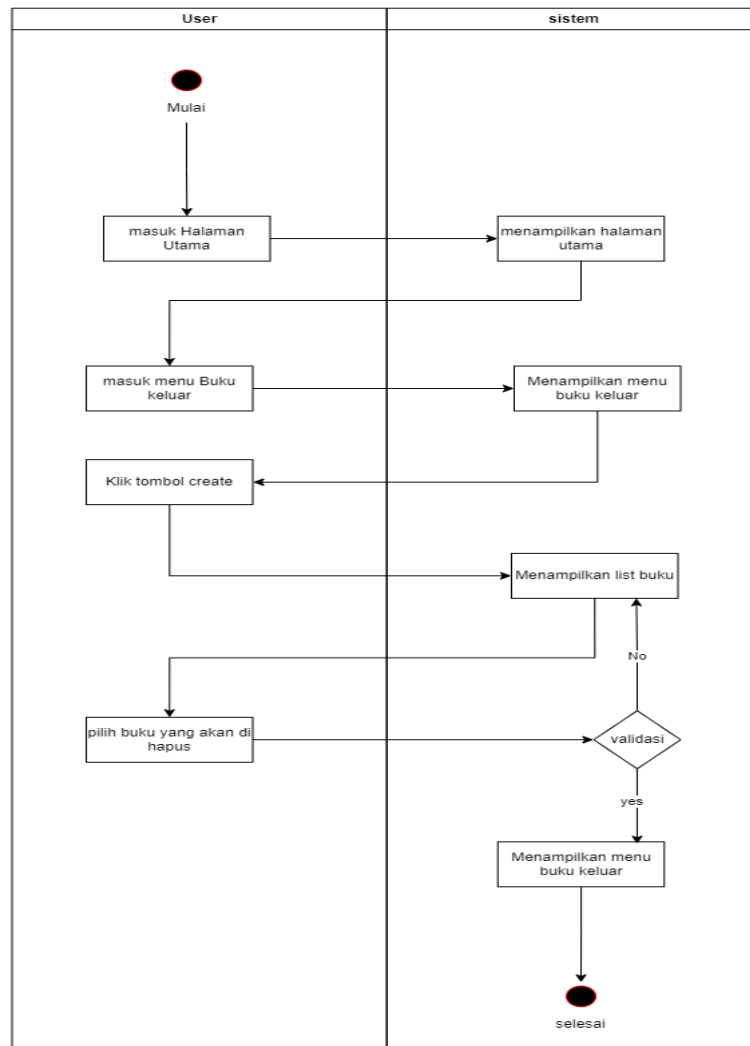
16) Activity Diagram Cetak Laporan



Gambar 4. 18 Activity Diagram Cetak laporan

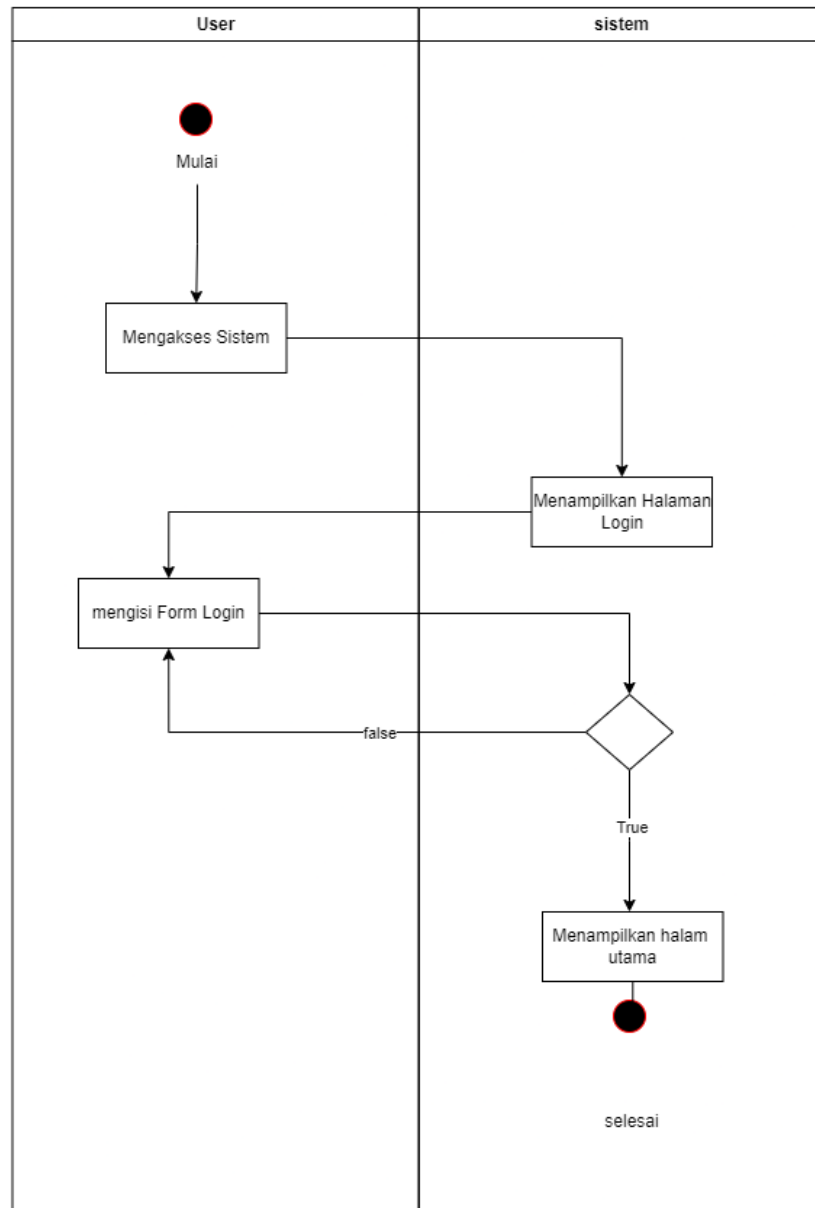
Pada Activity diagram cetak laporan diawali dengan user yang mengakses halaman utama kemudian sistem akan menampilkan halaman utama setelah itu user memilih menu laporan dan mengklik tombol cetak laporan kemudian sistem akan memberikan laporan dalam bentuk data pdf.

17) Activity Diagram Buku Keluar



Gambar 4. 19 Activity Diagram Buku Keluar

Pada Activity diagram Buku Keluar dimulai dengan mengakses sistem kemudian akan di arahkan kehalaman utama setelah sistem menampilkan halaman utama kemudian user memilih menu Buku keluar yang nantinya akan di arahkan ke halaman Buku keluar lalu klik tombol create setelah itu user memilih buku yang ingin dihapus apabila sudah user akan Kembali di arahkan kehalaman buku keluar

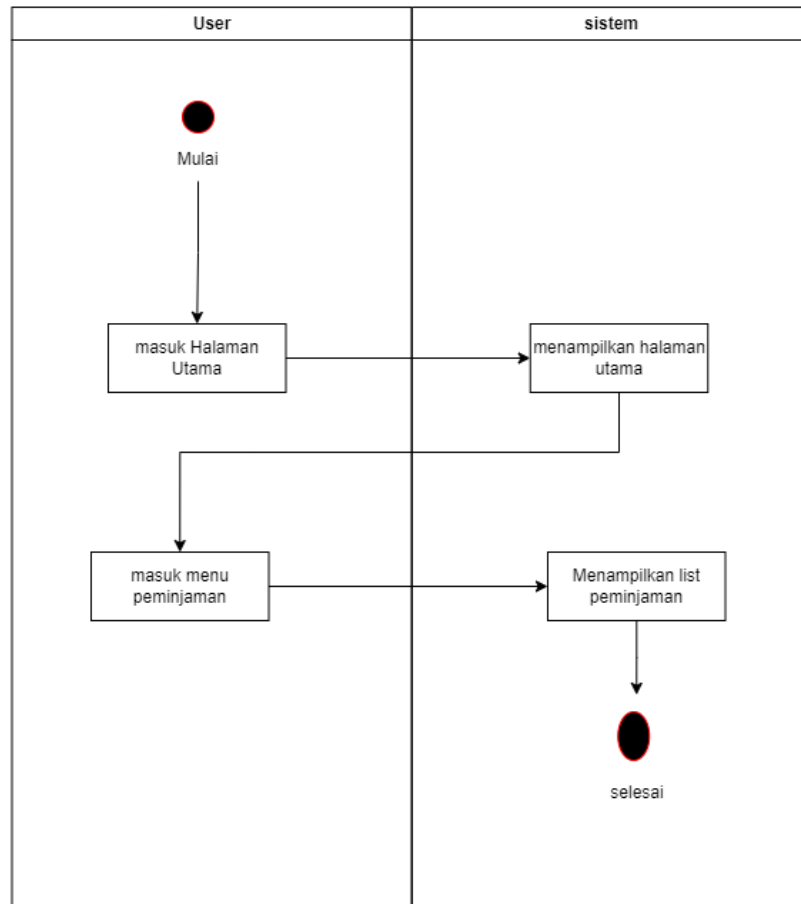
18) *Activity Diagram login User*

Gambar 4. 20 Activity Diagramm Halaman Login User

Pada gambar diatas menjelaskan bagaimana alur Ketika user atau admin ingin mengakses aplikasi manajemen pepustakaan. Pada *Activity diagram login* diawali dengan mengakses sistem kemudian sistem menampilkan form login yang harus di isi oleh user kemudian

form yang di isi user akan di verifikasi oleh sistem apabila login berhasil akan diarahkan ke halaman utama atau dashboard

19) Activity Diagram Peminjaman User

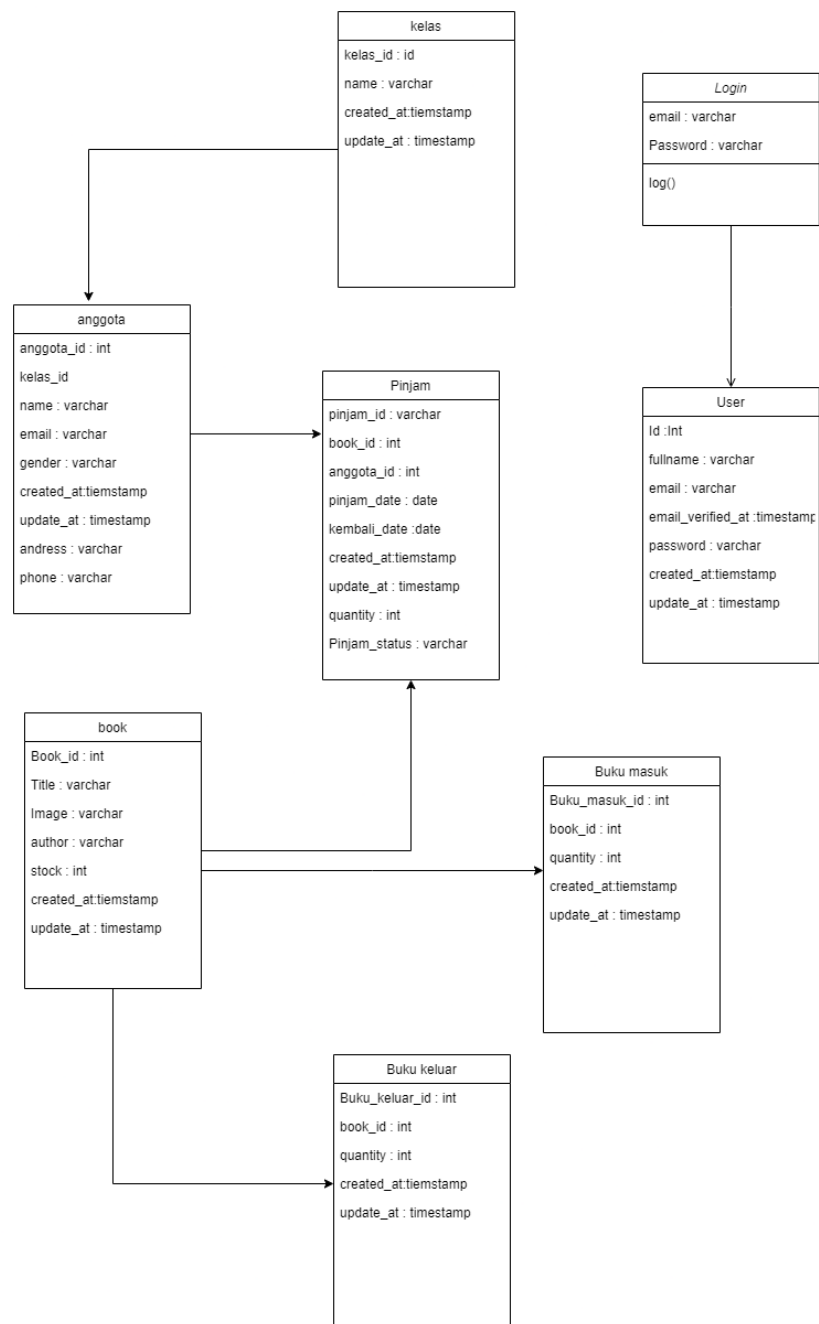


Gambar 4. 21 Activity Diagram Peminjaman User

Pada gambar diatas menjelaskan bagaimana proses melihat peminjaman pada user yang diawali dengan user yang mengakses sistem kemudian sistem yang menampilkan halaman utama kemudian user memilih menu peminjaman dan sistem menampilkan list peminjaman

3. Class Diagram

Class Diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun aplikasi.



Gambar 4. 22 Class Diagram

Dalam gambar 4.10 terdapat class-class umum yang digunakan untuk pengelolaan data pada database, diantaranya class login, class user, class buku, class anggota, class kelas, class pinjam, class buku keluar dan class buku masuk.

4.2.1 Struktur Tabel

Berikut adalah dekripsi tabel yang terdapat dalam basis data dan digunakan dalam aplikasi Manajemen Perpustakaan:

1. Perancangan tabel user

tabel 4. 4 Tabel User

Field	Type	Size	Index	Keterangan
ID	Integer	11	PK	ID user
Nama	varchar	30		Identitas user
Password	varchar	20		Password user
Email	varchar	30	FK	Identitas user
Remember_token	varchar	100		Identitas anggota
Created_at	Date_time	Timestamp		Waktu pembuatan
Update_at	Date_time	timestamp		Waktu pembaruan

2. Perancangan Tabel Anggota

tabel 4. 5 Tabel Anggota

Field	Type	Size	Index	Keterangan
ID_anggota	Integer	11	PK	ID anggota
Nama	varchar	30		Identitas anggota
ID_kelas	integer	11		Id kelas
Email	varchar	30		Identitas anggota
gender	varchar	20		Identitas anggota

alamat	varchar	30		Identitas anggota
No Hp	varchar	30		Identitas anggota
Created_at	Date_time	Timestamp		Waktu pembuatan
Update_at	Date_time	timestamo		Waktu pembaruan

3. Perancangan tabel buku

tabel 4. 6 Tabel Buku

Field	Type	Size	Index	Keterangan
Book_id	Integer	11	PK	ID book
title	varchar	30		Judul buku
image	varchar	30		Gambar buku
author	varchar	30		Penulis buku
stock	int	100		Jumlah buku
Created_at	Date_time	Timestamp		Waktu pembuatan
Update_at	Date_time	timestamo		Waktu pembaruan

4. Perancangan tabel buku masuk

tabel 4. 7 Tabel Buku Masuk

Field	Type	Size	Index	Keterangan
Buku_masuk_ID	Integer	11	PK	ID buku masuk
Book_id	Integer	11		Id buku

quantity	integer	100		Jumlah buku
Created_at	Date_time	Timestamp		Waktu pembuatan
Update_at	Date_time	timestamo		Waktu pembaruan

5. Perancangan tabel buku keluar

tabel 4. 8 Tabel Buku Keluar

Field	Type	Size	Index	Keterangan
Buku_keluar_ID	Integer	11	PK	ID buku keluar
Book_id	Integer	11		Id buku
quantity	integer	100		Jumlah buku
Created_at	Date_time	Timestamp		Waktu pembuatan
Update_at	Date_time	timestamo		Waktu pembaruan

6. Perancangan tabel kelas

tabel 4. 9 Tabel Kelas

Field	Type	Size	Index	Keterangan
kelas_ID	Integer	11	PK	ID buku masuk
Name	varchar	30		Nama kelas
Created_at	Date_time	Timestamp		Waktu pembuatan
Update_at	Date_time	timestamo		Waktu pembaruan

7. Perancangan tabel peminjaman

tabel 4. 10 Tabel Peminjaman

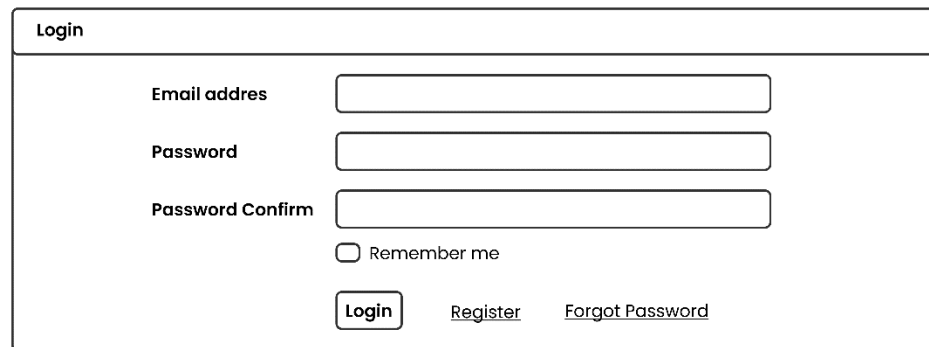
Field	Type	Size	Index	Keterangan
Pinjam_id	varchar	30	PK	ID peminjaman
Book_id	Id	11		Id buku
Anggota_id	integer	11		Id anggota
Pinjam_date	Date_time	30		Tanggal pinjam
Kembali_date	Date_time	30		Tanggal kembali
quantity	integer	100		Jumlah buku yang di pinjam
Pinjam_status	varchar	20		Status peminjaman
Created_at	Date_time	Timestamp		Waktu pembuatan
Update_at	Date_time	timestamo		Waktu pembaruan

4.2.2 Desain Antarmuka

Desain antarmuka, atau User Interface (UI) Design, adalah proses merancang tampilan dan interaksi pengguna dengan perangkat lunak, perangkat keras, atau situs web. Tujuan utama desain antarmuka adalah untuk menciptakan pengalaman pengguna yang mudah digunakan, intuitif dan menyenangkan,.berikut adalah desain antarmuka dari aplikasi manajemen perpustakaan

A. Admin

1. Desain Antarmuka Login

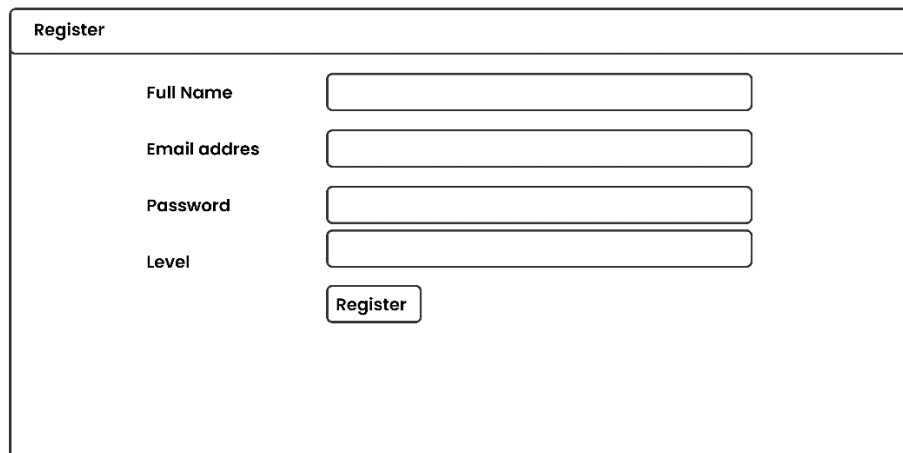


The image shows a login form with a title bar labeled "Login". Below the title bar, there are three input fields: "Email address", "Password", and "Password Confirm". Below the "Password Confirm" field, there is a checkbox labeled "Remember me". At the bottom of the form, there is a "Login" button, a "Register" link, and a "Forgot Password" link.

Gambar 4. 23 Desain antarmuka Login

Gambar di atas merupakan desain tampilan dari halaman *Login* yang memuat form *input* data *username* dan *password*. selain itu ada juga register yang digunakan apabila user belum memiliki akun

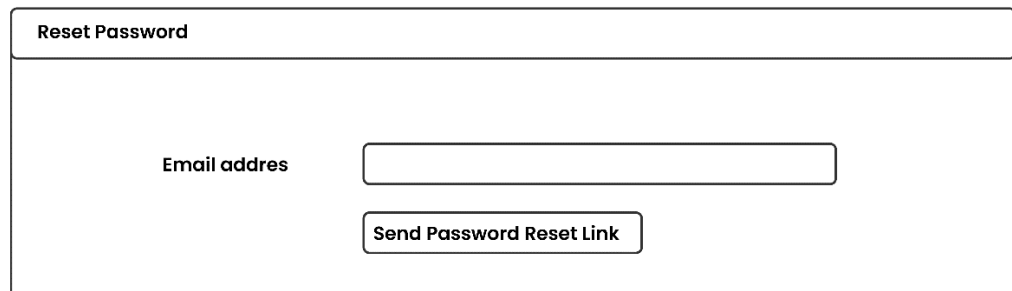
2. Desain Antarmuka Register



The image shows a register form with a title bar labeled "Register". Below the title bar, there are four input fields: "Full Name", "Email address", "Password", and "Level". Below the "Level" field, there is a "Register" button.

Gambar diatas merupakan desain antarmuka dari tampilan register Dimana memuat form register yang berisi nama email password dan level biasanya digunakan ketika user belum memiliki akun

3. Desain Antarmuka Reset password



The image shows a 'Reset Password' form. It has a title bar 'Reset Password'. Below it, there is a label 'Email address' followed by a text input field. Below the input field is a button labeled 'Send Password Reset Link'.

Gambar 4. 24 Desain Antarmuka Reset Password

Gambar diatas merupakan desain atarmuka dari tampilan reset password Dimana memuat form yang mana harus memasukan email apabila sudah memasukan email user halus mengklik botton send password reset

4. Desain Antarmuka Dashboard



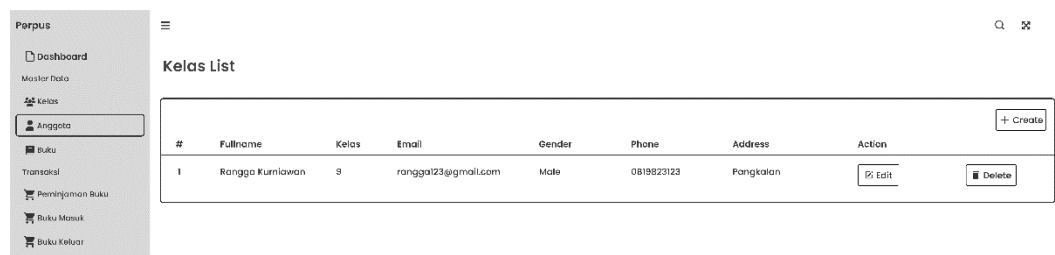
The image shows a dashboard for 'Perpus'. It has a sidebar with navigation links: Dashboard, Master Data, Kelas, Anggota, Buku, Transaksi, Peminjaman Buku, Buku Masuk, and Buku Keluar. The main content area is titled 'Dashboard' and contains several summary cards with counts and 'View Details' links:

- Book (2) - View Details
- Anggota (2) - View Details
- Peminjaman (2) - View Details
- Total Peminjaman Laki-laki (2) - View Details
- Total Peminjaman Perempuan (2) - View Details
- Buku Masuk (2) - View Details
- Buku Keluar (2) - View Details

Gambar 4. 25 Desain Antarmuka Dashboard

Desain dari halaman dashboar berisikan informasi informasi umum yang nantinya berguna dalam pembuatan laporan seperti jumlah buku,jumlah anggota,jumlah peminjam laki laki dan Perempuan serta jumlah buku masuk dan keluar

5. Desain Antarmuka Anggota



The image shows an 'Anggota List' table. It has a sidebar with navigation links: Dashboard, Master Data, Kelas, Anggota, Buku, Transaksi, Peminjaman Buku, Buku Masuk, and Buku Keluar. The main content area is titled 'Kelas List' and contains a table with the following columns: #, Fullname, Kelas, Email, Gender, Phone, Address, and Action. The table has one row of data:

#	Fullname	Kelas	Email	Gender	Phone	Address	Action
1	Rangga Kurniawan	9	ranggal23@gmail.com	Male	0819823123	Pangkajene	[Edit] [Delete]

There is a '+ Create' button in the top right corner of the table area.

Gambar 4. 26 Desain Antarmuka Anggota

Desain dari halaman anggota memuat list dari anggota beserta data data mengenai anggota yang dapat meminjam buku, dan juga ada *botton* yang digunakan untuk membuat anggota,mengedit ataupun menghapus anggota

6. Desain Antarmuka Buku



Gambar 4. 27 Desain Antarmuka Buku

Desain antarmuka dari halaman buku berisikan list buku yang ada mulai dari judul,foto,*author* stok serta ada button edit delet serta create

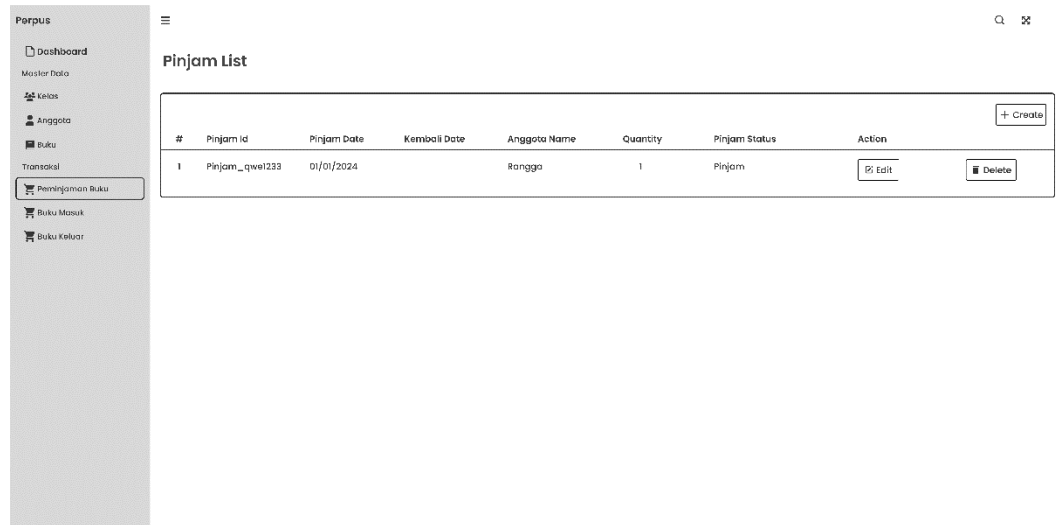
7. Desain Antarmuka Kelas



Gambar 4. 28 Desain Antarmuka Kelas

Desain dari halaman kelas memuat list dari kelas yang ada dan juga ada button yang digunakan untuk membuat kelas,mengedit ataupun menghapus kelas

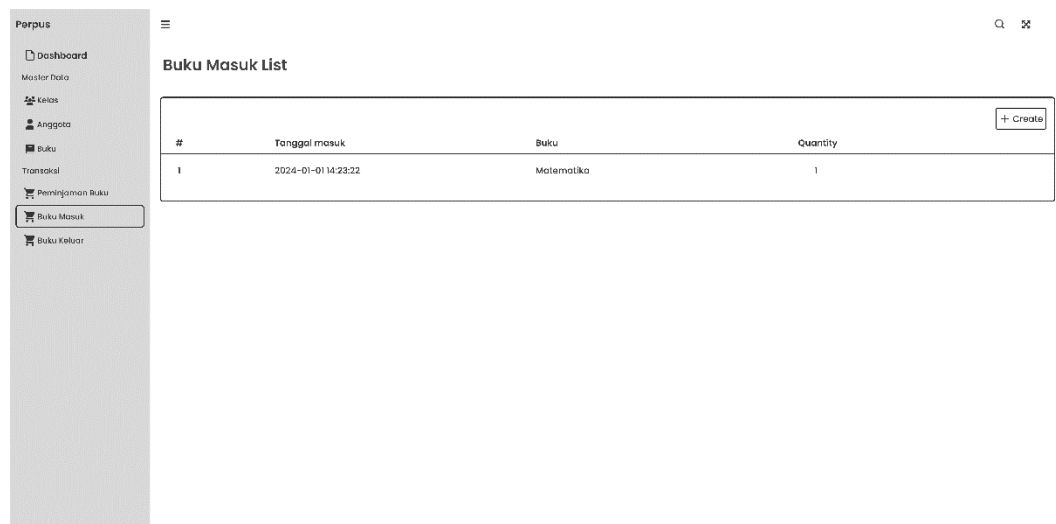
8. Desain Antarmuka Peminjaman



Gambar 4. 29 Desain Antarmuka Peminjaman

Desain antarmuka halaman peminjaman memuat mengenai daftar orang-orang yang meminjam buku serta ada button yang digunakan untuk melakukan peminjaman/pengembalian.

9. Desain Antarmuka Buku Masuk



Gambar 4. 30 Desain Antarmuka Buku Masuk

Desain antarmuka dari halaman buku masuk memuat list buku-buku yang telah ditambahkan kemudian ada *button* create yang digunakan untuk menambahkan stok buku.

10. Desain Antarmuka Buku Keluar

#	Tanggal Keluar	Buku	Quantity
1	2024-01-01 14:23:22	Matematika	1

Gambar 4. 31 Desain Antar Muka Buku Keluar

Desain antarmuka dari halaman buku masuk memuat list buku buku yang telah dihapus kemudian ada *button* create yang digunakan untuk mengurangi stok buku

B. User

1. Desain Antarmuka Login User

Gambar 4. 32 Desain Antarmuka User

Gambar di atas merupakan desain tampilan dari halaman *Login* yang memuat form *input* data *username* dan *password*. selain itu ada juga register yang digunakan apabila user belum memiliki akun

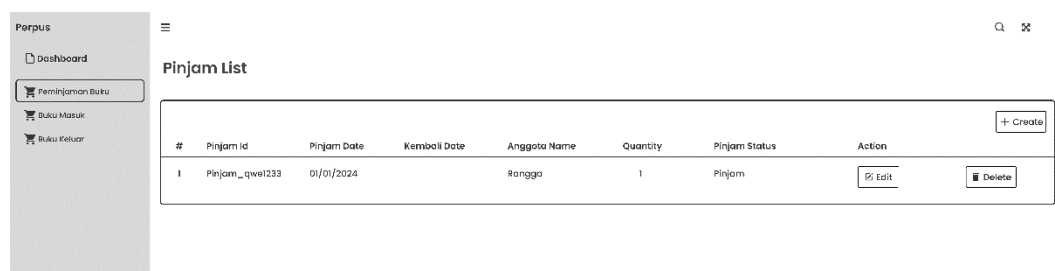
2. Desain antarmuka Dashboard



Gambar 4. 33 Desain Antarmuka Dashboard User

Desain dari halaman dashboar User sama dengan admin berisikan informasi informasi umum yang nantinya berguna dalam pembuatan laporan seperti jumlah buku, jumlah anggota, jumlah peminjam laki laki dan Perempuan serta jumlah buku masuk dan keluar yang membedakan hanya dari jumlah menu yang dapat di akses

3. Desain Antarmuk Peminjaman



Gambar 4. 34 Desain Antarmuka Peminjaman

Desain antarmuka halaman peminjaman user sama dengan admin memuat mengenai daftar orang orang yang meminjam buku serta ada botton yang digunakan untuk melakuka peminjaman pengembalian yang membedakan user hanya bisa melihat saja

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Setelah melakukan analisis dan perancangan maka dilanjutkan ketahap pengimplementasian untuk menjalankan hasil analisis dan perancangan yang sebelumnya telah dibuat.

5.1.1 Listing Program

1. Login

```
public function index(){
    return view('auth.login');
}

public function register(){
    return view('auth.register');
}

public function proses_login(Request $request){

    // isi credential hanya berupa username dan password
    $credentials = $request->only('email','password');
    //validasi menggunakan Illuminate\Support\Facades\Validator
    //isi field validasi dan rules nya yaitu wajib di isi
    $validate = Validator::make($credentials,[
        'email'=>'required|email',
        'password'=>'required'
    ]);

    // jika terdapat field yang kosong
    if($validate->fails()){
        //kembali ke halaman login & tampilkan error pada
        //setiap inputnya
        return back()->withErrors($validate)->withInput();
    }

    // verifikasi data user pada kolom email dan password
    //sesuai atau belum
    if(Auth::attempt($credentials)){
        //jika sesuai maka jalankan fungsi dashboard
        return redirect()->intended('dashboard')->with('success','Successfully Login');
    }
    // kembali ke halaman login dan tampilkan pesan error pada
    //login_error
    return redirect('login')->withInput()->withErrors(['login_error'=>'Username or password are wrong!']);
}
```



```

public function dashboard(){
    // cek berhasil login
    if(Auth::check()){
        // menampilkan total data
        $totalAnggota = Anggota::all()->count();
        $totalBook = Book::all()->count();
        $totalPinjam = Pinjam::all()->count();
        $totalPinjamLaki = Pinjam::join('anggota',
'pinjam.anggota_id', '=', 'anggota.anggota_id')
->where('gender', 'male')
->count();
        $totalPinjamCewe = Pinjam::join('anggota',
'pinjam.anggota_id', '=', 'anggota.anggota_id')
->where('gender', 'female')
->count();
        $totalQuantityMasuk = BukuMasuk::sum('quantity');
        $totalQuantityKeluar = BukuKeluar::sum('quantity');
        return
view('home',compact('totalAnggota','totalBook','totalPinjam','totalPinjamLaki','totalPinjamCewe','totalQuantityMasuk','totalQuantityKeluar'));
    }

    return redirect('login')->with('You dont have access');
}

public function proses_register(Request $request){
    //validasi menggunakan Illuminate\Support\Facades\Validator
    //isi field validasi dan rules nya yaitu wajib di isi
    $validate = Validator::make($request->all(),[
        'fullname'=>'required',
        'email'=>'required|email',
        'password'=>'required',
    ]);
    // jika terdapat field yang kosong
    if($validate->fails()){
        //kembali ke halaman login & tampilkan error pada
        //setiap inputnya
        return back()->withErrors($validate)->withInput();
    }

    // tambahkan field level dan kita isi dengan admin
    $request['level'] = 'admin';
    // panggil model User dan jalankan fungsi ORM create untuk
    //melakukan insert semua data
    User::create($request->all());

    return redirect('dashboard')->with('success','You have
successfully register');
}

public function logout(){
    // clear session dan memberitahu auth dengan status logout
    Session::flush();
    Auth::logout();
}

```

```

        return Redirect('login');
    }
}

```

2. Pinjam

```

public function index()
{
    //query join antara tabel anggota,buku dan pinjam
    // setelah joint baru ambil kolom nya pakai fungsi select
    $pinjam = Pinjam::join('anggota', 'pinjam.anggota_id',
'=', 'anggota.anggota_id')
->join('book', 'pinjam.book_id', '=', 'book.book_id')
->select([
    'pinjam.pinjam_id',
    'pinjam.pinjam_date',
    'pinjam.kembali_date',
    'anggota.fullname',
    'anggota.email',
    'book.title',
    'pinjam.quantity',
    'pinjam.pinjam_status'
])
->orderBy('pinjam.pinjam_id', 'desc')
->paginate(5);
return view('pinjam_list',compact('pinjam'))
->with('i',(request()->input('page',1)-1)*5);
}

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    //
    $anggota = Anggota::all();
    $book = Book::all();
    return view('pinjam_add',compact('anggota','book'));
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    $book = Book::where('book_id', $request->book_id)->first();
    if ($book->stock < $request->quantity) {
        return redirect()->route('pinjam.index')->with('error', 'Stock not enough');
    }

    try {
        // Menjalankan fungsi insert pada table buku keluar
        Pinjam::create($request->all());
        // Mengurangi stok buku yang keluar dari stok total
    }
}

```

```

        $book->stock -= $request->quantity;
        $book->save();

        // Redirect ke halaman list buku keluar
        return redirect()->route('pinjam.index')-
>with('success','Successfully created new buku keluar');
    } catch (\Throwable $th) {
        // Munculkan pesan error jika ada error
        return redirect()->route('pinjam.index')-
>with('error', $th->getMessage());
    }
}

/**
 * Display the specified resource.
 */
public function show(string $id)
{
    //

}

public function kembalikan($id)
{
    //
    $anggota = Anggota::all();
    $book = Book::all();

    $pinjam =
Pinjam::join('anggota','pinjam.anggota_id','=','anggota.anggota_i
d')
    ->join('book','pinjam.book_id','=','book.book_id')
    ->where('pinjam_id',$id)
    ->select([
        'pinjam.pinjam_id as
pinjam_id','pinjam.pinjam_date as pinjam_date','anggota.email as
email','anggota.anggota_id
anggota_id','book.book_id','anggota.fullname
fullname','book.title as title','pinjam.quantity as
quantity','pinjam.pinjam_status as pinjam_status'])
    ->firstOrFail();

    if($pinjam){
        return
view('pinjam_kembalikan',compact('pinjam','anggota','book'));
    }else{
        return redirect()->route('pinjam.index')-
>with('error','pinjam not found');
    }

}

/**
 * Show the form for editing the specified resource.
 */
public function edit($id)
{

```

```

//
$anggota = Anggota::all();
$book = Book::all();

$pinjam =

Pinjam::join('anggota','pinjam.anggota_id','=','anggota.anggota_id')
->join('book','pinjam.book_id','=','book.book_id')
->where('pinjam_id',$id)
->select([
    'pinjam.pinjam_id as pinjam_id', 'pinjam.pinjam_date as pinjam_date', 'anggota.email as email', 'anggota.anggota_id as anggota_id', 'book.book_id as book_id', 'anggota.fullname as fullname', 'book.title as title', 'pinjam.quantity as quantity', 'pinjam.pinjam_status as pinjam_status'])
->firstOrFail();
if($pinjam){

    return
view('pinjam_edit',compact('pinjam','anggota','book'));
}else{
    return redirect()->route('pinjam.index')->with('error','pinjam not found');
}

}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, $id)
{
    if($request->status_update == "Kembali"){
        // Mendapatkan buku yang masuk berdasarkan ID
        $buku = Book::findOrFail($request->book_id);

        // Menambahkan stok buku yang masuk ke stok total
        $buku->stock += $request->quantity;
        $buku->save();
    }else{
        // Mendapatkan buku yang masuk berdasarkan ID
        $buku = Book::findOrFail($request->book_id);
        $buku->stock += $request->quantity_asal;
        // Menambahkan stok buku yang masuk ke stok total
        $buku->stock -= $request->quantity;
        $buku->save();
    }
    Pinjam::where('pinjam_id',$id)->update([
        'pinjam_date'=> $request->pinjam_date,
        'kembali_date'=> $request->kembali_date,
        'anggota_id'=> $request->anggota_id,
        'book_id'=> $request->book_id,
        'quantity'=> $request->quantity,
        'pinjam_status'=> $request->pinjam_status
    ]);
}

```

```

        return redirect()->route('pinjam.index')-
>with('success','Successfully update data');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy($id)
    {
        //
        Pinjam::where('pinjam_id',$id)->delete();

        return redirect()->route('pinjam.index')-
>with('success','Successfully delete data');
    }

    public function getpinjamId(){

        $randomString = uniqid("pinjam_");

        return response()->json([
            'key'=>$randomString
        ]);
    }
}

```

Untuk Listing program selengkap nya penulis cantumkan dilampiran 4

5.1.2 Implementasi sistem

Implementasi sistem merupakan proses penerapan sistem yang dilakukan setelah sistem mendapatkan persetujuan pada program yang telah dibuat. Waktu dan tempat penerapan sistem sebagai berikut:

Tempat : Perpustakaan SMP KP 2 Majalaya
 Alamat : Jl-Wangisagara,Desa Wangisagara,kecamatan
 Majalaya,kabupaten Bandung,Jawa Barat
 Waktu : Juli

5.1.3 Spesifikasi Sistem

Spesifikasi sistem mengacu kepada kebutuhan perangkat yang digunakan dalam pengimplementasian aplikasi manajemen perpustakaan.

1. Perangkat Keras

Tabel 5. 1 Spesifikasi Perangkat Keras

Processor	Intel (R) Celeron (R) CPU 3867U @ 1.80GHz (2CPUs), ~1.8GHz
RAM	4096MB
SSD	256G

2. Perangkat Lunak

Tabel 5. 2 Spesifikasi Perangkat Lunak

Sistem Operasi	Windows 11
Database	Mysql
Aplikasi pembuatan	Visual Studio Code, HTML
<i>FRAMEWORK</i>	<i>Laravel</i>

5.1.4 Instalasi Sistem

1. Instalasi Perangkat Lunak

a) XAMPP

XAMPP digunakan untuk menjalankan aplikasi Manajemen Perpustakaan pada sisi server lokal dengan memanggil *localhost* pada URL. Berikut tahapan dari instalasi XAMPP:

- 1) Unduh XAMPP pada situs (<https://www.apachefriends.org/>) dan unduh versi terbaru sesuai dengan sistem operasi Anda (Windows, mac OS, atau Linux).
- 2) Jalankan *installer* XAMPP yang baru diunduh. Ikuti setiap instruksi instalasinya.
- 3) Pilih komponen yang ingin diinstal.
- 4) Tentukan direktori instalasi, klik *next*.
- 5) Pilih bahasa, klik *next*, lalu tunggu hingga proses instalasi selesai.
- 6) Setelah proses instalasi selesai klik *finish*. XAMPP siap digunakan.

b) Instalasi *Google Chrome*

Google Chrome digunakan untuk menjalankan tampilan dan fungsionalitas dari aplikasi Manajemen Perpustakaan. Berikut tahapan instalasi *Google Chrome*:

- 1) Buka *web browser* saat ini (misalnya, *Internet Explorer* atau *Microsoft Edge*) lalu kunjungi situs resmi *Google Chrome*: <https://www.google.com/chrome/>
- 2) Di halaman tersebut, klik "Unduh *Chrome*" untuk mulai mengunduh.
- 3) Setelah proses unduhan selesai, cari file instalasi yang baru saja diunduh (biasanya bernama "*ChromeSetup.exe*") dan jalankan.
- 4) Akan ada peringatan pengguna kontrol akun (*User Account Control*). Klik "Ya" untuk memungkinkan *installer* menjalankan proses instalasi.
- 5) *Google Chrome* akan diinstal. *Google Chrome* siap untuk digunakan.

c) Instalasi *Visual Studio Code*

Visual Studio Code digunakan untuk implementasi program atau proses pengkodean menggunakan bahasa pemrograman. Berikut tahapan instalasi *Visual Studio Code*:

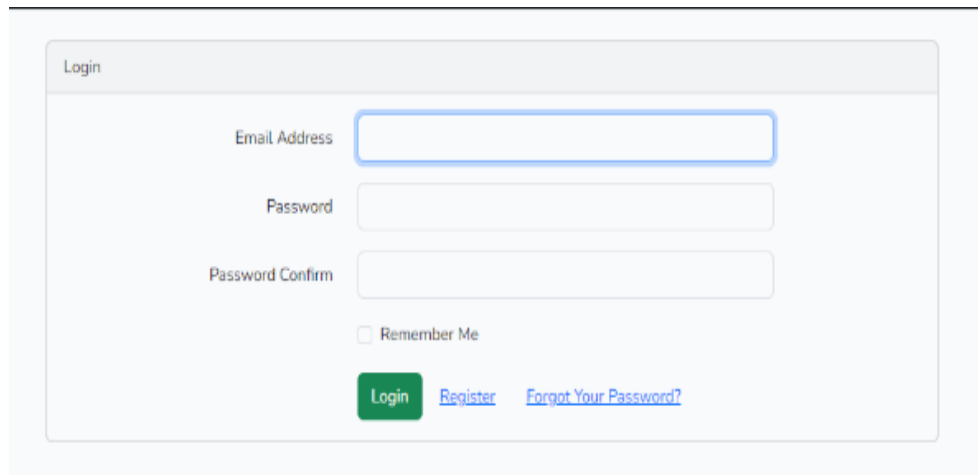
- 1) Buka *web browser* dan kunjungi situs resmi *Visual Studio Code*: <https://code.visualstudio.com/>
- 2) Di halaman tersebut, akan ada tombol "*Download for Windows*". Klik tombol tersebut untuk mengunduh *installer Visual Studio Code* untuk *Windows*.
- 3) Setelah unduhan selesai, cari file instalasi (biasanya bernama "*VSCodeSetup.exe*") lalu jalankan.
- 4) Ikuti instruksi pada layar untuk menyelesaikan instalasi.
- 5) Setelah instalasi selesai, *Visual Studio Code* siap digunakan.

5.1.5 Menjalankan Sistem

Bagian ini merupakan tampilan akhir dari aplikasi manajemen Perpustakaan yang sebelumnya telah dirancang berikut dengan penjelasannya.

1. Admin

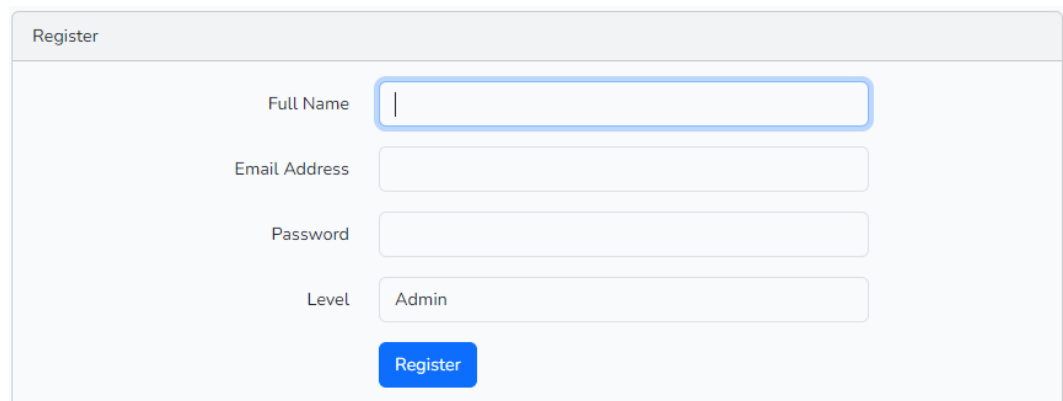
a Halaman Login admin

The image shows a web form titled "Login" in a light gray header. The form area is white and contains three input fields: "Email Address", "Password", and "Password Confirm". The "Email Address" field is highlighted with a blue border. Below the input fields is a checkbox labeled "Remember Me". At the bottom of the form, there is a green "Login" button, a blue "Register" link, and a blue "Forgot Your Password?" link.

Gambar 5. 1 Halaman Login

Sebelum masuk dan mengakses aplikasi, admin diarahkan ke tampilan Login yang berisi form login yang harus user isi dengan menginputan email dan password yang telah terdaftar. apabila belum memiliki user dapat melakukan register untuk membuat akun baru.

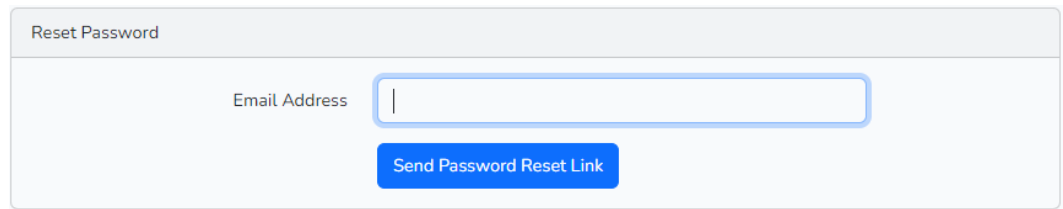
b Halaman Register

The image shows a web form titled "Register" in a light gray header. The form area is white and contains four input fields: "Full Name", "Email Address", "Password", and "Level". The "Full Name" field is highlighted with a blue border. The "Level" field has a dropdown menu with "Admin" selected. Below the input fields is a blue "Register" button.

Gambar 5. 2 Halaman register

Halaman registrasi merupakan halaman yang digunakan ketika pengguna belum memiliki akun dan ingin mendaftarkan akun nya agar dapat mengakses aplikasi dimana didalamnya pengguna akan di arahkan untuk mengeisi form yang akan menjadi akun pengguna untuk mengakses aplikasi.

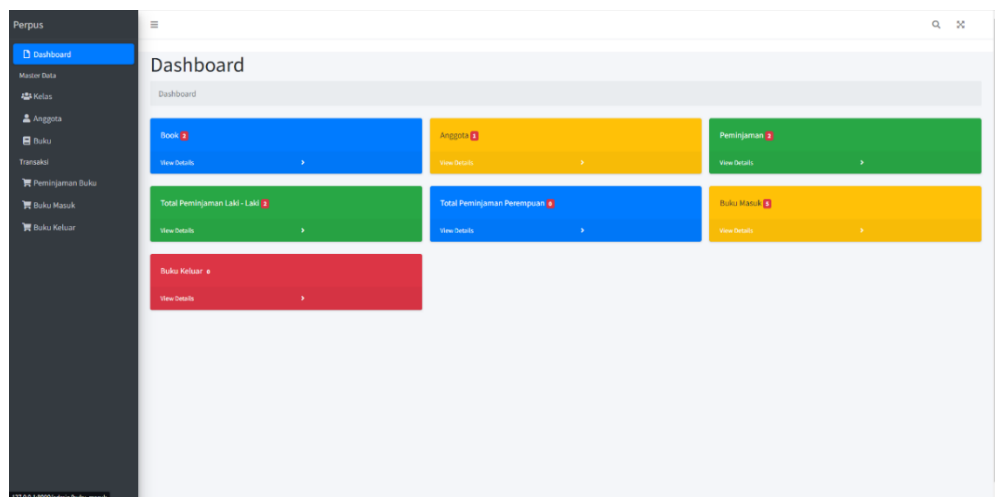
c Halaman Lupa Password



Gambar 5. 3 Halaman Lupa password

Halaman lupa password digunakan apabila user telah memiliki akun tetapi lupa password dari akunnya tersebut maka user harus mereset password dengan mengisi form yang mana user akan di suruh untuk menginput email yang nantinya reset password akan dikirim melalui email ke akun yang digunakan

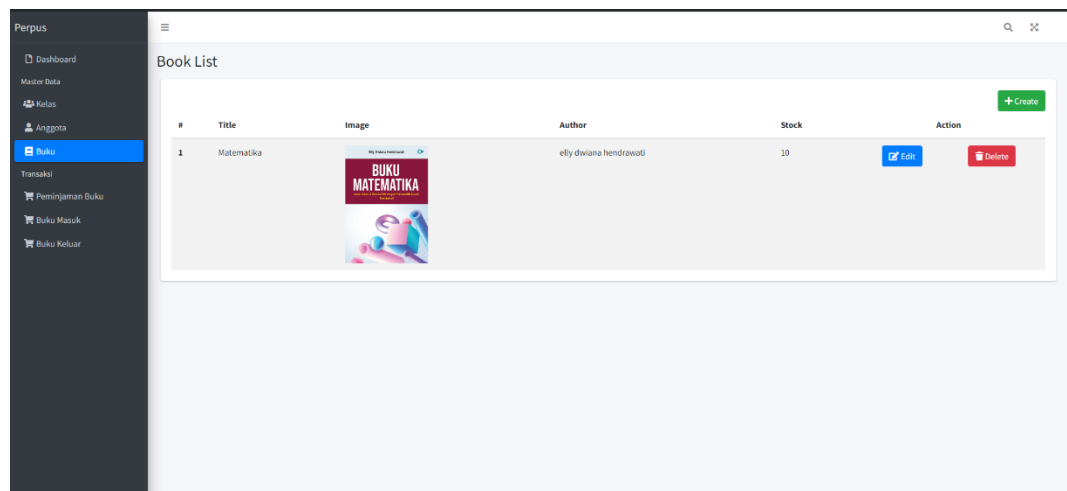
d Halaman Dashboard



Gambar 5. 4 Halaman Dashboard

Halaman dashboar ini merupakan halaman utama dari aplikasi manajemen perpustakaan yang mana apabila user telah berhasil login maka akan di arahkan kehalaman dashboard,halaman ini berisikan informasi informasi umum yang digunakan dalam pembuatan laporan seperti jumlah anggota,jumlah buku,jumlah peminjam laki laki,jumlah peminjam Perempuan jumlah buku masuk,jumlah dan jumlah buku keluar

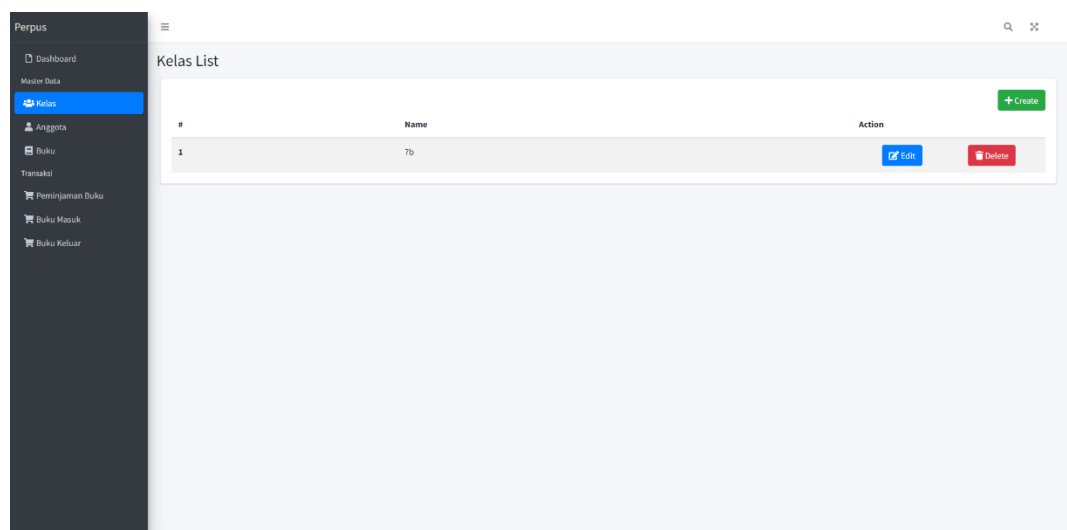
e Halaman Buku



Gambar 5. 5 Halaman Buku

Halaman buku ini merupakan halaman yang digunakan apabila user ingin melihat list buku yang ada berserta stoknya kemudian user juga dapat menambahkan atau mengurangi stok pada buku yang ada pada list selain itu user juga dapat mengedit buku yang ada di list apabila ada kesalahan data

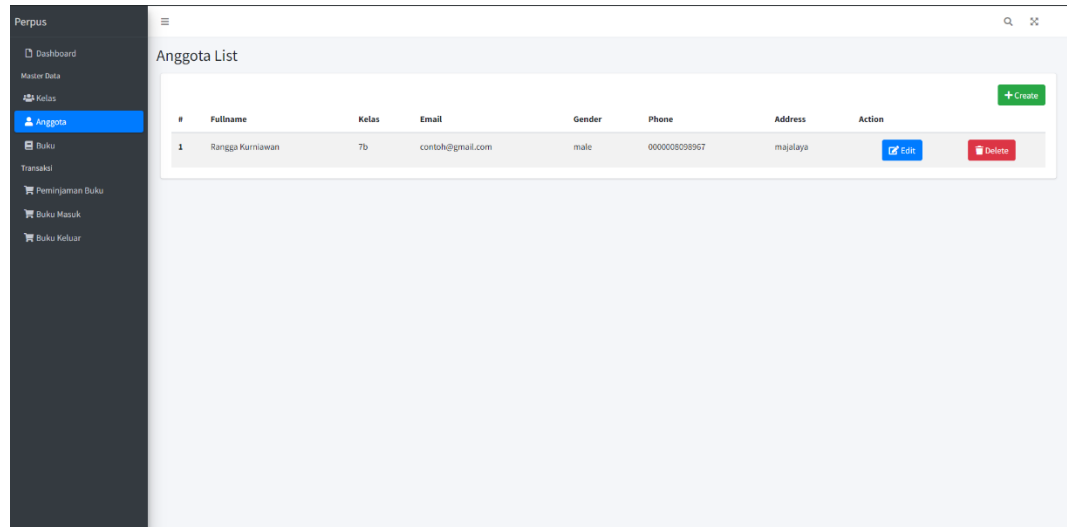
f Halaman Kelas



Gambar 5. 6 Halaman kelas

Halaman kelas ini merupakan halaman yang digunakan apabila user ingin membuat kelas yang nantinya digunakan untuk identitas anggota atau siswa selain itu data kelas juga bisa dihapus bila tidak digunakan ataupun di edit jika ada kesalahan

g Halaman Anggota

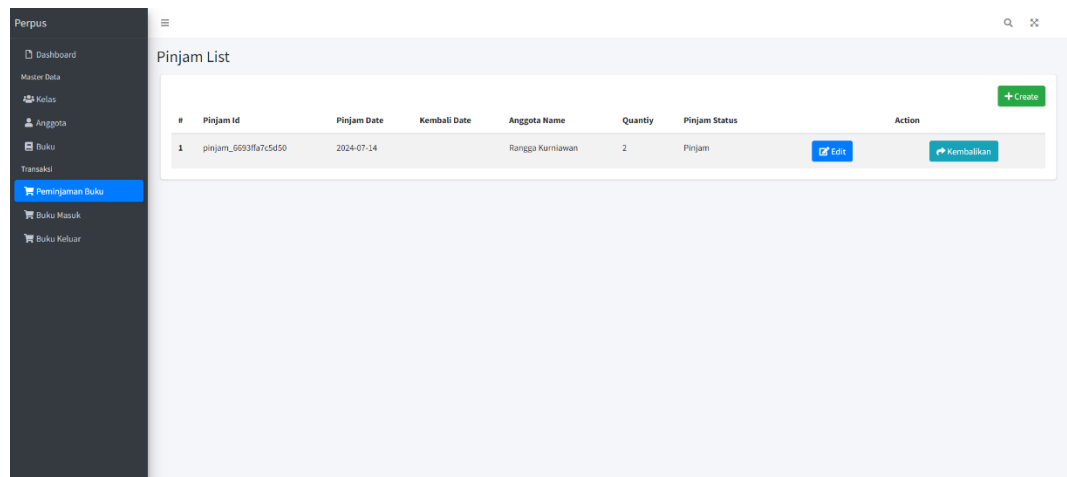


#	Fullname	Kelas	Email	Gender	Phone	Address	Action
1	Rangga Kurniawan	7b	contoh@gmail.com	male	0000008098967	majalaya	Edit Delete

Gambar 5. 7 Halaman Anggota

Halaman anggota ini merupakan halaman yang menunjukkan list anggota yang dapat meminjam buku di perpustakaan selain itu halaman ini juga bisa menambah atau menghapus siswa dari list anggota yang dapat meminjam buku di perpustakaan.

h Halaman Peminjaman

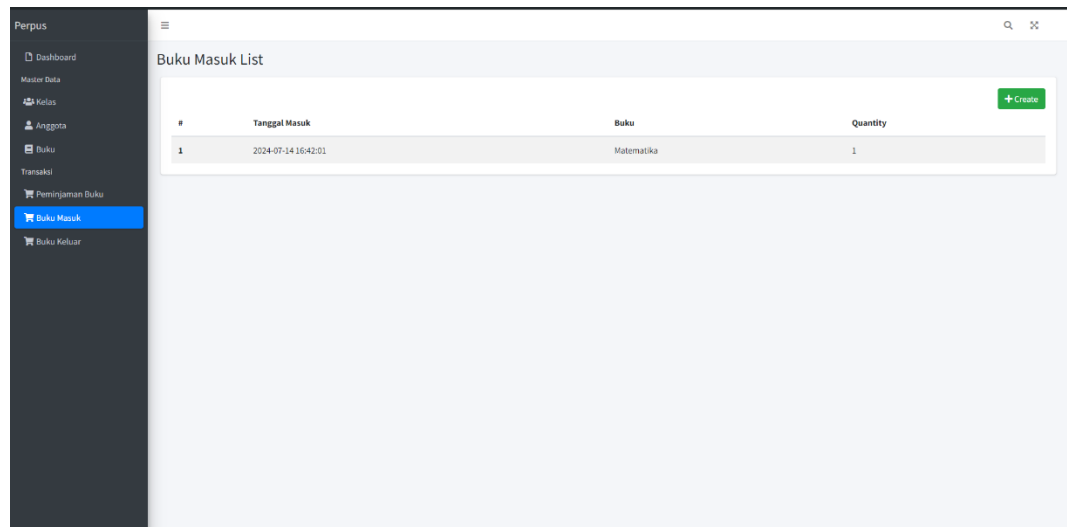


#	Pinjam Id	Pinjam Date	Kembali Date	Anggota Name	Quantly	Pinjam Status	Action
1	pinjam_6693ffa7c5d50	2024-07-14		Rangga Kurniawan	2	Pinjam	Edit Kembalikan

Gambar 5. 8 Halaman Peminjaman

Halaman peminjaman ini merupakan halaman yang menampilkan transaksi pinjam kembali buku yang ada di perpustakaan selain itu halaman ini juga digunakan apabila user ingin menuliskan peminjaman atau pengembalian buku.

i Halaman Buku Masuk

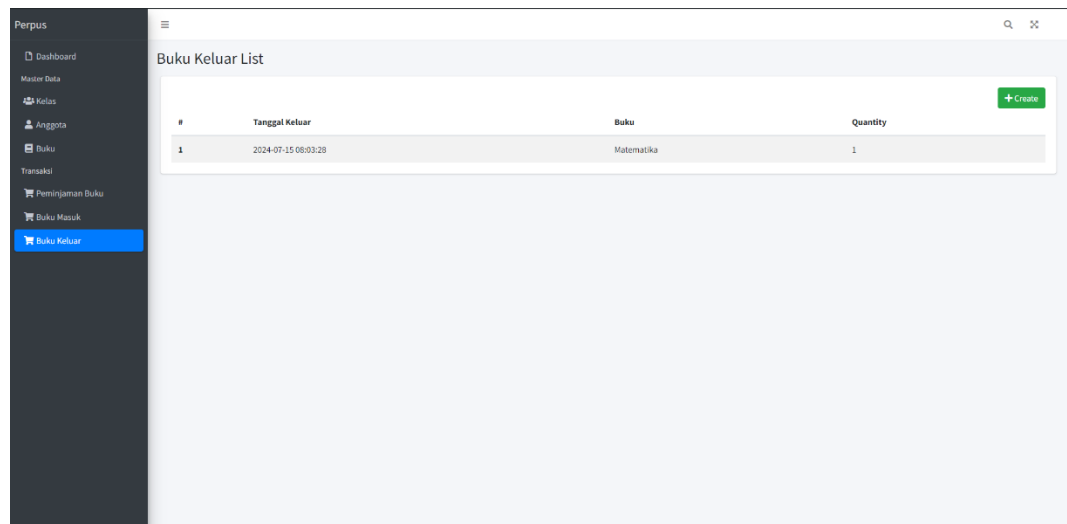


#	Tanggal Masuk	Buku	Quantity
1	2024-07-14 16:42:01	Matematika	1

Gambar 5. 9 Halaman buku Masuk

Halaman ini digunakan apabila ada buku baru tetapi buku tersebut sudah terdaftar di list buku perpustakaan atau bisa dikatakan digunakan apabila ada penambahan stok pada buku yang telah ada di perpustakaan

j Halaman Buku keluar



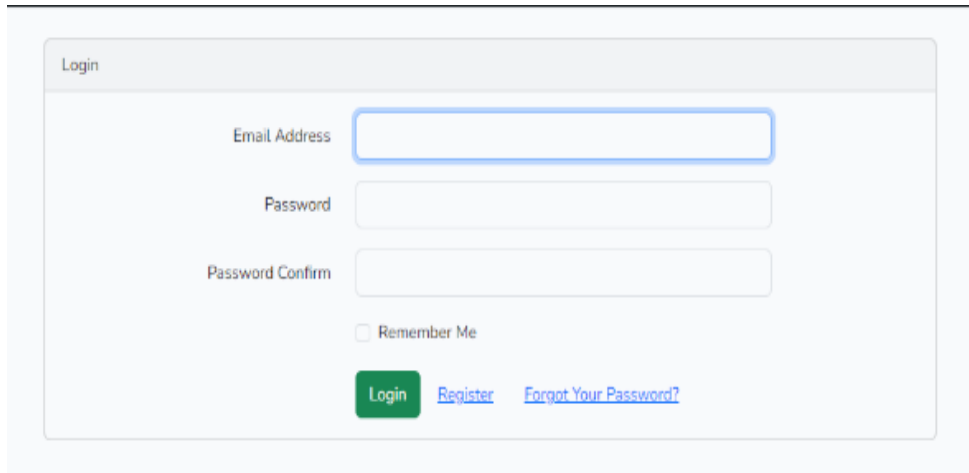
#	Tanggal Keluar	Buku	Quantity
1	2024-07-15 08:03:28	Matematika	1

Gambar 5. 10 Halaman Buku Keluar

Halaman ini digunakan apabila ada buku yang sudah tidak layak pakai tetapi buku tersebut sudah terdaftar di list buku perpustakaan atau bisa dikatakan digunakan apabila ada pengurangan stok pada buku yang telah ada di perpustakaan

2. User

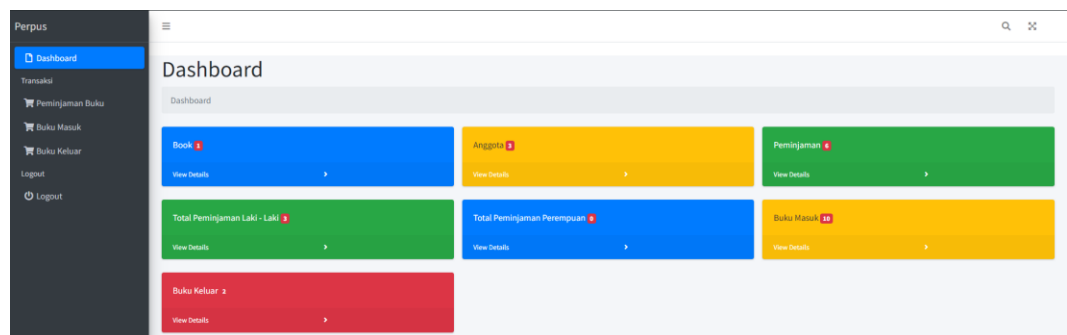
a Halaman Login User



Gambar 5. 11 Halaman Login User

Sebelum masuk dan mengakses aplikasi, user diarahkan ke tampilan Login yang berisi form login yang harus user isi dengan menginputkan email dan password yang telah terdaftar. apabila belum memiliki user dapat melakukan register untuk membuat akun baru.

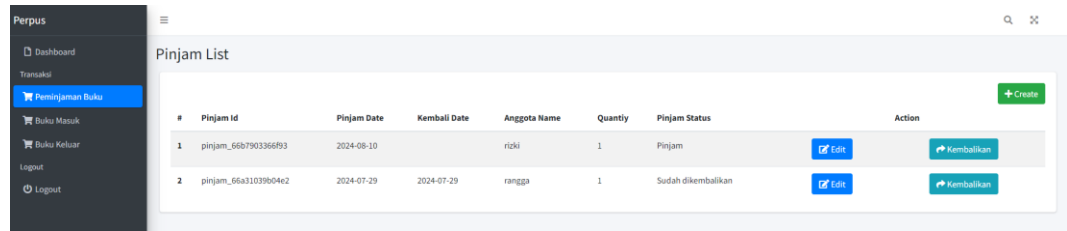
b Halaman dashboard



Gambar 5. 12 Halaman Dashboard User

Halaman dashboard ini merupakan halaman utama dari aplikasi manajemen perpustakaan yang mana apabila user telah berhasil login maka akan diarahkan ke halaman dashboard, halaman ini berisi informasi umum yang digunakan dalam pembuatan laporan seperti jumlah anggota, jumlah buku, jumlah peminjam laki-laki, jumlah peminjam Perempuan, jumlah buku masuk, jumlah dan jumlah buku keluar pada menu dashboard tidak berbeda dengan admin yang membedakan hanya jumlah menu yang dapat diakses.

c Halaman Peminjaman User



Gambar 5. 13 halaman peminjaman Anggota

Pada halaman ini untuk segi tampilan tidak berbeda dengan admin yang membedakan user hanya dapat melihat tidak bisa melakukan action

5.2 Pengujian Aplikasi

Tahap selanjutnya yaitu pengujian aplikasi. Pada tahap ini aplikasi diuji yang bertujuan untuk mengetahui apakah aplikasi yang dibuat berdasarkan perancangan telah sesuai dan berfungsi atau masih belum sesuai. Pada tahap pengujian ini dilakukan dengan menggunakan metode black box testing yaitu dengan memberikan beberapa skenario dengan hasil yang diharapkan kemudian dibandingkan dengan hasil pengujian nya apakah sudah sesuai atau belum.

Tabel 5. 3 Pengujian Aplikasi

NO	Item	Skenario	Hasil yang diharapkan	Hasil Pengujian
1	Form Login admin	admin melakukan registrasi kemudian melakukan login	admin berhasil registrasi kemudian berhasil login ke aplikasi	Valid
2	Halaman Buku	admin memilih menu buku kemudian memasukkan buku, menghapus buku, mengedit buku	admin berhasil masuk ke halaman buku, kemudian dapat menambahkan buku, menghapus buku dan mengedit buku	valid

3	Halaman Anggota	admin memilih menu anggota kemudian menambahkan anggota ,mengedit anggota kemudian menghapus anggota	Admin berhasil mengakses halaman anggota lalu user dapat menambahkan anggota,mengedit data anggota,kemudian menghapus anggota	valid
4	Halaman Kelas	Admin memilih menu kelas kemudian membuat kelas lalu menghapus kelas	Admin dapat mengakses halaman kelas kemudia membuat kelas baru dan menghapus kelas yang sudah ada	Valid
5	Halaman Peminjaman	Admin memilih menu peminjaman kemudian user melakukan peminjaman buku dan juga mengembalikan buku	Admin dapat mengakses halaman peminjaman kemudian melakukan peminjaman buku dan melakukan pengembalian buku	Valid
6	Halaman buku masuk	Admin memilih manu buku masuk kemudia menambahkan stok pada salah satu buku yang ada	Admin dapat mengakses menu buku masuk kemudian menambah stok pada salah satu buku	Valid

7	Halaman buku keluar	Admin memilih buku keluar kemudian menghapus atau mengurangi stok salah satu buku yang ada di list	Admin dapat mengakses halaman buku keluar kemudian dapat menghapus atau mengurangi stok salah satu buku	Valid
8	Halaman Login User	User melakukan registrasi kemudian melakukan login	User berhasil registrasi kemudian berhasil login ke aplikasi	valid
9	Halaman peminjaman user	User memilih menu peminjaman kemudian user melihat list peminjaman buku	User dapat mengakses halaman peminjaman melihat list peminjaman	valid

Dari hasil testing menggunakan metode *blackbox* testing dalam pengujian nya penulis melakukan testing dengan 7 item atau 7 fitur yaitu diantaranya ada halaman login,halaman buku,halaman anggota,halaman kelas,halaman peminjaman,halaman buku masuk dan halaman buku keluar dari ketujuh fitur yang di tes tersebut penulis membuat scenario kemudia ada hasil yang diharapkan kemudian membandingkannya dengan hasil yang di dapat dan dari 7 fitur yang di uji didapatkan hasil 7 valid atau bisa dikatakan fitur fitur yang di testing sesuai dengan harapan atau telah berjalan dengan baik

Tabel 5. 4 Hasil Uji oleh Petugas Perpustakaan

NO	Jenis Pengujian	Presentase Capaian
1	<i>User Interface</i>	100%
2	Fungsi Dasar Sistem	87,5%

3	Validasi	100%
	Rata rata keseluruhan aspek	95,83%

Tabel 5. 5 Hasil Uji Oleh Anggota

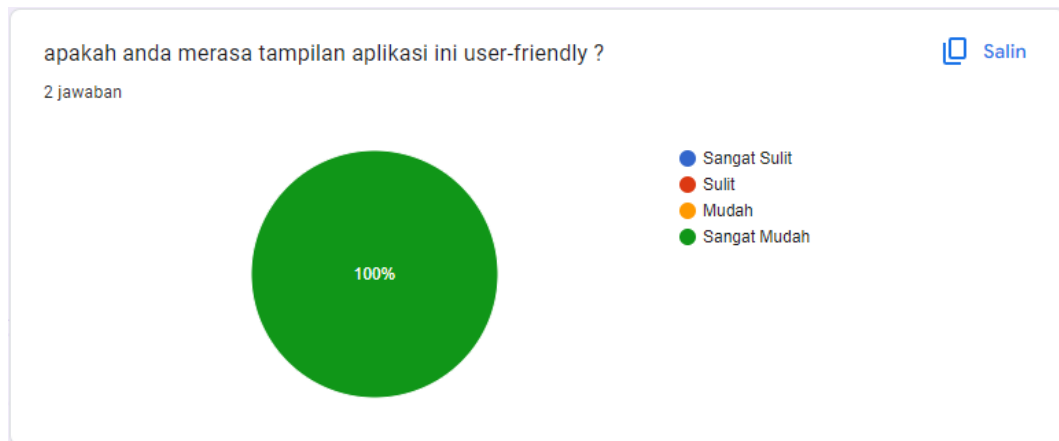
NO	Jenis Pengujian	Presentase Capaian
1	<i>User Interface</i>	100%
2	Fungsi Dasar Sistem	86,66%
3	Validasi	86,66%
	Rata rata keseluruhan aspek	91,10%

Berikut adalah data kuesioner yang berisi hasil pengujian validitas dan reliabilitas, yang akhirnya menghasilkan persentase capaian pengujian pada tabel sebelumnya. Data kuesioner dianalisis menggunakan metode statistik seperti Cronbach's Alpha untuk mengukur konsistensi internal kuesioner melalui grafik. Nilai yang tinggi menunjukkan konsistensi yang baik antara item-item kuesioner. Berikut adalah jawaban dari pertanyaan-pertanyaan

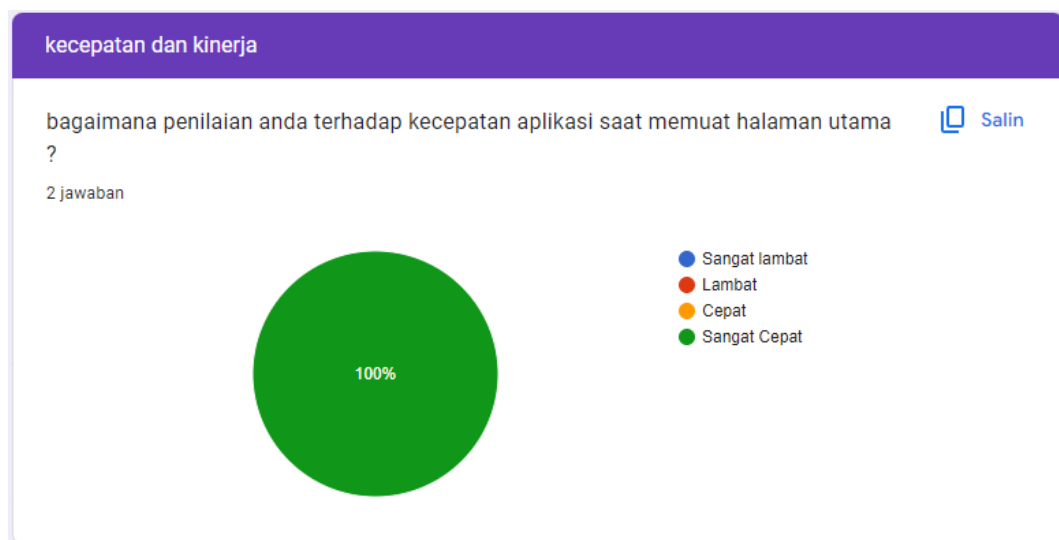
1. Data Kuisoner Penjaga Perpustakaan



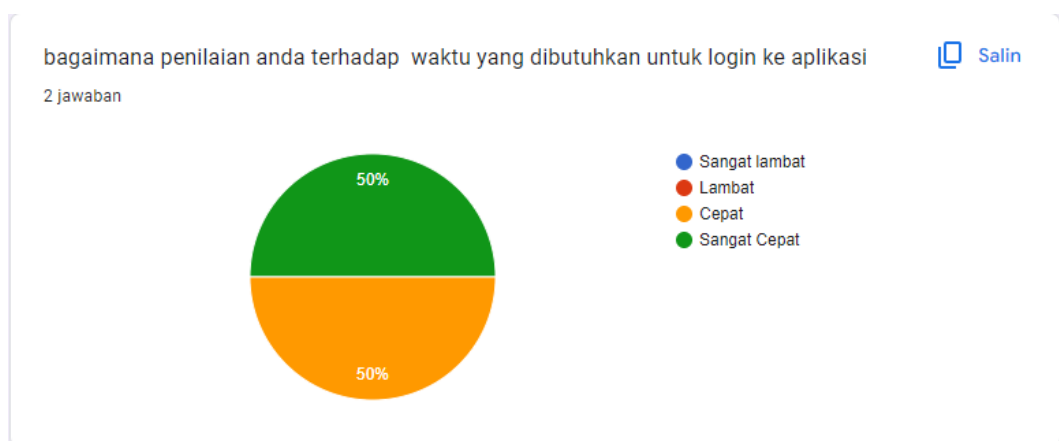
Gambar 5. 14 Grafik Kuesioner 1



Gambar 5. 15 Grafik kuesioner 2



Gambar 5. 16 Grafik Kueiisoner 3



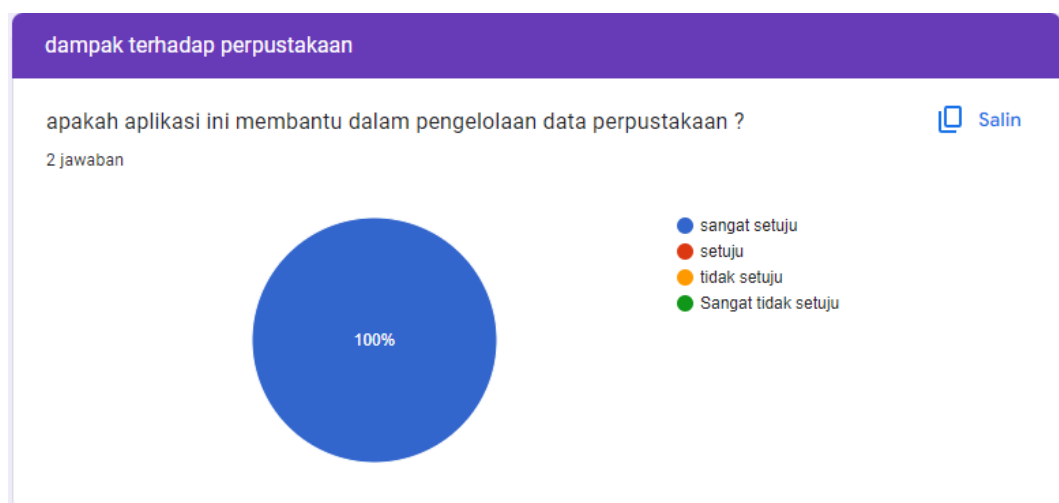
Gambar 5. 17 Grafik Kuesioner 4



Gambar 5. 18 Grafik kuesioner 5



Gambar 5. 19 grafik Kuesioner 6



Gambar 5. 20 Grafik kuesioner 7



Gambar 5. 21 Grafik Kuesioner 8

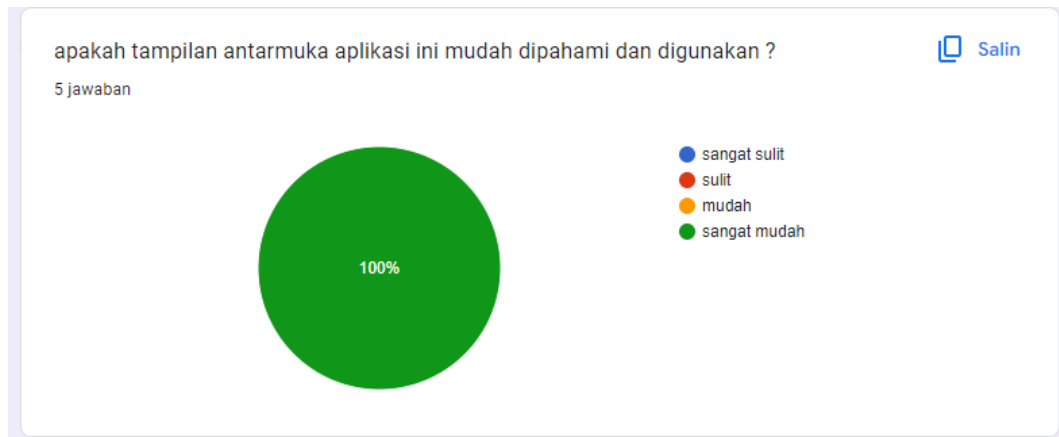


Gambar 5. 22 Grafik kuesioner 9

2. Data Kuisoner anggota



Gambar 5. 23 Grafik kuesioner 10



Gambar 5. 24 Grafik kuesioner 11



Gambar 5. 25 Grafik kuesioner 12



Gambar 5. 26 Grafik kuesioner 13



Gambar 5. 27 Grafik kuesioner 14



Gambar 5. 28 Grafik kuesioner 15



Gambar 5. 29 Grafik kuesioner 16



Gambar 5. 30 Grafik kuesioner 17



Gambar 5. 31 Grafik kuesioner 18

Hasil pengujian yang dilakukan oleh petugas perpustakaan menunjukkan bahwa capaian untuk user interface pengguna adalah 100% fungsi dasar sistem 87,5% validasi 100% dengan rata-rata seluruh aspek yang diukur sebesar 95,83%., pengujian yang dilakukan oleh anggota perpustakaan menunjukkan bahwa capaian untuk user interface pengguna adalah 100% fungsi dasar sistem 86,66% validasi 86,66% dengan rata-rata seluruh aspek yang diukur sebesar 91,10%. hasil pengujian menunjukkan bahwa aplikasi sudah berfungsi dengan baik, tetapi masih ada beberapa perbaikan yang diperlukan, terutama pada fungsi dasar sistem dan validasi.

BAB VI

KESIMPULAN

6.1 Kesimpulan

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

1. Adanya aplikasi manajemen perpustakaan diharapkan dapat membantu perpustakaan SMP KP 2 Majalaya dalam pengelolaan data buku dengan baik untuk meningkatkan efisiensi dalam mengelola buku yang ada di perpustakaan.
2. Adanya aplikasi manajemen perpustakaan diharapkan dapat meningkatkan
3. yang ada, jumlah buku yang baru ataupun buku buku yang telah rusak dan harus diganti.
4. Aplikasi manajemen perpustakaan ini diharapkan dapat membantu petugas perpustakaan dalam melacak inventaris seperti ada berapa jumlah buku dan apabila ada buku yang hilang petugas perpustakaan dapat mengetahui siapa yang meminjam nya dengan lebih mudah.
5. Aplikasi manajemen perpustakaan ini diharapkan dapat membantu petugas perpustakaan dalam pembuatan laporan karena dengan adanya aplikasi ini petugas perpustakaan lebih mudah untuk mengetahui informasi informasi yang dibutuhkan seperti jumlah buku, jumlah anggota, jumlah peminjaman laki laki dan Perempuan, serta ada berapa buku masuk dan keluar.
6. Hasil pengujian yang dilakukan oleh petugas perpustakaan menunjukkan bahwa capaian untuk seluruh aspek adalah 95,83% sedangkan pengujian terhadap anggota mendapatkan capaian untuk seluruh aspek sebesar 91.10%.

6.2 Saran

Untuk memaksimalkan hasil pada penelitian ini, diberikan saran agar aplikasi manajemen perpustakaan ini dapat lebih baik. Berikut merupakan saran yang diambil

1. Perlu adanya pelatihan kepada pengguna (Pengelola perpustakaan) agar dapat mengoperasikan dan mengembangkan aplikasi manajemen perpustakaan agar lebih baik lagi kedepannya.
2. Menambah keamanan sistem pada agar data terlindungi dari pihak yang tidak berwenang.dan membuat halaman registrasi yang berbeda untuk admin dan user
3. Perlu adanya evaluasi dan pengembangan aplikasi secara berkala untuk meningkatkan kualitas dan layanan.
4. Perlu adanya pemeliharaan secara berkala yang dilakukan oleh instansi
5. Perlu adanya pengembangan aplikasi lebih lanjut seperti membuat halaman registrasi yang berbeda untuk admin dan user

DAFTAR PUSTAKA

- Adivia Gilang Prakarsa, Ari Sujarwo., “Pemanfaatan Vue Js Pada Fitur Pengaturan Tambak Dalam Aplikasi Budi Daya Tambak Jala” . Automata, Vol 2, No. 1, 2021
- Agus Muhyidin, M., Sulhan, M. A., & Sevtiana, A. (2020). *PERANCANGAN UI/UX APLIKASI MY CIC LAYANAN INFORMASI AKADEMIK MAHASISWA MENGGUNAKAN APLIKASI FIGMA* (Vol. 10, Issue 2). <https://my.cic.ac.id/>.
- Asnur, Rosita, Bernadus Very Christioko, and Info Artikel. n.d. “Sistem Informasi Perpustakaan Berbasis Web Menggunakan *Framework* Codeigniter” 3 (2): 1–6. <https://doi.org/10.26623/jisl>.
- Basuki, S. (2003). Pengantar Ilmu Perpustakaan, Jakarta: Universitas Terbuka. Depdikbud.
- Carolus Ningki , Noviyanti.P (2023).Implementasi Aplikasi Penjualan Produk Tradisional Berbasis Website Menggunakan Metode *Waterfall*. JURNAL INFORMATIK Edisi ke-19, Nomor 2
- Dewi Wulan Dari, A. O. (2019). Rancang Bangun Aplikasi perpustakaan Berbasis Website. *Jurnal Ilmu Pengetahuan Dan teknologi komputer*
- Firdaus, R., Sakethi, D., & Rosman, F. (2015).
- Hartono. (2016). Manajemen Perpustakaan Sekolah: Menuju Perpustakaan Modern dan Profesional. Yogyakarta: Ar-Ruzz Media.
- Hendrawan¹, E., Meisel², M., & Sari³, D. N. (2022). ANALYSIS AND IMPLEMENTATION OF COMPUTER NETWORK SYSTEMS USING SOFTWARE DRAW.IO. In *Asia Information System Journal* (Vol. 2, Issue 1).
- <http://ejournal.radenintan.ac.id/index.php/AISJ/index://creativecommons.org/licenses/by-sa/4.0/>

- Herdiana, Y., & Nursalam, A. I. (n.d.). STUDI KASUS PRODI TEKNIK INFORMATIKA FTI UNIBBA. In Jurnal Informatika-COMPUTING (Vol. 07).
- INFORMATIKA, C., & yudi herdiana. (2020). MEMBUAT APLIKASI BIMBINGAN SKRIPSI ONLINE UNTUK MEMPERMUDAH BIMBINGAN SKRIPSI (STUDI KASUS PRODI TEKNIK INFORMATIKA FTI UNIBBA) . COMPUTING | Jurnal Informatika, 7(02).Retrievedfrom <https://ejournal.unibba.ac.id/index.php/computing/article/view/853>
- INFORMATIKA, C., Iyus A muslimin, & Denny Rusdianto. (2022). PEMBUATAN APLIKASI PENJUALAN BINATANG TERNAK BERBASIS ANDROID DI PETERNAKAN BEBEK ALFALAH DESA PADAULUN. COMPUTING | Jurnal Informatika, 9(02), 80– 89. Retrieved from x <https://ejournal.unibba.ac.id/index.php/computing/article/view/102>
- INFORMATIKA, C., & Iyus A Muslimin. (2021). MEMBANGUN APLIKASI PEMBUATAN SURAT KETERANGAN KULIAH, SURAT PENGANTAR PENELITIAN DAN PENGAJUAN CUTI BERBASIS WEB. COMPUTING | Jurnal Informatika, 8(02), 1–7. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/698>
- INFORMATIKA, C., Khilda Nistrina, & Sukiman. (2022). MEMBANGUN APLIKASI SENSUS KEPENDUDUKAN BERBASIS WEB MENGGUNAKAN METODE *WATERFALL* DI DESA NEGLASARI. COMPUTING | Jurnal Informatika, 9(01), 44–50. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/862>
- INFORMATIKA, C., & Khilda Nistrina. (2021). RANCANG BANGUN GAME EDUKASI SEBAGAI MEDIA PEMBELAJARAN BERBASIS ANDROID MENGGUNAKAN ALGORITMA FISHER YATES. COMPUTING | Jurnal Informatika, 8(02), 8–13. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/699>
- INFORMATIKA, C., & Nurul Imamah. (2021). PERANCANGAN SISTEM MONITORING DAN PENGENDALIAN LAMPU MENGGUNAKAN SENSOR GERAK DAN SENSOR CAHAYA DILENGKAPI INTERNET

- OF THINGS (IOT). COMPUTING | Jurnal Informatika, 8(02), 14–21.
<https://ejournal.unibba.ac.id/index.php/computing/article/view/700>
- INFORMATIKA, C., Nurul Imamah, sutyono WP, & Ari Reynaldi. (2022). PERANCANGAN DAN IMPLEMENTASI SISTEM MONITORING KEAMANAN TOKO BERBASIS INTERNET OF THINGS (IOT) DENGAN MENGGUNAKAN MIKROKONTROLER ESP32-CAM (STUDI KASUS DI TOKO STELIOS AQUATIC). COMPUTING | Jurnal Informatika, 9(02), 70–79. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/1025>
- INFORMATIKA, C., & Nurul Imamah. (2021). PERANCANGAN SISTEM MONITORING DAN PENGENDALIAN LAMPU MENGGUNAKAN SENSOR GERAK DAN SENSOR CAHAYA DILENGKAPI INTERNET OF THINGS (IOT). COMPUTING | Jurnal Informatika, 8(02), 14–21. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/700> xi
- INFORMATIKA, C., & Nurul Imamah. (2021). PERBANDINGAN ALGORITMA SEQUENTIAL SEARCH DAN ALGORITMA BINARY SEARCH PADA APLIKASI KAMUS BAHASA INDONESIA MENGGUNAKAN PHP DAN JQUERY . COMPUTING | Jurnal Informatika, 8(01), 1–6. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/564>
- INFORMATIKA, C., & Rosmalina. (2022). SISTEM INFORMASI AKADEMIK BERBASIS WEB MENGGUNAKAN BOOTSTRAP DI SMP AMS PAMEUNGPEUK: SISTEM INFORMASI AKADEMIK BERBASIS WEB MENGGUNAKAN BOOTSTRAP DI SMP AMS PAMEUNGPEUK. COMPUTING | Jurnal Informatika, 9(01), 21– 25. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/857>
- INFORMATIKA, C., & Rosmalina. (2021). PENERAPAN ALGORITMA SAW (SIMPLE ADDITIVE WEIGHTING) PADA MODUL APLIKASI SPK DI MA AL-AZHAR MAJALAYA BERBASIS WEB. COMPUTING | Jurnal

- Informatika, 8(01), 7–15. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/565>
- INFORMATIKA, C., & Rosmalina. (2021). APLIKASI PENGELOLAAN KAMAR INDEKOS BERBASIS WEB DI KAMAR INDEKOS ALNAJAH CIPARAY. *COMPUTING | Jurnal Informatika*, 8(02), 22– 28. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/701>
- INFORMATIKA, C., & Rustiyana. (2022). IMPLEMENTASI WEBSITE SEBAGAI MEDIA MEMBUAT TEMPLATE KONFIGURASI PERANGKAT JARINGANTELEKOMUNIKASI DI PT ICON+. *COMPUTING | Jurnal Informatika*, 9(01), 26–30. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/858>
- INFORMATIKA, C., & Rustiyana. (2021). IMPLEMENTASI ALGORITMA FISHER YATES SHUFFLE DALAM PEMBUATAN UJIAN ONLINE BERBASIS WEB . *COMPUTING | Jurnal Informatika*, 8(01), 16–21. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/566>
- INFORMATIKA, C., Rustiyana, Khilda Nistrina, Sukiman, & Sandhy Dwi A. (2023). PERANCANGAN DAN IMPLEMENTASI TEMPAT SAMPAH OTOMATIS BERBASIS ARDUINO UNO MENGGUNAKAN ALGORITMA FUZZY LOGIC. *COMPUTING | Jurnal Informatika*, 10(01), 21–25. Retrieved from <https://ejournal.unibba.ac.id/index.php/computing/article/view/1146>
- Kadir, A. (2009). *Mastering AJAX dan PHP*. Yogyakarta: CV. Andi Offset.
- Kurniadi, Rahman, Cecep Riki, and Milah Nurkamilah. 2022. “Rancang Bangun Aplikasi Perpustakaan Berbasis Web Dengan Menggunakan *Framework CodeIgniter*.” *Formosa Journal of Science and Technology* 1 (5): 507–18. <https://doi.org/10.55927/fjst.v1i5.1209>.
- Lararenza, E. (2020). *PHP Adalah dari Bahasa Pemograman, Berikut Penjelasan Selengkapnya*.
- Nugraha, F. (2014). ANALISA DAN PERANCANGAN SISTEM INFORMASI PERPUSTAKAAN. *Jurnal SIMETRIS*, vol 5 No 1, 27-32.

- Rancang Bangun Sistem Informasi Perpustakaan Berbasis Web. *Jurnal Komputasi*, 3, 85-94.
- Supriatiningsih. (2020). Implementasi Metode *Waterfall* Pada Aplikasi Perpustakaan Berbasis Web. *IJSE-Indonesian Journal Software engineering*.
- Saputra, P. A., & Nugroho, A. (2017). PERANCANGAN DAN IMPLEMENTASI Sopandi, Hendra, and Novita Anggraini. n.d. "APLIKASI PERPUSTAKAAN BERBASIS WEB MENGGUNAKAN *FRAMEWORK* CODEIGNITER (STUDI KASUS: SMPN 3 PACET)." Vol. 18. <https://journal.fkom.uniku.ac.id/ilkomTerakreditasiSINTA5>.
- SURVEI KEPIJASAN PENGUNJUNG BERBASIS WEB DI PERPUSTAKAAN DAERAH KOTA SALATIGA. *Jurnal Ilmiah Teknologi Informasi*, Volume 15, 63-71
- Syafi'i, M. (2004). *Membangun Aplikasi Berbasis PHP dan MySQL*. Yogyakarta:

LAMPIRAN

Lampiran 1 : Dokumentasi



Lampiran 2 : Wawancara

Pada kesempatan ini penulis mendapatkan kesempatan untuk mewawancarai narasumber yang bertugas terkait pengelolaan perpustakaan SMP KP 2 Majalaya yaitu dengan bapak Tedi Setiadi Si.Pust selaku pustawakan di SMP KP 2 majalaya yang berlokasi di Jalan Wangisagara,Desa Wangisagara,Kecamatan majalaya,kabupaten Bandung.pada wawancara kali ini penulis menanyakan beberapa hal diantaranya sebagai Berikut :

No	Pertanyaan	Jawaban
1	Apa permasalahan utama yang dihadapi dalam pengelolaan perpustakaan	Dari hasil wawancara dengan narasumber didapatkan informasi mengenai masalah masalah yang dihadapi oleh pustakawan dalam mengelola perpustakaan yaitu keakuratan data yang rendah mengenai stok atau ketersediaan buku,terkadang terjadi kehilangan buku serta pustakawan kesulitan dalam pembuatan laporan secaracepat
2	Bagaimana sistem pembukuan perpustakaan di smp kp 2 Majalaya?	Dari hasil wawancara dengan narasumber diketahui bahwa Sistem pembukuan perpustakaan di smp kp 2 majalaya masih menggunakan cara manual yaitu dengan cara peminjam buku yang menuliskan buku apa yang dia pinjam dan berapa jumlah buku yang di pinjam
3	Apa tantangan utama yang dihadapi dalam pencatatan perpustakaan secara manual?	Berdasarkan hasil wawancara dengan narasumber diketahui bahwa Proses pembukuan yang manual terkadang menimbulkan suatu kesalahan serta ketidak akuratan data kemudian kesulitan dalam mengetahui siapa yang

		meminjam kemudian seringkali terjadi keterlambatan
4	Mengapa pencatatan perpustakaan secara manual dapat menimbulkan masalah ?	Dari hasil wawancara dengan narasumber di dapatkan bahwa pencatatan secara manual dapat menimbulkan masalah karena pencatatan secara rentan akan <i>human error</i> seperti kesalahan dalam menulis nama atau tulisan yang dari peminjam yang sulit dibaca ataupun kadang buku pencatatan peminjaman yang robek atau terkena air yang membuat data data yang ada hilang.
5	Apa telah ada Upaya dalam mengatasi masalah tersebut	Dari hasil wawancara narasumber mengatakan bahwa pihak perpustakaan pernah menyediakan kartu perpustakaan kepada setiap siswa jadi pada saat siswa atau siswi meminjam buku di perpustakaan harus menggunakan kartu agar dapat mempermudah dalam mengetahui siapa yang meminjam buku

Penjaga Perpustakaan

Tedi Setiadi Si.Pust

Lampiran 3 : TOR (Term Of Reference)

Sebelum Melaksanakan penelitian skripsi penulis melakukan beberapa metode penelitian yaitu observasi, wawancara, dan studi pustaka. Setelah mengamati dan mempelajari lokasi penelitian yang telah ditentukan dan disetujui oleh instansi tempat penelitian yaitu di Perpustakaan SMP KP 2 Majalaya yang berlokasi di jl-Wangisagara,desa Wangisagara,Kecamatan Majalaya,Kabupaten Bandung Provinsi Jawa Barat Perpustakaan SMP KP 2 Majalaya menghadapi kendala dalam pengelolaan data Buku, sehingga diperlukan sistem yang dapat meningkatkan efisiensi Pengelolaan Data buku. Penelitian ini bertujuan untuk membangun aplikasi Pengelolaan perpustakaan dengan menggunakan *Framework Laravel*.dengan memanfaatkan aplikasi tersebut,pengelolaan perpustakaan akan lebih efektif karena pustakawan dapat dengan cepat mendapatkan informasi mengenai ketersediaan inventaris, meningkatkan akurasi data, dan membuat laporan secara cepat. Bahasa pemrograman yang digunakan yaitu PHP, HTML, dan database MySQL, dengan penerapan *Framework Laravel* dan Bootstrap. Untuk memastikan fokus penelitian adapun batasan masalah yang ditetapkan adalah sebagai berikut:

1. Fokus pada pembangunan aplikasi manajemen perpustakaan berbasis web.
2. Penggunaan *Framework Laravel* dalam pembuatan aplikasi
3. Metode yang digunakan dalam penelitian ini adalah *Waterfall*.
4. Aplikasi mencakup fitur peminjaman dan pengembalian buku, penambahan dan penghapusan data buku,penambahan dan penghapusan data anggota dan menampilkan laporan secara real time.
5. Pengujian aplikasi dilakukan secara simulasi atau pada lingkungan uji coba

Bandung,Mei 2024

Disetujui Oleh :

Mahasiswa

Pustakawan

Rangga Kurniawan

Tedi Setiadi Si.Pust

Lampiran 4 : Hasil Kuesioner

Kuesioner petugas perpustakaan

Q1 = seberapa mudah petugas melayani anggota melalui aplikasi?

Q2 = apakah anda merasa tampilan aplikasi ini user-friendly ?

Q3 = bagaimana penilaian anda terhadap kecepatan aplikasi saat memuat halaman utama ?

Q4 = bagaimana penilaian anda terhadap waktu yang dibutuhkan untuk login ke aplikasi?

Q5 = apakah anda pernah mengalami aplikasi ini tiba tiba berhenti atau crash saat digunakan ?

Q6 = apakah anda merasa aplikasi ini berjalan lancar saat berpindah dari satu fitur ke fitur lainnya ?

Q7 = apakah aplikasi ini membantu dalam pengelolaan data perpustakaan ?

Q8 = apakah aplikasi ini membantu dalam memonitor dan mengelola peminjaman buku yang dilakukan?

Q9 = apakah aplikasi ini membantu dalam meningkatkan kepuasan anggota ?

Nama	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
reni rohayani	Sangat Mudah	Sangat Mudah	Sangat Cepat	Cepat	Tidak pernah	sangat setuju	sangat setuju	sangat setuju	sangat setuju
Tedi Setiadi	Sangat Mudah	Sangat Mudah	Sangat Cepat	Sangat Cepat	Tidak pernah	sangat setuju	sangat setuju	sangat setuju	sangat setuju

Kuesioner User

Q1 : seberapa mudah anda menemukan informasi yang anda butuhkan dalam aplikasi ini ?

Q2 : apakah tampilan antarmuka aplikasi ini mudah dipahami dan digunakan ?

Q3 : apakah aplikasi ini memudahkan anda dalam melihat data peminjaman dan pengembalian buku yang anda lakukan ?

Q4 : apakah anda mengalami kesulitan dalam mengakses aplikasi ini ?

Q5 : apakah anda merasa aplikasi ini konsisten dalam kinerja dan kecepatan setiap kali digunakan ?

Q6 : seberapa puas anda dengan keseluruhan aplikasi ini ?

Q7 : seberapa jelas informasi yang ditampilkan di aplikasi ?

Q8 : apakah transaksi peminjaman dan pengembalian buku yang anda lakukan tercatat dengan jelas ?

Q9 : seberapa sering anda menemukan kesalahan dalam pencatatan peminjaman yang anda lakukan ?

Nama	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Muhammad Iqbal	sangat mudah	sangat mudah	sangat setuju	tidak pernah	selalu	sangat puas	sangat jelas	sangat jelas	sering
saepul sobana	sangat mudah	sangat mudah	sangat setuju	kadang kadang	selalu	sangat puas	sangat jelas	sangat jelas	tidak pernah
aripin rustandi	sangat mudah	sangat mudah	sangat setuju	tidak pernah	selalu	sangat puas	sangat jelas	sangat jelas	tidak pernah
buki darmawan	sangat mudah	sangat mudah	sangat setuju	tidak pernah	selalu	sangat puas	sangat jelas	sangat jelas	tidak pernah
sagara muslim	sangat mudah	sangat mudah	sangat setuju	kadang kadang	selalu	sangat puas	sangat jelas	sangat jelas	jarang

Lampiran 5 : Listing Program

1. Anggota

```

public function index()
{
    //ambil semua data last lalu bagi menjadi 5 data setiap
    page
    $anggota
    Anggota::join('kelas','anggota.kelas_id','=','kelas.kelas_id')-
    >select([
    'anggota.anggota_id','anggota.fullname','anggota.email','anggota.
    gender','anggota.phone','anggota.address','kelas.name          as
    kelas_name'])->paginate(5);
    // kembalikan halaman view anggota list dengan mengirim
    datanya
    return view('anggota_list',compact('anggota'))
    ->with('i',(request()->input('page',1)-1)*5);
}

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    $kelas = Kelas::all();
    //tampilkan halaman add anggota
    return view('anggota_add',compact('kelas'));
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    //
    try {
        // menjalankan fungsi insert pada table anggota
        Anggota::create($request->all());
        // redirect ke halaman list anggota
        return redirect()->route('anggota.index')-
        >with('success','Successfully to create new anggota');
    } catch (\Throwable $th) {
        //throw $th;
        // munculkan pesan error jika ada error
        return redirect()->route('anggota.index')-
        >with('error',$th->getMessage());
    }
}

/**
 * Display the specified resource.
 */
public function show(string $id)
{
    //
}

```

```

/**
 * Show the form for editing the specified resource.
 */
public function edit($id)
{
    //munculkan data anggota sesuai parameter id dan ambil satu
data
    $anggota = Anggota::where('anggota_id',$id)-
>firstOrFail();
    $kelas = Kelas::all();
    // jika ada data anggota
    if($anggota){
        // buka halaman view anggota_edit dengan mengirim
datanya
        return
view('anggota_edit',compact('anggota','kelas'));
    }else{
        return redirect()->route('anggota.index')-
>with('error','anggota not found');
    }
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, $id)
{
    //tambahkan validasi
    //ambil data anggota sesuai parameter id dan lakukan update
pada modelnya
    Anggota::where('anggota_id',$id)->update([
        'fullname'=> $request->fullname,
        'email'=> $request->email,
        'gender'=> $request->gender,
        'address'=> $request->address,
        'kelas_id'=> $request->kelas_id
    ]);

    return redirect()->route('anggota.index')-
>with('success','Successfully update data');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy($id)
{
    //lakukan delete pada data anggota sesuai parameter id
    Anggota::where('anggota_id',$id)->delete();

    return redirect()->route('anggota.index')-
>with('success','Successfully delete data');
}

```

```

    public function getAnggotaById(Request $request){

        //ajax akan meminta request untuk mengambil data anggota
        berdasarkan parameter
        $anggota = Anggota::where('anggota_id',$request->id)-
        >firstOrFail();
        // kembalikan data dalam bentuk response json
        return response()->json([
            'anggota'=>$anggota
        ]);
    }
}

```

2. Buku

```

public function index()
{
    //ambil semua data buku latest lalu bagi menjadi 5 data
    setiap page
    $book = Book::latest()->paginate(5);
    // kembalikan halaman view buku list dengan mengirim
    datanya
    return view('book_list',compact('book'))
    ->with('i',(request()->input('page',1)-1)*5);
}

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    //
    return view('book_add');
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    //
    try {
        // fungsi dibawah digunakan untuk mengambil nama file
        $imageName = $request->image-
        >getClientOriginalName();
        // fungsi move untuk mengupload file ke lokal folder
        public
        $request->image-
        >move(public_path('images'),$imageName);

        Book::create([
            'title'=>$request->title,
            'isbn13'=>$request->isbn13,
            'num_pages'=>$request->num_pages,
            'image'=>$imageName,
            'author'=>$request->author,

```

```

        'stock'=>$request->stock

    });

    return redirect()->route('book.index')-
>with('success','Successfully to create new book');
    } catch (\Throwable $th) {
        //throw $th;
        return redirect()->route('book.index')-
>with('error',$th->getMessage());
    }
}

/**
 * Display the specified resource.
 */
public function show(string $id)
{
    //
}

/**
 * Show the form for editing the specified resource.
 */
public function edit($id)
{
    //
    $book = Book::where('book_id',$id)->firstOrFail();

    if($book){

        return view('book_edit',compact('book'));
    }else{
        return redirect()->route('book.index')-
>with('error','book not found');
    }

}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, $id)
{
    if(!$request->image){
        Book::where('book_id',$id)->update([
            'title'=> $request->title,
            'num_pages'=> $request->num_pages,
            'author'=> $request->author,
            'stock'=> $request->stock

        ]);
    }else{
        $imageName = $request->image-
>getClientOriginalName();
    }
}

```



```

        $request->image->
>move(public_path('images'),$imageName);

        Book::where('book_id',$id)->update([
            'title'=> $request->title,
            'image'=> $imageName,
            'num_pages'=> $request->num_pages,
            'author'=> $request->author,
            'stock'=> $request->stock
        ]);
    }
    //

    return redirect()->route('book.index')->
>with('success','Successfully update data');

}

/**
 * Remove the specified resource from storage.
 */
public function destroy($id)
{
    //
    Book::where('book_id',$id)->delete();

    return redirect()->route('book.index')->
>with('success','Successfully delete data');
}

public function getPriceById(Request $request){
    $book = Book::where('book_id',$request->id)->
>firstOrFail();

    return response()->json([
        'book'=>$book
    ]);
}
}

```

3. Buku Keluar

```

public function index()
{
    //ambil semua data last lalu bagi menjadi 5 data setiap
    page
    $buku_keluar =
    BukuKeluar::join('book','buku_keluar.book_id','=','book.book_id')
->select(['book.title as
title','buku_keluar.created_at','buku_keluar.quantity'])->
>paginate(5);
    // kembalikan halaman view buku_keluar list dengan mengirim
    datanya
    return view('buku_keluar_list',compact('buku_keluar'))
->with('i',(request()->input('page',1)-1)*5);
}

```

```

    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        $book = Book::all();
        //tampilkan halaman add buku_keluar
        return view('buku_keluar_add',compact('book'));
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        // Menjalankan validasi untuk memeriksa stok buku
        $book = Book::findOrFail($request->book_id); //
        Menggunakan model buku yang Anda miliki, ganti dengan model yang
        sesuai
        if ($book->stock < $request->quantity) {
            return redirect()->route('buku_keluar.index')->with('error', 'Stock not enough');
        }

        try {
            // Menjalankan fungsi insert pada table buku keluar
            BukuKeluar::create($request->all());
            // Mengurangi stok buku yang keluar dari stok total
            $book->stock -= $request->quantity;
            $book->save();

            // Redirect ke halaman list buku keluar
            return redirect()->route('buku_keluar.index')->with('success','Successfully created new buku keluar');
        } catch (\Throwable $th) {
            // Munculkan pesan error jika ada error
            return redirect()->route('buku_keluar.index')->with('error', $th->getMessage());
        }
    }

    /**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit($id)
    {

```

```

        //munculkan data buku keluar sesuai parameter id dan ambil
        satu data
        $buku_keluar = BukuKeluar::where('buku_keluar_id',$id)-
>firstOrFail();
        $book = Book::all();
        // jika ada data anggota
        if($buku_keluar){
            // buka halaman view buku_keluar dengan mengirim
            datanya
            return
            view('buku_keluar_edit',compact('buku_keluar','book'));
        }else{
            return redirect()->route('buku_keluar.index')-
>with('error','buku keluar not found');
        }

    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, $id)
    {
        //tambahkan validasi
        //ambil data buku keluar sesuai parameter id dan lakukan
        update pada modelnya
        BukuKeluar::where('buku_keluar_id',$id)->update([
            'created_at'=> $request->fullname,
            'quantity'=> $request->quantity,
        ]);

        return redirect()->route('buku_keluar.index')-
>with('success','Successfully update data');

    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy($id)
    {
        //lakukan delete pada data anggota sesuai parameter id
        BukuKeluar::where('buku_keluar_id',$id)->delete();

        return redirect()->route('buku_keluar.index')-
>with('success','Successfully delete data');
    }

    public function getBukuKeluarById(Request $request){

        //ajax akan meminta request untuk mengambil data anggota
        berdasarkan parameter
        $buku_keluar =
        BukuKeluar::where('buku_keluar_id',$request->id)->firstOrFail();
        // kembalikan data dalam bentuk response json
        return response()->json([

```

```

        'buku_keluar'=>$buku_keluar
    });
}
}

```

4. Buku Masuk

```

public function index()
{
    //ambil semua data last lalu bagi menjadi 5 data setiap
    page
    $buku_masuk
    BukuMasuk::join('book','buku_masuk.book_id','=','book.book_id')-
    >select([
        'book.title
    title','buku_masuk.created_at','buku_masuk.quantity'])-
    >paginate(5);
    // kembalikan halaman view buku_masuk list dengan mengirim
    datanya
    return view('buku_masuk_list',compact('buku_masuk'))
    ->with('i',(request()->input('page',1)-1)*5);
}

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    $book = Book::all();
    //tampilkan halaman add anggota
    return view('buku_masuk_add',compact('book'));
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    try {
        // Menjalankan fungsi insert pada table buku masuk
        $bukuMasuk = BukuMasuk::create($request->all());

        // Mendapatkan buku yang masuk berdasarkan ID
        $buku = Book::findOrFail($request->buku_id);

        // Menambahkan stok buku yang masuk ke stok total
        $buku->stock += $request->quantity;
        $buku->save();

        // Redirect ke halaman list buku masuk
        return redirect()->route('buku_masuk.index')-
    >with('success','Successfully created new buku masuk');
    } catch (\Throwable $th) {
        // Munculkan pesan error jika ada error
        return redirect()->route('buku_masuk.index')-
    >with('error', $th->getMessage());
    }
}
}

```

```

/**
 * Display the specified resource.
 */
public function show(string $id)
{
    //
}

/**
 * Show the form for editing the specified resource.
 */
public function edit($id)
{
    //munculkan data buku keluar sesuai parameter id dan ambil
    satu data
    $buku_masuk = BukuMasuk::where('buku_masuk_id',$id)-
>firstOrFail();
    $book = Book::all();
    // jika ada data anggota
    if($buku_masuk){
        // buka halaman view buku_masuk dengan mengirim datanya
        return
view('buku_masuk_edit',compact('buku_masuk','book'));
    }else{
        return redirect()->route('buku_masuk.index')-
>with('error','buku keluar not found');
    }

}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, $id)
{
    //tambahkan validasi
    //ambil data buku keluar sesuai parameter id dan lakukan
    update pada modelnya
    BukuMasuk::where('buku_masuk_id',$id)->update([
        'created_at'=> $request->fullname,
        'quantity'=> $request->quantity,
    ]);

    return redirect()->route('buku_masuk.index')-
>with('success','Successfully update data');

}

/**
 * Remove the specified resource from storage.
 */
public function destroy($id)
{
    //lakukan delete pada data anggota sesuai parameter id
    BukuMasuk::where('buku_masuk_id',$id)->delete();
}

```

```

        return redirect()->route('buku_masuk.index')-
>with('success','Successfully delete data');
    }

    public function getBukuMasukById(Request $request){

        //ajax akan meminta request untuk mengambil data anggota
        berdasarkan parameter
        $buku_masuk = BukuMasuk::where('buku_masuk_id',$request-
>id)->firstOrFail();
        // kembalikan data dalam bentuk response json
        return response()->json([
            'buku_masuk'=>$buku_masuk
        ]);
    }
}

```

5. Kelas

```

public function index()
{
    //ambil semua data last lalu bagi menjadi 5 data setiap
    page
    $kelas = Kelas::latest()->paginate(5);
    // kembalikan halaman view kelas list dengan mengirim
    datanya
    return view('kelas_list',compact('kelas'))
    ->with('i',(request()->input('page',1)-1)*5);
}

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    return view('kelas_add');
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    //
    try {
        // menjalankan fungsi insert pada table kelas
        Kelas::create($request->all());
        // redirect ke halaman list kelas
        return redirect()->route('kelas.index')-
>with('success','Successfully to create new kelas');
    } catch (\Throwable $th) {
        //throw $th;
        // munculkan pesan error jika ada error
        return redirect()->route('kelas.index')-
>with('error',$th->getMessage());
    }
}

```

```

    }

    /**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit($id)
    {
        //munculkan data kelas sesuai parameter id dan ambil satu
data
        $kelas = Kelas::where('kelas_id',$id)->firstOrFail();
        // jika ada data kelas
        if($kelas){
            // buka halaman view kelas_edit dengan mengirim datanya
            return view('kelas_edit',compact('kelas'));
        }else{
            return redirect()->route('kelas.index')-
>with('error','kelas not found');
        }
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, $id)
    {
        //tambahkan validasi
        //ambil data kelas sesuai parameter id dan lakukan update
pada modelnya
        Kelas::where('kelas_id',$id)->update([
            'name'=> $request->name,
        ]);

        return redirect()->route('kelas.index')-
>with('success','Successfully update data');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy($id)
    {
        //lakukan delete pada data kelas sesuai parameter id
        Kelas::where('kelas_id',$id)->delete();

        return redirect()->route('kelas.index')-
>with('success','Successfully delete data');
    }

```

```
    }  
  
    public function getKelasById(Request $request){  
        //ajax akan meminta request untuk mengambil data kelas  
        berdasarkan parameter  
        $kelas = Kelas::where('kelas_id',$request->id)-  
>firstOrFail();  
        // kembalikan data dalam bentuk response json  
        return response()->json([  
            'kelas'=>$kelas  
        ]);  
    }  
}
```


Lampiran 6 : Riwayat Hidup

Data Diri



Nama : Rangga Kurniawan
 TTL : Bandung, 02 September 2002
 Jenis Kelamin : Laki-laki
 Agama : Islam
 Alamat : Desa Wangisagara, Majalaya,
 Bandung

Pendidikan

- Universitas Bale Bandung 2020 - 2024
- SMAN 2 Majalaya 2017 - 2020

Pengalaman

- MSIB - PT. Impactbyte Teknologi Edukasi Agu - Des 2023

Motto

Masalah tidak akan menjadi masalah kalau tidak dipermasalahkan.

Contact

- Instagram : raku_12
- E-mail : adipatiraku@gmail.com